Functional Concurrent Linear Regression Model for Spatial Images

Jun ZHANG, Murray K. CLAYTON, and Philip A. TOWNSEND

Motivated by a problem in describing forest nitrogen cycling, in this paper we explore regression models for spatial images. Specifically, we present a functional concurrent linear model with varying coefficients for two-dimensional spatial images. To address overparameterization issues, the parameter surfaces in this model are transformed into the wavelet domain and a sparse representation is found by using a large-scale l_1 constrained least squares algorithm. Once the sparse representation is identified, an inverse wavelet transform is applied to obtain the estimated parameter surfaces. The optimal penalization term in the objective function is determined using the Bayesian Information Criterion (BIC) and we introduce measures of model quality. Our model is versatile and can be applied to both single and multiple replicate cases.

Key Words: Dimension reduction; LASSO; Regression models for spatial images; Remote sensing; Satellite images; Wavelet expansion.

1. INTRODUCTION

In this paper we develop methods applicable to a number of settings in the analysis of spatial data. In particular, there are numerous situations in the analysis of geographic information system data, or remotely sensed (satellite) data, where there is interest in relating the information in one image to the information in another image or set of images. We illustrate this notion with an example from a study of forest nitrogen cycling and its relationship to defoliation.

Gypsy moth defoliation of oak trees is of concern in the Appalachian Mountains because it disrupts nitrogen cycling in forests (Eshleman et al. 1998; McNeil et al. 2007; Lovett et al. 2002). Since gypsy moth defoliation varies widely in intensity and usually occurs over

Jun Zhang (✉) is Post-Doctoral Fellow, Statistical and Applied Mathematical Sciences Institute, 19 T.W. Alexander Drive, Research Triangle Park, NC 27709-4006, USA (E-mail: *jzhang@samsi.info*). Murray K. Clayton is Professor, Department of Statistics, University of Wisconsin-Madison, Madison, WI 53706, USA. Philip A. Townsend is Associate Professor, Department of Forest and Wildlife Ecology, University of Wisconsin-Madison, Madison, WI 53706, USA

Journal of Agricultural, Biological, and Environmental Statistics, Volume 16, Number 1, Pages 105–130
DOI: 10.1007/s13253-010-0047-1

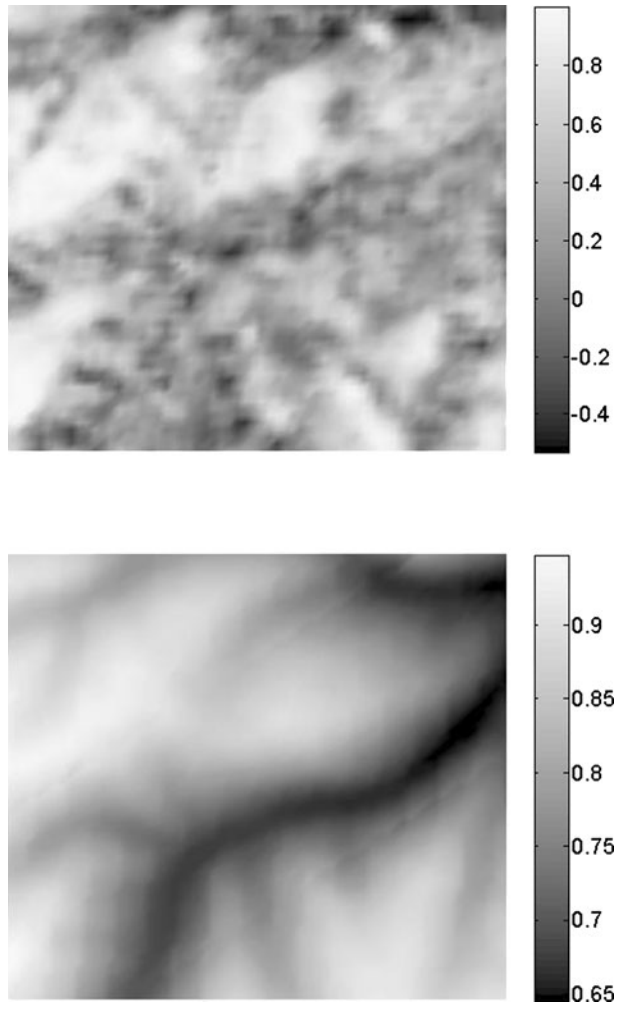

Figure 1. Top: satellite-derived map of proportion of forest defoliated (negative values indicate forest regrowth). Bottom: elevation. Units for the two images are standardized.

large areas, Townsend, Eshleman, and Welcker (2004) found it useful to use remote sensing data to predict disturbances to nitrogen cycling as a consequence of defoliation status.

Figure 1 shows satellite images of defoliation rates (proportion of defoliated forest on a per-pixel basis) caused by gypsy moth and surface elevation for the Savage River State Forest in Western Maryland, USA. The defoliation image is for an insect infestation that occurred in June–July of 2006, generated from Landsat satellite images with algorithms developed by Townsend, Eshleman, and Welcker (2004). The elevation surface is derived from the National Elevation Data set of the US Geological Survey. White in the defoliation image corresponds to high defoliation rates and black indicates foliage growth. On the elevation image, low to high elevations are depicted in black to white tones, respectively. Both the elevation data and the defoliation image have 30 m pixel resolution. These im-

ages resulted from the only available cloud-free Landsat data corresponding to the insect disturbance—a typical case in the field of remote sensing.

Several authors have observed that gypsy moth defoliation rates increase with increased elevation (Houston and Valentine 1977; Kleiner and Montgomery 1994). In Figure 1, we can see that defoliation rates are generally higher on ridges than in the valleys. In a typical Geographical Information System approach, this would be assessed by overlaying the two images and performing a visual inspection of them. It would be desirable to have a less subjective, more quantitative description of the spatial variation in the relationship between elevation and gypsy moth defoliation rates. This would provide researchers with a more accurate view of the true relationship, and it would facilitate the identification of other possible factors impacting the dynamics of defoliation caused by gypsy moth.

From Figure 1 it seems reasonable to assume that, locally, elevation affects defoliation rates linearly, although from one subregion to the next there could be a different linear relationship for these two attributes; e.g. the slopes could vary across the region. This implies that a conventional spatial linear regression model is not suitable for the data and models based on non-stationary Gaussian processes might be considered instead. Another issue with the application of conventional spatial linear regression models to remote sensing data is that satellite images are huge matrices with many elements (e.g. Landsat images are 8000 by 8000 grid cells, with multiple channels). For common regression data analyzed by geostatistical methods typical data sets might number in the hundreds. However, for even a small 640-by-480 image, to obtain a variogram requires computing the summand in a variogram estimator almost $\binom{300,000}{2} \approx 4.5 \times 10^{10}$ times. One alternative would be to sample 1000 cells (say) and compute a variogram from the sample, although the non-stationarity of the data might not be captured by this. The extraordinary richness of remotely sensed data implies that, even when a variogram can be calculated, common models are unlikely to fit. If we want to use all of the data at hand and address non-stationarity at the same time, then the development of new modeling approaches is necessary.

In this paper we will explore local linear models. The simplest local linear model is a functional concurrent linear model with varying coefficients:

$$y_i = a_i + b_i x_i + e_i \tag{1.1}$$

where y_i is the value of pixel i in the response image, x_i is the value of pixel i in the explanatory image, e_i is the error term associated with pixel i, and a_i and b_i are parameters. This model is over-parameterized, and so to estimate parameters for the model, we need to impose some constraints. In essence, our approach will involve estimating two parameter surfaces, a constant term surface and a slope term surface, the constraints that they should be at least locally smooth. In particular, we use wavelet expansions to estimate the parameter surfaces, thus constraining the a_i and b_i values and, as we will see, reducing the problem of one with potentially millions of parameters to a problem of several dozen parameters.

Although the specifics regarding the use of wavelet expansions are discussed in Section 2, we note here an important contrast between our approach and some other commonly used tools in spatial statistics. First, unlike a typical geostatistical approach based

on a correlation structure described with a variogram (Cressie 1993), our model does not include a covariance structure among the errors. But, similar to typical spline models, our model does include spatial dependence in the parameter surfaces through the sparsity penalization term. Generally, we believe that it is important in the modeling to capture fine scale spatial relationships, but acknowledge that there are competing approaches for doing so. Functional models and geostatistical models both work well in the spatial domain and both have weakness and strengths—the discussion between Cressie and Wahba is relevant in this regard (Wahba and Cressie 1990).

Another common tool, again for describing spatial correlation is the use of a Conditional AutoRegressive model (CAR) or Simultaneously AutoRegressive model (SAR) each of which provide covariance structures for random fields (Besag 1974; Besag and Kooperberg 1995), and, for example, Gelfand and Vounatson (2003) use multivariate CAR on spatial data analysis. Moreover, CAR models have a specific advantage in terms of computational feasibility (Anselin and Florax 1995; Stern and Cressie 2000; Bell and Broemeling 2000). At the same time, CAR and SAR models are highly structured (perhaps deceptively so Wall 2004), and in their most useful form are also stationary, which we think is contrary to the form of our data. Overall, due to the non-stationarity and computational tractability problems mentioned in previous paragraphs, we think functional concurrent linear regression models are a reasonable choice, especially when a form of point-wise linear relationship matches the scientific goals at hand.

The concurrent linear model has been investigated in the time domain by a number of authors. Such a model is called a varying-coefficient model by Hastie and Tibshirani (1993). Earlier, West, Harrison, and Migon (1985) investigated dynamic linear models including AR(1)-type varying coefficients. Nevertheless, Hastie and Tibshirani (1993) covered more details and examples of this type of model; more recently Eubank et al. (2004) investigated smoothing spline estimation with a concurrent model and Gelfand et al. (2003) studied a concurrent model in the spatial setup based on a Bayesian framework. But the model discussed in Gelfand et al. (2003) assumes two stationary Gaussian processes for the constant and slope parameter surfaces. Our model does not have such an assumption and can be applied to a broader set of problems.

An alternative perspective on addressing the problem in this paper is to use functional linear models. Although conceptually these have a different construction, in fact in our application they will be equivalent to the locally linear models outlined above. Functional linear models have been developed primarily in the time series domain—the key idea behind functional regression according to Ramsay and Silverman (2005) is to view a series of data in the time domain as a smooth function and express this function as the summation of *a few* weighted basis functions:

$$X(t) = \sum_{j=1}^{K} c_j \phi_j(t). \tag{1.2}$$

We note that "a few" is important here. The direct consequence of Equation (1.2) is that we have many fewer parameters c_j than the number of original data points. This serves as

a way to reduce the dimensionality of the original data if we view each time point as one dimension.

There are many dimension reduction methods available. We often choose the appropriate dimension reduction method based on the structure of the dataset and the capabilities of the dimension reduction methods (Fodor 2002). Our approach is in the family of Matching Pursuit (MP) methods first proposed by Mallat and Zhang (1993). These methods are popular in the signal processing community (Pati, Rezaiifar, and Krishnaprasad 1993; Tropp and Gilbert 2007) and work well for one-dimensional or two-dimensional grid data. By the appropriate choice of basis functions, and by expressing the problem as a constrained optimization problem, we have constructed a form of MP method that can handle the non-stationary spatial images in our problem.

Among the many bases available for dimension reduction, we chose wavelets for several reasons. First, we would like to apply our model to non-stationary spatial images. For stationary images, conventional Fourier bases should be sufficient. With non-stationary two-dimensional spatial images, a Wavelet Transform will keep the spatial domain information while extracting the frequency domain information from the signals. Second, there are many other advantages when we use wavelets. Donoho and Johnstone (1994) prove that adaptive wavelet filtering such as SUREShrink (Stein Unbiased Risk Estimation Shrinkage) can adapt to unknown smoothness. In other words, wavelet based reconstruction will contain abrupt changes if the original curves contain abrupt changes and will contain smooth surfaces if the original curves are smooth. The spatial adaptability of wavelet bases includes many smooth function classes such as the Sobolev class and Bounded Variation (Donoho et al. 1995). Donoho et al. (1995) also show that data-driven wavelet shrinkage is nearly minimax. Finally, since the wavelet bases are designed hierarchically, we can conveniently use wavelet expansions to extract multiscale features in the parameter surfaces.

As noted, most of the literature on functional linear models addresses data in the time domain. For example, Faraway (2000) studied reach motion data with a functional linear regression model with scalar explanatory variables, functional responses and functional parameters. Ratcliffe, Leader, and Heller (2002) used a functional linear regression model with both functional and scalar covariates to study periodically stimulated fetal heart rate data. Yamanishi and Tanaka (2003) presented a spatially weighted functional linear model with functional responses, functional covariates and functional parameters. Goutis (1998) studied a functional regression model with second derivatives and demonstrated the flexibility of functional models by showing that we can also include the second derivatives in the model.

There is considerably less literature dealing with functional regression analysis in the spatial domain. For example, Hastie, Buja, and Tibshirani (1995) described an image classification functional model while Kaufman and Sain (2007) applied a Bayesian functional analysis of variance model on two-dimensional temperature data. In light of the vastly increasing volume of remotely sensed data being collected, there is a clear need for the additional development of tools in this realm.

In this paper, we present a functional concurrent linear model for the regression analysis of images. We utilize two-dimensional wavelet expansions to represent the spatially

indexed intercept and slope surfaces. To reduce the dimensionality, it is common in wavelet approaches to use a form of thresholding. We propose a data-driven form of this via a variation of LASSO. Put another way, we achieve a sparse basis representation by using l_1 penalized least squares estimation. The penalization term itself is determined with a data-based approach using BIC. In addition, we develop quantitative measures to evaluate the model fit and express variation in the estimates.

This paper is organized as follows: In Section 2, we describe the main theory, how to automatically find the penalization term with a data-driven approach, and how to quantitatively evaluate the model fit; Section 3 provides examples and in Section 4 we discuss possible extensions of our current framework.

2. FUNCTIONAL CONCURRENT LINEAR MODEL

2.1. MAIN THEORY

In Section 1, we introduced a functional concurrent linear model for gypsy moth defoliation data. Henceforth we assume that the real parameter surfaces of the model are smooth or at least locally smooth and can be sparsely expressed in the wavelet domain. Without the smoothness assumption it is impossible to solve the problem effectively.

We illustrate the functional concurrent linear model assuming that there is one explanatory variable—the following argument can be easily extended to cases with multiple explanatory variables. Suppose we have n pairs of images, $(\mathbf{Y}_i, \mathbf{X}_i)$, $i = 1, \ldots, n$, $n \geq 1$. A functional linear concurrent model with varying coefficients for these n pairs of images can be written as:

$$Y_i(t_1, t_2) = \beta_0(t_1, t_2) + X_i(t_1, t_2)\beta(t_1, t_2) + E_i(t_1, t_2) \tag{2.1}$$

where $\beta_0(t_1, t_2)$ is a constant term, $\beta(t_1, t_2)$ is the slope term for $X_i(t_1, t_2)$, $E_i(t_1, t_2)$ is the error term and t_1 and t_2 are spatial indexes in two (orthogonal) directions. We rewrite our model in matrix form:

$$\mathbf{Y}_i = \mathbf{A} + \mathbf{X}_i \circ \mathbf{B} + \mathbf{E} \tag{2.2}$$

where $\mathbf{Y}_i$, $\mathbf{X}_i$, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{E}$ are M by N matrices, "$\circ$" stands for the Schur product, and $\mathbf{E}$ is the error matrix. For matrix $\mathbf{B}$, the element at row t_1 column t_2 is $\beta(t_1, t_2)$. For matrix $\mathbf{A}$, the element at row t_1 column t_2 is $\beta_0(t_1, t_2)$.

We next expand the parameter matrices $\mathbf{A}$ and $\mathbf{B}$ (2.2) with a two-dimensional discrete wavelet expansion and obtain

$$\mathbf{Y}_i = \sum_{j=1}^{H} v_j \mathbf{\Phi}_j + \mathbf{X}_i \circ \left\{ \sum_{j=1}^{H} w_j \mathbf{\Phi_j} \right\} + \mathbf{E} = \sum_{j=1}^{H} v_j \mathbf{\Phi_j} + \sum_{j=1}^{H} w_j (\mathbf{X}_i \circ \mathbf{\Phi}_j) + \mathbf{E} \tag{2.3}$$

where $H = M \times N$ and $\mathbf{\Phi_j}$ is the two-dimensional wavelet base (Walker 1999). To estimate the coefficients in this expression we use a shrinkage procedure based on LASSO (Hastie,

Tibshirani, and Friedman 2001). Specifically, we seek to solve

$$\min_{\mathbf{v},\mathbf{w}} \sum_{i=1}^{n} \left\| \mathbf{Y}_i - \sum_{j=1}^{H} v_j \mathbf{\Phi}_j - \sum_{j=1}^{H} w_j (\mathbf{X}_i \circ \mathbf{\Phi}_j) \right\|_F^2 \tag{2.4}$$

$$\text{subject to} \quad \sum_{j=1}^{H} |v_j| + \sum_{j=1}^{H} |w_j| \le t$$

where $\|\cdot\|_F$ is the Frobenius matrix norm. In (2.4), we use the sum of absolute values of wavelet coefficients i.e. l_1 norm, as the constraint. One major advantage of using l_1 constraints is that many of the wavelet coefficients v_j and w_j in (2.4) will become exactly zero (Tibshirani 1996). This is appealing since under our local smoothness assumption we should have only a few non-zero v_js and w_js. Moreover, this means that our original problem in (1.1) will no longer be over-parameterized.

To proceed, we want to utilize the l_1 constrained Least Squares Estimation (LSE) algorithm mentioned in Kim et al. (2007), and this requires us to change (2.4) to the form used by Kim et al. (2007). First, if we define $\mathbf{X}_i^j = \mathbf{X}_i \circ \mathbf{\Phi}_j$, then (2.3) becomes:

$$\mathbf{Y}_i = \sum_{j=1}^{H} v_j \mathbf{\Phi}_j + \sum_{j=1}^{H} w_j \mathbf{X}_i^j + \mathbf{E}. \tag{2.5}$$

We can write (2.5) in a vectorized form:

$$\begin{aligned} \operatorname{vec}(\mathbf{Y}_i) = \big(&\operatorname{vec}(\mathbf{\Phi}_1) \quad \dots \quad \operatorname{vec}(\mathbf{\Phi}_H) \quad \operatorname{vec}\big(\mathbf{X}_i^1\big) \quad \dots \quad \operatorname{vec}\big(\mathbf{X}_i^H\big)\big) \\ &\times (v_1 \dots v_H \; w_1 \dots w_H)^T \end{aligned}$$

where $\operatorname{vec}(\mathbf{Z})$ is defined by writing a matrix $\mathbf{Z}$ as a vector column-wise. Now define:

$$\mathbf{u} = \begin{pmatrix} v_1 & \dots & v_H & w_1 & \dots & w_H \end{pmatrix}^T,$$

$$\mathbf{t} = \begin{pmatrix} \operatorname{vec}(\mathbf{Y}_1)^T & \operatorname{vec}(\mathbf{Y}_2)^T & \dots & \operatorname{vec}(\mathbf{Y}_n)^T \end{pmatrix}^T,$$

$$\mathbf{Z} = \begin{pmatrix} \operatorname{vec}(\mathbf{\Phi}_1) & \dots & \operatorname{vec}(\mathbf{\Phi}_H) & \operatorname{vec}(\mathbf{X}_1^1) & \dots & \operatorname{vec}(\mathbf{X}_1^H) \\ \operatorname{vec}(\mathbf{\Phi}_1) & \dots & \operatorname{vec}(\mathbf{\Phi}_H) & \operatorname{vec}(\mathbf{X}_2^1) & \dots & \operatorname{vec}(\mathbf{X}_2^H) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \operatorname{vec}(\mathbf{\Phi}_1) & \dots & \operatorname{vec}(\mathbf{\Phi}_H) & \operatorname{vec}(\mathbf{X}_n^1) & \dots & \operatorname{vec}(\mathbf{X}_n^H) \end{pmatrix}.$$

Then we can rewrite (2.4) in the standard form:

$$\min_{\mathbf{u}} \|\mathbf{t} - \mathbf{Z}\mathbf{u}\|_2^2 + \lambda \|\mathbf{u}\|_1. \tag{2.6}$$

With (2.6), we can directly apply the large-scale l_1 penalized least squares algorithm described in Kim et al. (2007) to obtain $\hat{\mathbf{u}}$. The wavelet coefficients corresponding to large details in the parameter surfaces normally will not have identifiability issues since $\mathbf{X}_i \circ \mathbf{\Phi}_j$ and $\mathbf{\Phi}_j$ cannot be highly correlated unless the covariate image $\mathbf{X}$ is very flat inside the support of $\mathbf{\Phi}_j$. We also note that although Haar wavelet bases are coarse, they are symmetric and have small support. This makes the design matrix $\mathbf{Z}$ in (2.6) sparse, which in turn makes the computation fast.

Now, applying a 2-D inverse wavelet transform on the first half of $\hat{\mathbf{u}}$ or $(\hat{v}_1 \ \dots \ \hat{v}_H)$, we obtain the estimated constant surface $\hat{\mathbf{A}}$. Similarly, the estimated slope surface $\hat{\mathbf{B}}$ can be obtained by applying a 2-D inverse wavelet transform on the second half of $\hat{\mathbf{u}}$ or $(\hat{w}_1 \ \dots \ \hat{w}_H)$. Our estimation algorithm can be described as follows:

(1) Transform parameter surfaces into the wavelet domain to obtain their sparse representations.

(2) Use a large-scale l_1 penalized least squares algorithm to find appropriate sparse representations in the wavelet domain.

(3) Perform an inverse discrete wavelet transform to obtain estimated parameter surfaces.

As we mentioned earlier in Section 1, a typical case would be that (Y_1, X_1) is the only set of images available. In this case u, t and Z in (2.6) would be

$$\begin{aligned}
\mathbf{u} &= \begin{pmatrix} v_1 & \dots & v_H & w_1 & \dots & w_H \end{pmatrix}^T, \\
\mathbf{t} &= \begin{pmatrix} \operatorname{vec}(\mathbf{Y}_1) \end{pmatrix}, \\
\mathbf{Z} &= \begin{pmatrix} \operatorname{vec}(\mathbf{\Phi}_1) & \dots & \operatorname{vec}(\mathbf{\Phi}_H) & \operatorname{vec}\left(\mathbf{X}_1^1\right) & \dots & \operatorname{vec}\left(\mathbf{X}_1^H\right) \end{pmatrix}.
\end{aligned}$$

We should note that the problem becomes underdetermined, since $\mathbf{t}$ has only H elements and $\mathbf{u}$ has $2H$ unknown variables, although this does not present a problem in terms of using the LASSO solver to obtain a solution.

The underdetermined nature of the problem does make it more difficult to evaluate the uncertainty of the solution, a point we cover in more detail in the Section 2.3. For this reason, we invoke a minor simplification. Since the finest wavelet bases have small support size, they will make the estimation sensitive to any local abnormal condition of $\mathbf{X}_1$. Thus we exclude the finest wavelet bases from the model and $\mathbf{u}$, $\mathbf{t}$ and $\mathbf{Z}$ become

$$\begin{aligned}
\mathbf{u} &= \begin{pmatrix} v_1 & \dots & v_{H/4} & w_1 & \dots & w_{H/4} \end{pmatrix}^T, \\
\mathbf{t} &= \begin{pmatrix} \operatorname{vec}(\mathbf{Y}_1) \end{pmatrix}, \\
\mathbf{Z} &= \begin{pmatrix} \operatorname{vec}(\mathbf{\Phi}_1) & \dots & \operatorname{vec}(\mathbf{\Phi}_{H/4}) & \operatorname{vec}\left(\mathbf{X}_1^1\right) & \dots & \operatorname{vec}\left(\mathbf{X}_1^{H/4}\right) \end{pmatrix}.
\end{aligned}$$

Now we obtain an overdetermined system with $H/2$ variables and H observations, and we have $H/4$ wavelet coefficients for each parameter surface since we have removed $3H/4$ wavelet coefficients corresponding to the finest wavelet bases. Although excluding the finest wavelet coefficients from the model may result in the loss of some details of the parameter surfaces, it improves the computational speed substantially. Regardless, we note that in most cases, we obtained similar results when either including or excluding the finest wavelet bases. From this point forward, we will focus on the case with only one set of images and apply functional concurrent linear models with finest level wavelet bases excluded.

2.2. FINDING THE OPTIMAL PENALIZATION TERM λ

An essential step in LASSO is the selection of the penalization term λ, and there are many papers and books devoted to this topic. λ (Efron et al. 2004; Hastie, Tibshirani, and Friedman 2001). In particular, the Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) are two useful criteria for picking a good λ. AIC and BIC are defined here as follows:

$$\mathrm{AIC}(\lambda) = \frac{1}{N}\left(\frac{\| \mathbf{Y} - \hat{\mathbf{Y}} \|_F^2}{\hat{\sigma}^2} + 2k(\lambda)\right),$$

$$\mathrm{BIC}(\lambda) = \frac{1}{N}\left(\frac{\| \mathbf{Y} - \hat{\mathbf{Y}} \|_F^2}{\hat{\sigma}^2} + \log(N)k(\lambda)\right)$$

where N is the sample size, $\mathbf{Y}$ is the observed response image, $\hat{\mathbf{Y}}$ is the fitted response image, $\hat{\sigma}^2$ is the estimated variance of the noise in the response image and $k(\lambda)$ is the degrees of freedom or the effective parameter number of the model. We will focus on BIC in this paper.

The selection of the penalization term λ is done by finding the λ_{opt} which minimizes BIC or

$$\lambda_{\mathrm{opt}} = \arg\min_{\lambda} \mathrm{BIC}(\lambda).$$

Before we can minimize BIC, we need to estimate $k(\lambda)$ and σ^2 first. For $k(\lambda)$, we will use the number of non-zero coefficients as an unbiased estimate of the degrees of freedom of the LASSO based on the arguments in Zou, Hastie, and Tibshirani (2007). To estimate $\hat{\sigma}^2$, we apply a local linear model to obtain the point-wise residuals and use the sample variances of the point-wise residuals as $\hat{\sigma}^2$.

Searching for λ_{opt} requires us to compute the BIC scores through a sequence of equally spaced values of λ since we do not have an analytical form of BIC scores in terms of λ. There exists a $\lambda_{\max}$ such that, for any λ greater than $\lambda_{\max}$, a zero vector is the solution for (2.6) (Osborne, Presnell, and Turlach 2000). Thus we only need to search between 0 and $\lambda_{\max}$ to find the λ_{opt}, and $\lambda_{\max}$ is easy to find by $\lambda_{\max} = \|\mathbf{Z}^T\mathbf{T}\|_\infty$ using $\mathbf{Z}$ and $\mathbf{T}$ in (2.6) (Osborne, Presnell, and Turlach 2000).

2.3. EVALUATION OF THE FIT

Since we focus on the case of a single pair of input and response images, we use a measure suggested by Ramsay and Silverman (2005) and defined by

$$R^2 = 1 - \frac{\|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2}{\|\mathbf{Y} - \bar{\mathbf{Y}}\|_F^2}$$

where $\mathbf{Y}$ is the observed response image, $\hat{\mathbf{Y}}$ is the estimated response image and $\bar{\mathbf{Y}}$ is the mean image whose pixel values are equal to the mean of all pixel values in $\mathbf{Y}$.

Although useful, R^2 can only give us limited information on the fit, and as we will see, it is also useful to determine the Point-wise Standard Deviation (PSD) of the estimated

parameter surface. This is quite challenging, and here we confine ourselves to a brief discussion of methods for doing so, and propose one approach.

To estimate the PSD, we must estimate the variance of the LASSO estimates. As noted, we can think of the LASSO as a tool for selecting variables in a model (any variable estimated to be zero is not "selected" for the model). Recall the formulation of our problem in (2.6) and suppose the first d elements in $\mathbf{u}$ are those that are selected by the LASSO algorithm. Then the solution can be written as $\mathbf{u}^\star = [\mathbf{u}_1^{\star T}\, \mathbf{u}_2^{\star T}]^T$ where $\mathbf{u}_1^\star$ is a d-dimensional non-zero vector and $\mathbf{u}_2^\star$ is a $(H/2 - d)$-dimensional zero vector.

Many authors only estimate the variance of $\mathbf{u}_1^\star$ and set the variance of $\mathbf{u}_2^\star$ to zero. For example, Tibshirani (1996) uses an approximate ridge regression to estimate the uncertainties of $\mathbf{u}_1^\star$. Fan and Li (2001) and Zou (2006) use a local quadratic approximation to obtain a sandwich formula for the covariance matrix of $\mathbf{u}_1^\star$. Unfortunately, when the design matrix $\mathbf{Z}$ is ill-conditioned (i.e. highly collinear), LASSO may give a solution far from the true solution and yet the estimated variances for $\mathbf{u}_1^\star$ are small. Our experience, for example, with Fan and Li (2001)'s sandwich formula applied to numerical examples leads us to conclude that only estimating the variances for selected variables may perform poorly.

Tibshirani (1996) also suggests using the bootstrap to estimate the variances of a LASSO estimator. But a recent paper by Kyung et al. (2009) shows that if a true variable is zero then bootstrap estimation of that variable is not consistent. More broadly, only calculating the standard errors may be inadequate for describing the true uncertainty of the LASSO estimator since Pötscher and Leeb (2007) show the distribution of the LASSO estimator is a mixture of two truncated normal densities. Of course, it still gives a general sense of the variability of the estimator; perhaps just as important, we will see that it can be used to differentiate the good estimators from the bad ones in most cases.

Osborne, Presnell, and Turlach (2000) give an approximation for the covariance matrix of $\mathbf{u}^\star$ which gives non-zero variances for $\mathbf{u}_2^{\star T}$. Their formula is

$$\operatorname{cov}(\mathbf{u}^\star) = \big(\mathbf{Z}^T\mathbf{Z} + \mathbf{W}\big)^{-1}\mathbf{Z}^T\mathbf{Z}\big(\mathbf{Z}^T\mathbf{Z} + \mathbf{W}\big)^{-1}\hat{\sigma}^2 \tag{2.7}$$

where $\mathbf{W} = \mathbf{c}\mathbf{c}^T/(\|\mathbf{u}^\star\|_1 \| \cdot \|\mathbf{c}\|_{\inf})$, $\mathbf{c} = \mathbf{Z}^T\mathbf{r}$, $\mathbf{r} = \mathbf{t} - \mathbf{Z}\mathbf{u}^\star$ and $\hat{\sigma}^2$ is the estimated variance of the noise in the response image. We propose using this for estimating the variance of the parameter surfaces and note that, as an additional tool for model checking, it reflects the condition of the design matrix $\mathbf{Z}$. Thus the condition of the covariate image $\mathbf{X}$ can be measured by the PSD of the estimates of the parameter surfaces.

There are two limitations in the use of (2.7) to approximate the covariance matrix of $\mathbf{u}^\star$. First, since the number of wavelet coefficients is $O(H)$, the number of elements in $\operatorname{cov}(\mathbf{u}^\star)$ is $O(H^2)$, and this can grow quite fast as the number of pixels increases. To cope with this issue, one option is to exclude several levels of small detail wavelet coefficients from our model to keep the number of parameters relatively low when image sizes increase. This can result in the loss of some low level details but the global structure of the parameter surfaces is still captured. In addition, based on our empirical experience, the covariance matrix of the LASSO estimator has a large number of elements close to zero. Similarly, many matrices involved in the PSD computation have a large number of elements close to zero. We can approximate these matrices by setting near zero elements to zero. This

will save a great deal of memory consumption and will not have too much impact on the final PSD estimation. Thus, whenever we have large images, we can either exclude several levels of small detail wavelet coefficients from our model or use approximate matrices for the computation to alleviate the computational burden. Of course, we can combine these two approaches together to obtain the estimation of the PSD.

The second issue is that $(\mathbf{Z}^T\mathbf{Z}+\mathbf{W})^{-1}$ is not always computable. Osborne, Presnell, and Turlach (2000) assume that $\mathbf{Z}$ is of full rank. If this is not the case, but if only a few columns of $\mathbf{Z}$ are highly correlated, it is possible to use an incomplete LU decomposition to obtain an approximation of $(\mathbf{Z}^T\mathbf{Z}+\mathbf{W})^{-1}$. If many columns of $\mathbf{Z}$ are highly correlated, then it is nearly impossible to compute $(\mathbf{Z}^T\mathbf{Z}+\mathbf{W})^{-1}$. This latter situation arises when we have binary covariate images.

Given these limitations, we still find it useful to use Osborne, Presnell, and Turlach (2000)'s approximation to compute the covariance matrix of LASSO estimators. Specifically, once we obtain $\operatorname{cov}(\mathbf{u}^\star)$, we rewrite this as $\mathbf{u}^{\star T}=[\mathbf{u}_A^{\star T}\mathbf{u}_B^{\star T}]^T$ where $\mathbf{u}_A^{\star T}$ is an $H/4$-dimensional vector and $\mathbf{u}_B^{\star T}$ is an $H/4$-dimensional vector, and $\mathbf{u}_A^\star$ is the wavelet coefficient vector for the constant parameter surface $\mathbf{A}$ and $\mathbf{u}_B^\star$ is the wavelet coefficient vector for the slope parameter surface $\mathbf{B}$. Now $\operatorname{cov}(\mathbf{u}^\star)$ can be written as

$$\left(\begin{array}{c|c}\operatorname{cov}(\mathbf{u}_A^\star) & \operatorname{cov}(\mathbf{u}_{AB}^\star)\\ \hline \operatorname{cov}(\mathbf{u}_{BA}^\star) & \operatorname{cov}(\mathbf{u}_B^\star)\end{array}\right).$$

Since we have the following relationship:

$$\begin{pmatrix}\operatorname{vec}(\hat{\mathbf{A}})\\ \operatorname{vec}(\hat{\mathbf{B}})\end{pmatrix}=\begin{pmatrix}\mathbf{Q} & \mathbf{O}\\ \mathbf{O} & \mathbf{Q}\end{pmatrix}\begin{pmatrix}\mathbf{u}_A^\star\\ \mathbf{u}_B^\star\end{pmatrix}$$

where $Q=[\operatorname{vec}(\mathbf{\Phi}_1)\ \ldots\ \operatorname{vec}(\mathbf{\Phi}_{H/4})]$, then

$$\operatorname{cov}\begin{pmatrix}\operatorname{vec}(\hat{\mathbf{A}})\\ \operatorname{vec}(\hat{\mathbf{B}})\end{pmatrix}=\begin{pmatrix}\mathbf{Q}\operatorname{cov}(\mathbf{u}_A^\star)\mathbf{Q}^T & \mathbf{Q}\operatorname{cov}(\mathbf{u}_{AB}^\star)\mathbf{Q}^T\\ \mathbf{Q}\operatorname{cov}(\mathbf{u}_{BA}^\star)\mathbf{Q}^T & \mathbf{Q}\operatorname{cov}(\mathbf{u}_B^\star)\mathbf{Q}^T\end{pmatrix}.$$

Then the vectorized PSD of $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are

$$\operatorname{vec}(\mathbf{P}_A)=\operatorname{sqrt}\bigl(\operatorname{diag}\bigl(\mathbf{Q}\operatorname{cov}(\mathbf{u}_A^\star)\mathbf{Q}^T\bigr)\bigr) \tag{2.8}$$

and

$$\operatorname{vec}(\mathbf{P}_B)=\operatorname{sqrt}\bigl(\operatorname{diag}\bigl(\mathbf{Q}\operatorname{cov}(\mathbf{u}_B^\star)\mathbf{Q}^T\bigr)\bigr), \tag{2.9}$$

where sqrt is the point-wise squared root function and diag means the diagonal matrix of the input matrix. Finally we can resize $\operatorname{vec}(\mathbf{P}_A)$ and $\operatorname{vec}(\mathbf{P}_A)$ to $M\times N$ matrices and plot them as images. We should note that this is a point-wise standard deviation—simultaneous standard deviation surfaces are difficult to obtain since the distribution of $\mathbf{u}^\star$ is complex (Pötscher and Leeb 2007).

2.4. SOME IMPLEMENTATION DETAILS

In our examples the images have size $2^M \times 2^N$, which fits well with the Haar basis wavelets that we focus on. The issue of what to do when images do not conform to this is complex. In our applications here, we use "symmetric padding" at the boundary of the original response image $\mathbf{Y}$ and covariate image $\mathbf{X}$ to bring them to the right size. Padding is a common technique used in wavelet analysis of finite length signals, and Addison (2002) provides details. When the original $\mathbf{Y}$ and $\mathbf{X}$ images do not have the same pixel size, it is difficult to directly apply our model. To address this issue, we invoke the resizing function, imresize(), in the image processing toolbox of MATLAB® to bring all images to the same size. The resizing algorithm in MATLAB is described in Netravali and Haskell (1995) and Unser, Aldroubi, and Eden (1991).

Finally, prior to implementation, we must standardize the raw covariate image $\mathbf{X}_0$ (Tibshirani 1996) to remove the impact of different measurement units. Since our model can be written as $\mathbf{Y} = \mathbf{A} \circ \mathbf{J} + \mathbf{B} \circ \mathbf{X}$, we standardize the raw covariate image $\mathbf{X}_0$ by $\mathbf{X} = (\mathbf{X}_0 - m_x\mathbf{J})/s_x + \mathbf{J}$, where $\mathbf{J}$ is a $M \times N$ matrix with each element equal to 1, m_x is the point-wise mean of $\mathbf{X}_0$ and s_x is the point-wise standard error of $\mathbf{X}_0$. Although it is not required to standardize the raw response image $\mathbf{Y}_0$, we often standardize the raw response image $\mathbf{Y}_0$ by $\mathbf{Y} = (\mathbf{Y}_0 - m_y\mathbf{J})/s_y$, where m_y is the point-wise mean of $\mathbf{Y}_0$ and s_y is the point-wise standard error of $\mathbf{Y}_0$.

3. NUMERICAL RESULTS

In this section, we first use a series of simulated examples to demonstrate our methods, and then apply our model to the gypsy moth defoliation data. As noted previously, we focus on one-replicate examples since these are typical situation. For each example, the penalty term λ was determined by BIC, we implemented our algorithm with MATLAB version 7.1 and we used a commodity laptop with a 2G Hz Intel Core 2 T7200 CPU and 2G memory.

3.1. EXAMPLES BASED ON SIMULATED DATA

The functional concurrent linear model we used to generate simulated data is:

$$\mathbf{Y}(v_1, v_2) = \mathbf{A}(v_1, v_2) + \mathbf{X}(v_1, v_2)\mathbf{B}(v_1, v_2) + \mathbf{E}(v_1, v_2)$$

where $\mathbf{X}$ is the covariate, $\mathbf{Y}$ is the response, $\mathbf{A}$ is the constant parameter surface, $\mathbf{B}$ is the slope parameter surface, $\mathbf{E}$ is the error matrix, $0 \le v_1 \le 1$ and $0 \le v_2 \le 1$. The following two parameter surfaces $\mathbf{A}$ and $\mathbf{B}$ were used in every simulated data example:

$$\mathbf{A}(v_1, v_2) = \begin{cases} 1 & 0 \le v_1 \le 0.25, 0 \le v_2 \le 0.53, \\ 2 & 0.25 < v_1 \le 1.0, 0 \le v_2 \le 0.53, \\ 3.2 & 0 \le v_1 \le 0.5, 0.53 < v_2 \le 1.0, \\ 4 & 0.5 < v_1 \le 1.0, 0.53 < v_2 \le 1.0, \end{cases}$$

$$\mathbf{B}(v_1, v_2) = \begin{cases} 2.0 & 0 \le v_1 \le 0.5, 0 \le v_2 \le 0.5, \\ 1.0 & 0.5 < v_1 \le 1.0, 0 \le v_2 \le 0.5, \\ 3.2 & 0 \le v_1 \le 0.25, 0.5 < v_2 \le 1.0, \\ 1.5 & 0.25 < v_1 \le 1.0, 0.5 < v_2 \le 1.0. \end{cases}$$

The above two parameter surfaces are step functions. We purposely set 0.53 as one of the boundaries for the parameter surface **A**. This makes the estimation harder because we now have a thin stripe-like region in the middle of the image which has quite different combinations of values for surfaces **A** and **B**. For a specific $M \times N$ resolution, we discretize the region $(0 \le v_1 \le 1, 0 \le v_2 \le 1)$ to a $M \times N$ equally spaced grid indexed by (t_1, t_2). In the rest of Section 3, we use index (t_1, t_2) when we describe the functions in the numeric examples.

In the first two simulated data examples, the images sizes are 64×64, which means we have total of 2048 wavelet coefficients for the parameter surfaces **A** and **B**. We use a total of five levels of Haar wavelet bases in these two examples. For Haar wavelet bases, the widths and heights of the wavelet basis supports increase by a factor of two when moving from a finer wavelet level to a coarser wavelet level. Thus, in our examples the support of the smallest base is 4×4 and the support of the largest base is 64×64. The wavelet coefficients for the lower level correspond to finer details in the image while the wavelet coefficients for the higher level correspond to coarser details (Walker 1999). For the true parameter surfaces, we have a total of 42 non-zero wavelet coefficients out of a total of 2048 wavelet coefficients. The true **A** surface has 37 non-zero wavelet coefficients and the true **B** surface has five non-zero wavelet coefficients. Thus the true representation of the parameter surfaces in the wavelet domain is sparse.

In the first example, the original $\mathbf{X}_0$ was set equal to

$$\mathbf{X}_0(t_1, t_2) = 3\cos(8\pi t_1) + 4\sin(8\pi t_2)\, t_2 + 4 \tag{3.1}$$

from which we obtained the standardized **X**. We used the formula $\mathbf{Y} = \mathbf{A} + \mathbf{X} \circ \mathbf{B} + \mathbf{E}$ to generate the response image **Y**. For this and the next example, we use the centered response $\mathbf{Y}_c = \mathbf{Y} - m_y J$ in the actual computation. The reason why we use the centered response $\mathbf{Y}_c$ is we want to visually compare the estimated parameter surfaces $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ with the true **A** and **B** and it is straightforward to obtain estimated parameter surfaces with $\hat{\mathbf{A}} = \hat{\mathbf{A}}_c + m_y \mathbf{J}$ and $\hat{\mathbf{B}} = \hat{\mathbf{B}}_c$, where $\hat{\mathbf{A}}_c$ and $\hat{\mathbf{B}}_c$ are obtained based on $\mathbf{Y}_c$ and **X**.

We set **E**'s elements to be i.i.d. $N(0, \sigma_e^2)$ with $\sigma_e/\sigma_y = 10\%$, where σ_y is the estimated standard deviation of the pixel values of image **Y**. For this example, $\lambda_{\max}$ was 194.00, the estimated $\hat{\lambda}_{\text{opt}}$ was 1.45, the computing time required for estimating $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ was 18 seconds and for estimating the PSD of the estimated surfaces was 144 seconds. The R^2 was 0.9856.

The PSD of $\hat{\mathbf{B}}$ in Figure 2 has peaks where the covariate image **X** has local extrema. These peaks are caused by the flat area around each extremum in the covariate image **X** and correspond to a local relatively limited amount of information. Also, we can see that the amplitude of **X** affects the size of PSD. The smaller the local amplitude of **X** is, the higher the local size of the PSD of $\hat{\mathbf{B}}$ is. As for the PSD of $\hat{\mathbf{A}}$, it only has high peaks in the areas corresponding to the peak areas of **X** and has low peaks in the areas corresponding

to the trough area of $\mathbf{X}$. This particular outcome is caused by the standardization of the covariate matrix, and we have observed that if we standardize $\mathbf{X}$ with a different mean and standard error 1 then there will be high peaks in the area corresponding to the trough area of $\mathbf{X}$. In other words, our standardization strategy shrinks the estimation uncertainty.

Overall, the PSDs of $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are reasonable. Based on the results shown in Figure 2, we conclude that our model appears to work properly in the first example.

In the second simulated example, the original $\mathbf{X}_0$ was set equal to

$$\mathbf{X}_0(t_1, t_2) = 5(4t_1 - 2)\exp\left(-\left((4t_1 - 2)^2 + (4t_2 - 2)^2\right)\right) \tag{3.2}$$

and again we obtained standardized $\mathbf{X}$ and $\mathbf{Y} = \mathbf{A} + \mathbf{X} \circ \mathbf{B} + \mathbf{E}$. Again, we set $\mathbf{E}$'s elements to be i.i.d. $N(0, \sigma_e^2)$ with $\sigma_e/\sigma_y = 10\%$, where σ_y is the estimated standard deviation of the pixel values of image $\mathbf{Y}$. For this example, $\lambda_{\max}$ was 277.56, the estimated $\hat{\lambda}_{\text{opt}}$ was 2.08, the computing time required for estimating $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ was 26 seconds and for estimating the PSD of the estimated surfaces was 147 seconds. The R^2 was 0.9870. The results are shown in Figure 3. We note that R^2 for this example is still high and the estimated parameter surfaces are not close to the true parameter surfaces. Thus we need to rely on the PSD surfaces to judge the quality of estimated parameter surfaces.

For this example, we can see that the values of the two PSD surfaces in the four corner regions are large; so large, in fact, that we are led to question the overall quality of the estimated parameter surfaces (even without seeing the true parameter surfaces.) We are led to this conclusion because whenever we have a relatively flat $\mathbf{X}$ this flatness hampers the ability to estimate $\mathbf{B}$ well, somewhat analogous to the challenge of estimating a slope in simple linear regression when the range of the independent variable is small. Although the center regions of the PSD surfaces have small values, the overwhelming majority of the surfaces are large, and this leads us to question the overall fit.

Based on our empirical experience, a rough rule is that if the median of values in any PSD surface is ten times greater than the estimated noise standard deviation $\hat{\sigma}$, then the quality of the estimated parameter surfaces is in doubt. In fact, in Figure 3, the median of the values in the PSD surface of $\hat{\mathbf{B}}$ is $8.3\hat{\sigma}$ and the median of the values in the PSD surface of $\hat{\mathbf{A}}$ is $14.5\hat{\sigma}$. In Figure 2, the median of the values in the PSD surface of $\hat{\mathbf{B}}$ is $2.6\hat{\sigma}$ and the median of values in the PSD surface of $\hat{\mathbf{A}}$ is $2.7\hat{\sigma}$.

The only difference in the above two examples (apart from minor random noise) is the input image $\mathbf{X}$; the true parameter surfaces and signal to noise ratios are the same. It is clear that the condition of $\mathbf{X}$ can greatly affect the fit of the model. It is important, therefore, to compute R^2 and the PSD, point-wise standard deviation, to evaluate the fitness of the model.

In Figure 2, we notice that both estimated parameter surfaces have uneven areas in the middle, which also indicates that the fit in these areas is not good. As we have mentioned in the beginning of this section, We purposely set 0.53 as one of the boundaries for parameter surface $\mathbf{A}$. This creates a narrow strip with quite different combinations of values for the $\mathbf{A}$ and $\mathbf{B}$ surfaces. Our model identifies different combinations of values for surfaces $\mathbf{A}$ and $\mathbf{B}$, but if that combination exists in a region which cannot be cheaply recreated with wavelet bases, then our model will not be able to find this particular combination.

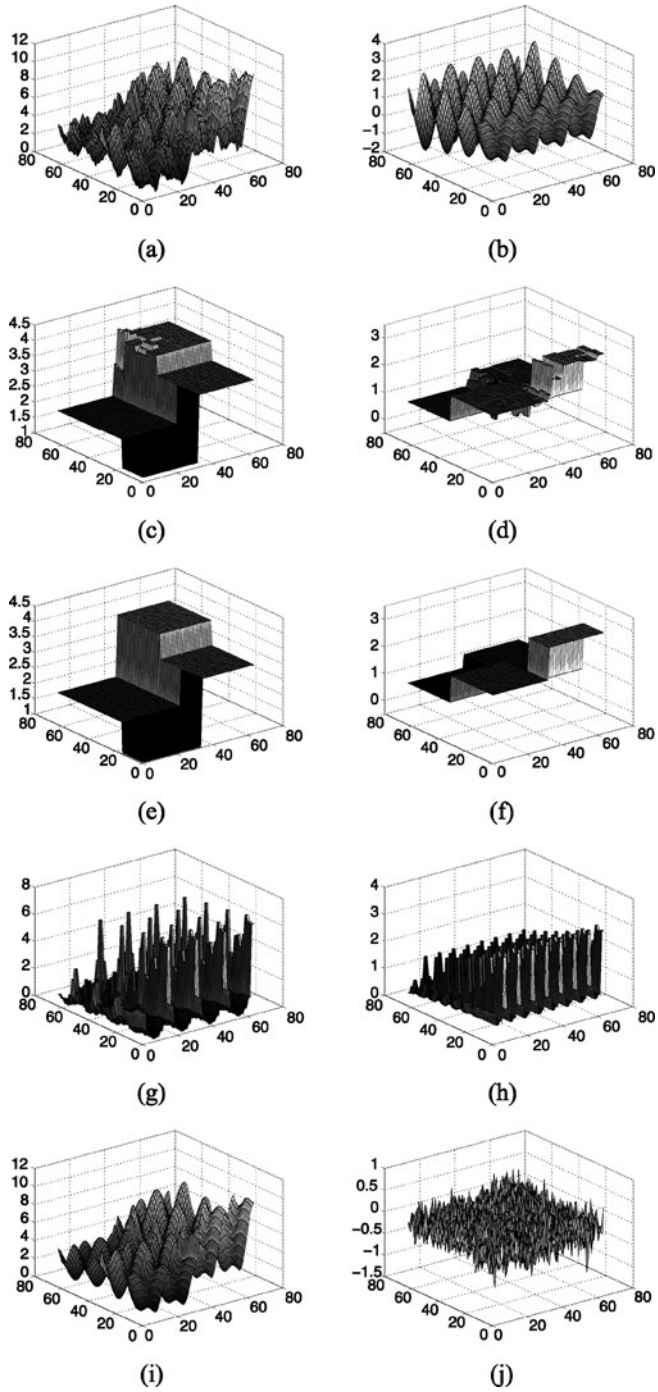

Figure 2. Figures for the first simulated example. Image size is 64×64. (a) Observed noisy response $\mathbf{Y}$, (b) covariate image $\mathbf{X}$, (c) estimated parameter surface $\hat{\mathbf{A}}$, (d) estimated parameter surface $\hat{\mathbf{B}}$, (e) true parameter surface $\mathbf{A}$, (f) true parameter surface $\mathbf{B}$, (g) the point-wise standard deviation of $\hat{\mathbf{A}}$, (h) the point-wise standard deviation of $\hat{\mathbf{B}}$, (i) estimated response $\hat{\mathbf{Y}}$, (j) residual plot. $\mathbf{Y} = \mathbf{A} + \mathbf{B} \circ \mathbf{X} + \mathbf{E}$ and noise standard deviation is equal to 10% of the standard deviation of pixel values of $\mathbf{Y}$.

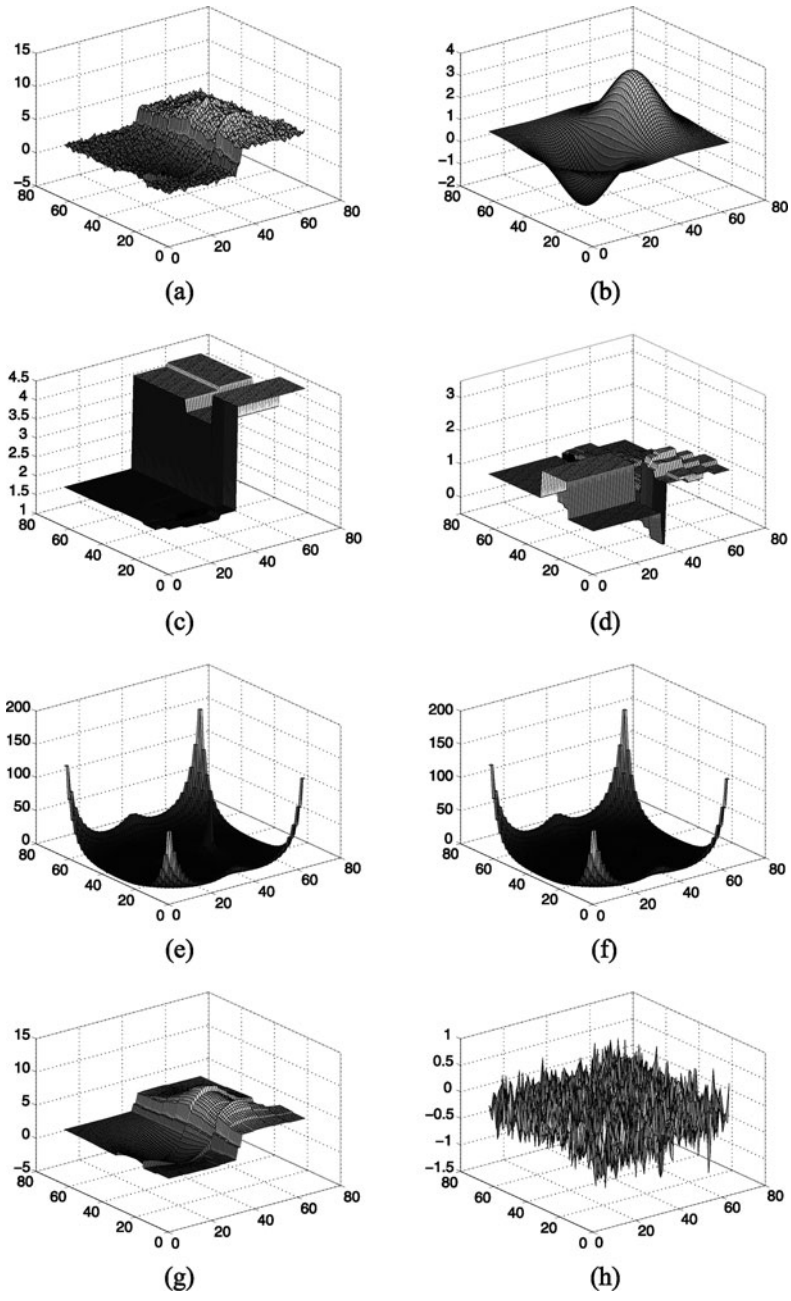

Figure 3. Figures for the second simulated example. Image size is 64×64. (a) Observed noisy response $\mathbf{Y}$, (b) covariate image $\mathbf{X}$, (c) estimated parameter surface $\hat{\mathbf{A}}$, (d) estimated parameter surface $\hat{\mathbf{B}}$, (e) the point-wise standard deviation of $\hat{\mathbf{A}}$, (f) the point-wise standard deviation of $\hat{\mathbf{B}}$, (g) estimated response $\hat{\mathbf{Y}}$, (h) residual plot. $\mathbf{Y} = \mathbf{A} + \mathbf{B} \circ \mathbf{X} + \mathbf{E}$ and noise standard deviation is equal to 10% of the standard deviation of pixel values of $\mathbf{Y}$.

Table 1. Scalability of the algorithm.

Image size	Non-zero coef.	Total coef.	λ_{opt}	Time1 (second)	Time2 (second)
64×64	42	2048	1.45	18	144
128×128	42	8192	1.97	113	1111
256×256	42	32728	1.94	721	5785

NOTES: Computing time measurements in column time1 are for surface estimation (second) and computing time measurements in column time2 are for PSD estimation (second). To estimate PSD for images greater than 64×64, we need to use some approximations.

One advantage of a wavelet expansion is its ability to capture multilevel properties of images under analysis. Small details like spikes can be represented with low level or fine wavelet bases and large details like trends can be represented with high level or coarse wavelet bases. Lacking high level or coarse wavelet bases will result in inefficiently using many low level or fine wavelet bases to represent large details in parameter surfaces, which means the true representation in the wavelet domain is no longer sparse. For example, we have mentioned that we have a total of 42 non-zero wavelet coefficients in our examples using six levels of Haar wavelet bases. If we only use one level of Haar wavelet bases (the finest level), then we will have 2048 non-zero wavelet coefficients for the true parameter surfaces. Using four levels of Haar wavelet bases will bring that number down to 60. Based on our experience, we suggest an empirical guideline for determining how many levels of wavelet bases we should use: we should make the support size of the largest wavelet bases at least a quarter of the original image for high Noise Signal Ratio (NSR) situations or at least one sixteenth of the original image for low NSR situations.

Next, we increase the size of our images to demonstrate the scalability of our algorithm. We use the $\mathbf{X}$ in (3.1) and the same noise matrix $\mathbf{E}$ with i.i.d. $N(0, \sigma_e^2)$ elements and we still set σ_e equal to 10% of σ_y. Again Haar wavelet bases are used in each case and the minimal support size is 4×4 and the maximal support size is equal to the current image size. We will increase the width and height of the images by two-fold each time. The results are listed in Table 1. The computing time measurements in column time1 are for parameter surface estimation and the computing time measurements in column time2 are for PSD estimation. To estimate PSD for images greater than 64×64, we need to use approximate matrices to save memory consumption, as noted in Section 2.3.

According to Kim et al. (2007), their large-scale l_1 penalized LSE will have computational time proportional to $O(n^{1.2})$ where n is the number of total coefficients. Based on Table 1, we estimate the computational complexity of our parameter estimation to be about $O(n^{1.3})$. The cost to estimate the PSD of the estimated surfaces is about 10 times the corresponding time spent on parameter surface estimation. So, if we were to continue to increase the size to 512×512, we would need approximately 3960 seconds to estimate the parameter surfaces and 39,600 seconds to estimate the PSD for parameter surfaces, although the computational times in Table 1 can be reduced with a faster computer. If we are not interested in the small scale features in large images, we can usually exclude multiple levels of fine wavelet bases and reduce the total number of parameters and computational

time significantly. Of course, if we want to extract the small scale details from a certain small region of the large images, we can restrict the input images to the small region and recalculate with more fine wavelet bases.

3.2. GYPSY MOTH DEFOLIATION EXAMPLE

We now return to the gypsy moth defoliation example and fit the model

$$\mathbf{Y} = \mathbf{A} + \mathbf{X}_1 \circ \mathbf{B}_1 + \mathbf{E} \tag{3.3}$$

where $\mathbf{Y}$ is standardized defoliation rate and $\mathbf{X}_1$ is standardized elevation. Here, the image size is 64×64 and Haar wavelet bases were used, with the smallest base support being 4×4 and the largest 64×64. For this example, $\lambda_{\max}$ was 65.3873, the estimated $\hat{\lambda}_{\text{opt}}$ was 0.8174, the computing time required for estimating $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}_1$ was 39 seconds and for estimating PSD of the estimated surfaces was 127 seconds. The R^2 was 0.8977. The results of the modeling are shown in Figure 4. The estimated parameter surface $\hat{\mathbf{B}}_1$ for elevation clearly shows a positive relation between defoliation rates and elevation. Low to high elevations in $\mathbf{X}_1$ usually match black to white tones in $\hat{\mathbf{B}}_1$. What is more important is that the spatial variation of the relationship between defoliation rate and elevation captured by the estimated parameter surfaces can be easily visualized and further inspected by the data analyst. For example, zones where the color tones in $\mathbf{X}_1$ and $\hat{\mathbf{B}}_1$ mismatch and may lead to the inclusion in the model of other factors of interest, such as the presence of unfavorable host tree species, nutrient availability, spraying with pesticides for insect suppression, etc.

We next inquire whether additional variables could be added to the model. Specifically, from Figure 4 we notice that in the lower-right corner of the defoliation rate image there is a V-shaped area with negative defoliation rates and high elevation which results in a similar V-shaped area in the estimated slope surface $\hat{\mathbf{B}}_1$. Inside this V-shaped area of $\hat{\mathbf{B}}_1$, the slope terms are smaller than those in the surrounding area and the elevation levels are higher than those in the surrounding area. Again the usual positive relationship between defoliation rate and elevation is reversed.

A possible explanation can be found by examining a species composition image (Foster and Townsend 2002). In Figure 5, the species composition image is a binary image, with white indicating forest types that are susceptible to gypsy moths and black indicating types less preferred by gypsy moths. By examining the species composition image we can clearly see that this V-shaped area in the defoliation rate image is similar to a V-shaped region corresponding to forest types susceptible to gypsy moth damage. Thus, we expand our model to be:

$$\mathbf{Y} = \mathbf{A} + \mathbf{X}_1 \circ \mathbf{B}_1 + \mathbf{X}_2 \circ \mathbf{B}_2 + \mathbf{E} \tag{3.4}$$

where $\mathbf{Y}$ is standardized defoliation rate, $\mathbf{X}_1$ is standardized elevation, and $\mathbf{X}_2$ is standardized binary species composition. Here, the image size is still 64×64 and Haar wavelet bases were used, with the smallest base support being 4×4 and the largest 64×64. For this example, $\lambda_{\max}$ was 65.3873, the estimated $\hat{\lambda}_{\text{opt}}$ was 0.8174, the computing time required for estimating $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}_1$ and $\hat{\mathbf{B}}_2$ was 51 seconds. The R^2 was 0.9074. The results of the modeling are shown in Figure 5. Note that, as noted in the previous section, because

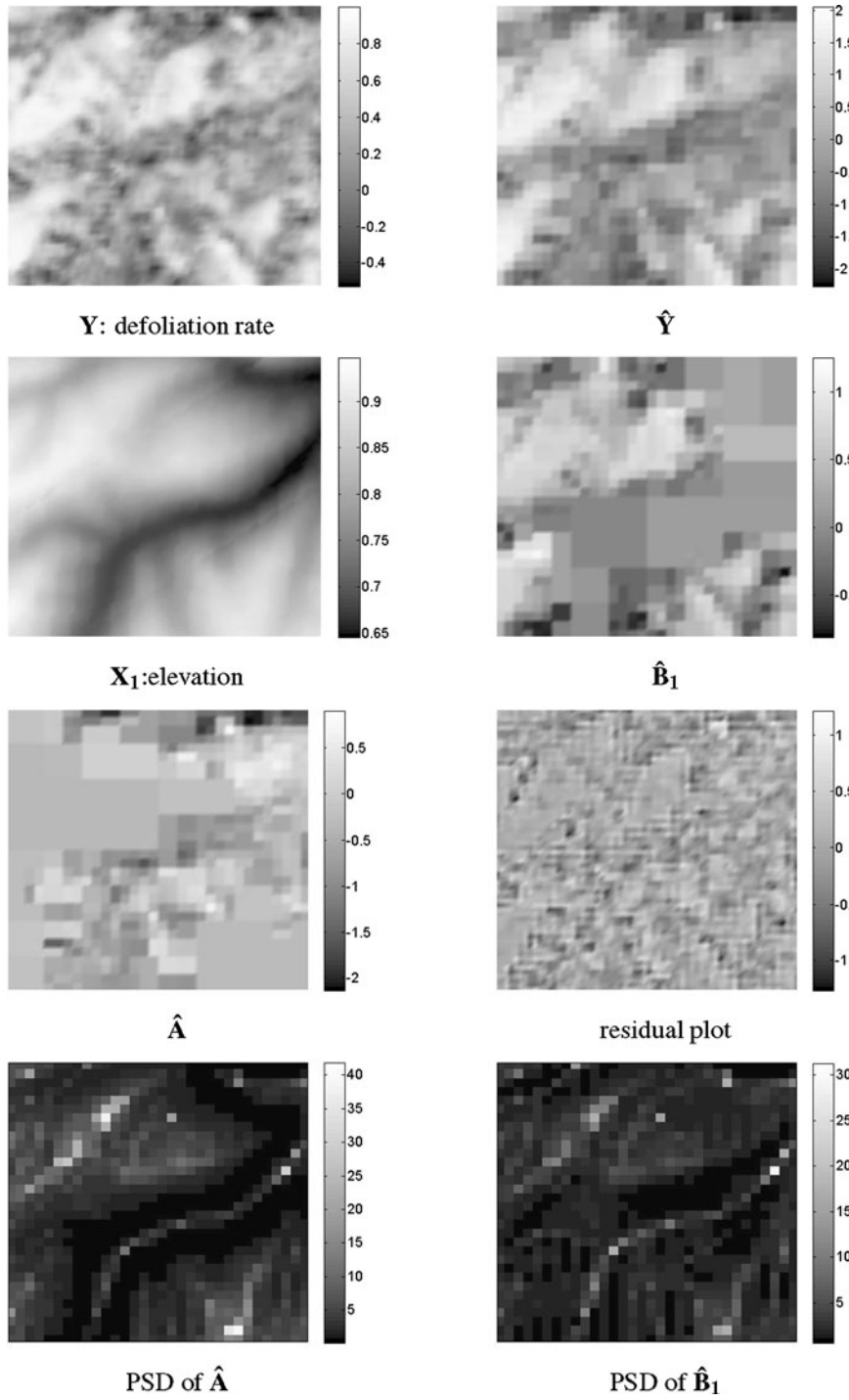

Figure 4. The model is $\mathbf{Y} = \mathbf{A} + \mathbf{X}_1 \circ \mathbf{B}_1 + \mathbf{E}$. Elevation is the explanatory variable $\mathbf{X}_1$. Defoliation rate is the response $\mathbf{Y}$.

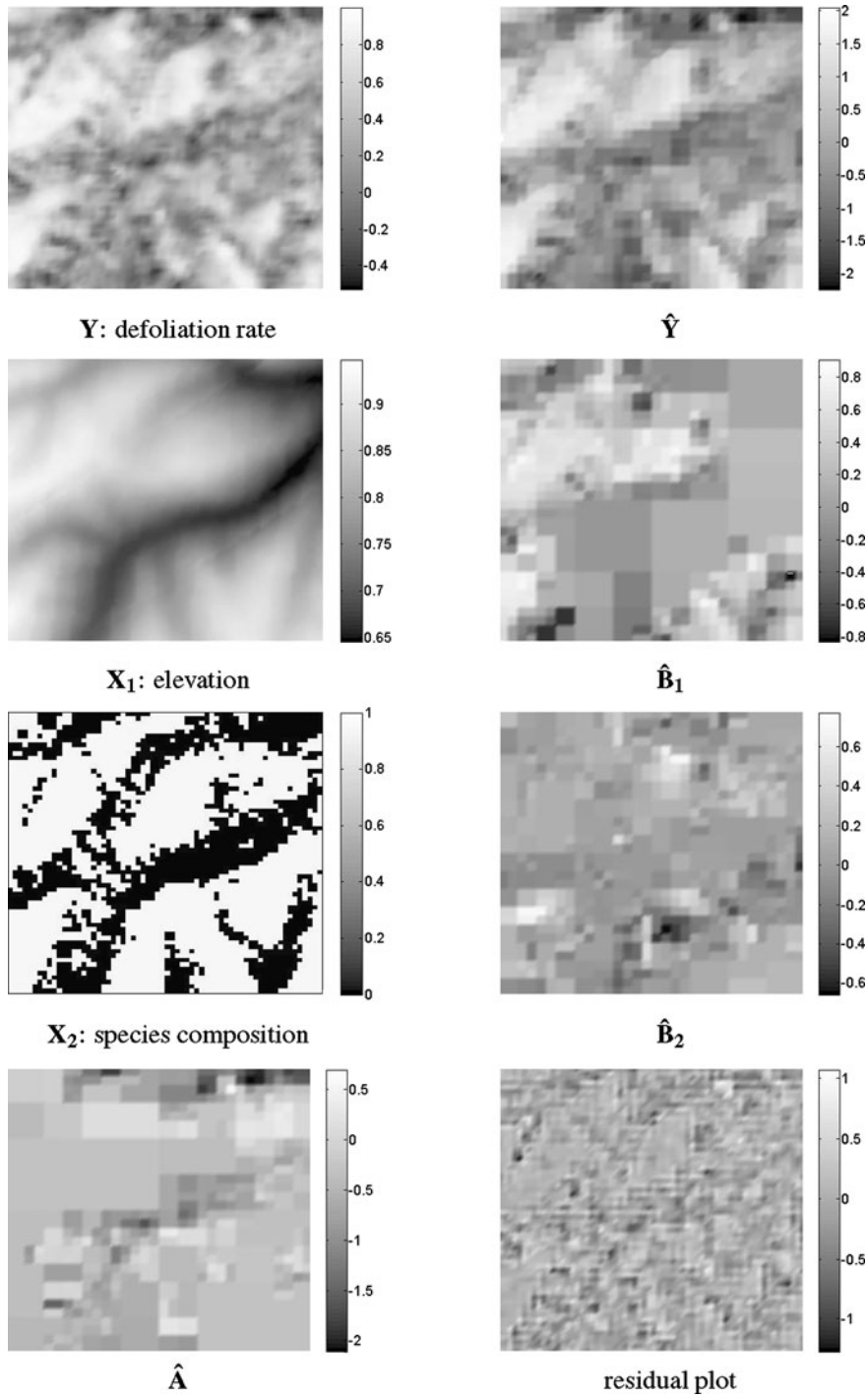

Figure 5. The model is $\mathbf{Y} = \mathbf{A} + \mathbf{X}_1 \circ \mathbf{B}_1 + \mathbf{X}_2 \circ \mathbf{B}_2 + \mathbf{E}$. Elevation is $\mathbf{X}_1$ and species composition is $\mathbf{X}_2$. Defoliation rate is the response $\mathbf{Y}$.

we have a binary image $\mathbf{X}_2$ in the model, Osborne, Presnell, and Turlach (2000)'s formula cannot be used to evaluate the covariance matrix of the LASSO estimator corresponding to a binary image like $\mathbf{X}_2$; too many of the columns of the design matrix are highly correlated and thus an LU, or approximate LU, decomposition will not work. Therefore, for the corresponding estimated parameter surface, we cannot determine a PSD using his method. This issue is discussed further in the next section.

Although the V-shaped corner does not completely disappear in $\hat{\mathbf{B}}_1$, the new estimated slope surface for elevation, it becomes less pronounced than before (e.g. the contrast in that part of the image is decreased). This occurs because $\mathbf{X}_2$, the species composition, explains some defoliation rates in that region. The contribution from each regressor can be measured with a partial R^2, namely

$$R_i^2 = 1 - \frac{\|\mathbf{Y} - \mathbf{X}_i \circ \hat{\mathbf{B}}_i\|_F^2}{\|\mathbf{Y} - \bar{\mathbf{Y}}\|_F^2}.$$

For elevation R_1^2 was 0.4831 and for species composition R_2^2 was 0.2379. So we can roughly say that overall the elevation contributes twice as much as the species composition does. The model in (3.4) could be called a concurrent Analysis of Covariance (ANCOVA) model. When we try to understand the estimated parameter surface $\hat{\mathbf{B}}_1$ for covariate $\mathbf{X}_1$, we should realize that $\hat{\mathbf{B}}_1$ now consists of the slope terms of elevations for two forest types. So if the regression of $\mathbf{Y}$ on elevation for each forest type is of interest, then the model in (3.4) is the one we should consider.

R^2 for the first model is 0.8977 and for the second model it is 0.9074, suggesting a slightly better fit. On the other hand, we have used more parameters in the second model, and so a higher R^2 might not be surprising. Regardless, examining the estimated defoliation rates $\hat{\mathbf{Y}}$ in Figures 5 shows that more details than $\hat{\mathbf{Y}}$ in Figures 4. Given the biologically sound reason for including $\mathbf{X}_2$, we are inclined to settle on the second model, although further study of other potential covariates remains warranted.

Thus far we have focused exclusively on Haar wavelets, but there is nothing in our methods that restrict the form of wavelets to this class. Indeed, it is reasonable to speculate that the lack of smoothness in Figures 4 and 5 results from the use of a Haar base, and if we were to choose a smoother base we might have different results. To briefly examine this, we close this section by applying Coiflets wavelet bases to the gypsy moth data with the model in (3.3). The Coiflets wavelet family has larger support than Haar wavelets do. They were first constructed by Daubechies at the request of Coifman. Coiflets are almost symmetric; a wavelet transform with Coiflets will keep the approximation subsignal close to the original signal (Walker 1999).

To proceed, we retain the image sizes of 64×64 and the images cover the same area of the Savage River State Forest as before. We use coif1 wavelet bases with the highest level being 3 and lowest level being 2. The smallest base support is 16×16 and the largest is 36×36. For this example, $\lambda_{\max}$ was 78.80, the estimated $\hat{\lambda}_{\text{opt}}$ was 0.5910, and the computing time required for estimating $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}_1$ was 190 seconds and for estimating PSD of the estimated surfaces it was 176 seconds. The R^2 was 0.9376, and as expected we find that the estimated parameter surfaces in Figure 6 are smoother than those in Figure 4 and

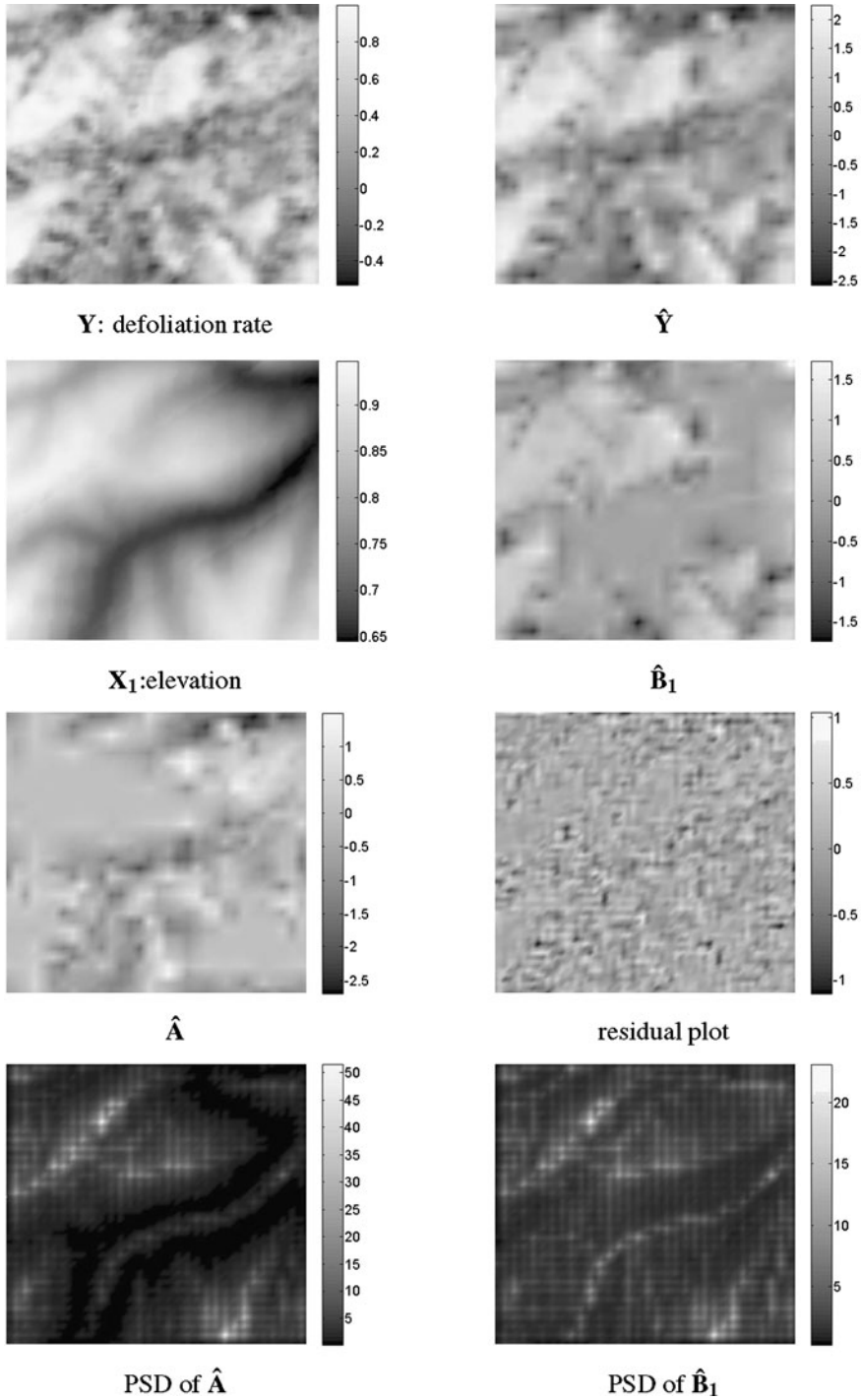

Figure 6. Coif1 wavelet bases are used and the model is $\mathbf{Y} = \mathbf{A} + \mathbf{X}_1 \circ \mathbf{B}_1 + \mathbf{E}$. Elevation is the explanatory variable. Defoliation rate is the response.

Figure 5. But the smoother results are obtained at a price: the computational time for parameter surface estimation is about five times as much as the time when Haar wavelet bases are used. Depending on the application, this might suggest a strategy wherein preliminary analyses are conducted using a Haar base to quickly gain a general view of the problem at hand, and then switching to some more complex base to ultimately obtain a smooth version of the parameter surface.

4. CONCLUSIONS AND FUTURE WORK

We have shown that general regression tools can be developed for image data. With examples, we have demonstrated that it is practical to fit a functional concurrent linear model with varying coefficients for images, and we have used wavelets to help constrain and model these coefficients. The potential applications of regression models for spatial images are not limited to remote sensing satellite images and can be extended to other types of images. For instance, Elad, Matalon, and Zibulevsky (2006) described a parallel matching pursuit algorithm in the wavelet domain for image denoising. Their model actually can be considered as a special case of the functional concurrent linear model $\mathbf{Y} = \mathbf{A} + \mathbf{B} \circ \mathbf{X} + \mathbf{E}$ if we set $\mathbf{X}$ to 0 and $\mathbf{Y}$ equal to the image to be denoised for the model.

Our work provides several unique contributions. First, it offers a systematic approach for building a quantitative model for describing the relationship among large images. This stands in contrast to the (more subjective) visual overlay methods frequently used in the analysis of remotely sensed data in a Geographic Information System. In contrast with a geostatistical approach, the use of wavelets provides a greater flexibility in modeling complex, spatially inhomogeneous relations. For example, in our numeric examples we show that functions which change rapidly in one region and are flat in another region can be handled easily. Second, in wavelet applications, the determination of which coefficients to retain and which to discard is performed by thresholding, which may or may not have a data-driven component. We address that by using an l_1 constrained least squares optimization with a penalty term λ, in essence providing a data-driven approach for thresholding that focuses on model fit. Finally, through the use of BIC, we propose a data-based approach to the selection of λ. The overall result is an approach that perhaps requires limited subjective assessment. Despite that, there remain several open questions, including the appropriate basis for conducting formal inference.

As noted above, it is not trivial to measure uncertainty when binary images are in the model. In addition, the exact effects of the image resolution on estimation is not covered in this paper. Extremely low resolution will certainly prevent us from obtaining good estimates. When the resolution increases, the situation becomes complicated. The signal to noise ratio, the conditions of the response and covariate images, the structures of the true parameter surfaces and whether the image domain is fixed or growing will affect the estimates. In Section 2.4, we utilize the resizing function in MATLAB when we have different image sizes. We are aware that other resizing algorithms exist and many new resizing algorithms are being developed (Candocia and Principe 1999). In the future, we plan to examine different resizing algorithms and study how to optimally choose a common image

size when resizing images with different initial sizes. We also intend to explore a Bayesian framework for our spatial implementation of a functional concurrent linear model. Smith and Fahrmeir (2007) showed that an Ising prior is efficient for spatial Bayesian variable selection. We plan to borrow from this notion and apply a hierarchical Ising prior on the wavelet coefficients and achieve a large-scale efficient Bayesian variable selection approach in the wavelet domain.

We close by noting that the functional concurrent linear model with varying coefficients is quite flexible. Not only can we include both continuous and discrete variables in the model, but also we can incorporate a neighborhood influence functional linear model into a functional concurrent linear model. One question is whether the complexity of the neighborhood influence model is needed, or whether the regular functional concurrent linear model is adequate. This might be the case because the concurrent model already is intended to account for local influence through the smoothing that takes place. If the neighborhood model is needed in some situations, then we will try to develop a test or diagnostic tool to determine when to choose it over the regular concurrent model.

ACKNOWLEDGEMENTS

We are grateful to Dr. Brenden McNeil for his assistance with the data preparation, and to two anonymous reviewers for their helpful suggestions and comments. This material was based upon work partially supported by the National Science Foundation under Grant DMS-0635449 to the Statistical and Applied Mathematical Sciences Institute. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

[Accepted March 2010. Published Online September 2010.]

REFERENCES

Addison, N. (2002), *The Illustrated Wavelet Transform Handbook*, London: Taylor & Francis.

Anselin, L., and Florax, R. J. G. M. (eds.) (1995), *New Directions in Spatial Econometrics*, Berlin: Springer.

Bell, B. S., and Broemeling, L. (2000), "A Bayesian Analysis for Spatial Processes With Application to Disease Mapping," *Statistics in Medicine*, 19, 957–974.

Besag, J. (1974), "Spatial Interaction and the Statistical Analysis of Lattice Systems (with discussion)," *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 36, 192–236.

Besag, J., and Kooperberg, C. (1995), "On Conditional and Intrinsic Autoregression," *Biometrika*, 82, 733–746.

Candocia, F. M., and Principe, J. C. (1999), "Superresolution of Images Based on Local Correlations," *IEEE Transactions on Neural Networks*, 10, 372–380.

Cressie, N. (1993), *Statistics for Spatial Data*, New York: Wiley-Interscience.

Donoho, D. L., and Johnstone, J. M. (1994), "Ideal Spatial Adaptation by Wavelet Shrinkage," *Biometrika*, 81, 425–455.

Donoho, D. L., Johnstone, I. M., Kerkyacharian, G., and Picard, D. (1995), "Wavelet Shrinkage: Asymptopia?" *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 57, 301–369.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression," *The Annals of Statistics*, 32, 494–499.

Elad, M., Matalon, B., and Zibulevsky, M. (2006), "Image Denoising with Shrinkage and Redundant Representations," June 17–22, 2006, CVPR.

Eshleman, K., Morgan, R., Webb, J., Deviney, F., and Galloway, J. (1998), "Temporal Patterns of Nitrogen Leakage From Mid-Appalachian Forested Watersheds: Role of Insect Defoliation," *Water Resources Research*, 34, 2005–2016.

Eubank, R., Huang, C., Maldonado, Y. M., and Buchanan, R. (2004), "Smoothing Spline Estimation in Varying-Coefficient Models," *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 66, 653–667.

Fan, J., and Li, R. (2001), "Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties," *Journal of the American Statistical Association*, 96, 1348–1360.

Faraway, J. J. (2000), "Modeling Reaching Motions Using Functional Regression Analysis," in *Digital Human Modeling for Design and Engineering, Conference and Exposition, Dearborn, Michigan*.

Fodor, I. K. (2002), "A Survey of Dimension Reduction Techniques," Technical report, US DOE Office of Scientific and Technical Information.

Foster, J., and Townsend, P. (2002), "Mapping Forest Composition in the Central Appalachians Using AVIRIS: Effects of Topography and Phenology," in *Proceedings of the Eleventh JPL Airborne Earth Science Workshop*, ed. R. O. Green, Pasadena, CA: Jet Propulsion Laboratory.

Gelfand, A., and Vounatson, P. (2003), "Proper Multivariate Conditional Autoregressive Models for Spatial Data Analysis," *Biostatistics*, 4, 11–25.

Gelfand, A. E., Kim, H.-J., Sirmans, C., and Banerjee, S. (2003), "Spatial Modeling With Spatially Varying Coefficient Processes," *Journal of the American Statistical Association*, 98, 387–396.

Goutis, C. (1998), "Second-Derivative Functional Regression With Applications to Near Infrared Spectroscopy," *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 60, 103–114.

Hastie, T., Buja, A., and Tibshirani, R. (1995), "Penalized Discriminant Analysis," *The Annals of Statistics*, 33, 73–102.

Hastie, T., and Tibshirani, R. (1993), "Varying-Coefficient Models," *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 55, 757–796.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning*, Berlin: Springer.

Houston, D., and Valentine, H. (1977), "Comparing and Predicting Forest Stand Susceptibility to Gypsy Moth," *Canadian Journal of Forest Research*, 7, 447–461.

Kaufman, C. G., and Sain, S. R. (2007), "Bayesian Functional ANOVA Modeling Using Gaussian Process Prior Distributions," Preprint.

Kim, S.-J., Koh, K., Lustig, M., Boyd, S. P., and Gorinevsky, D. (2007), "An Interior-Point Method for Large-Scale l_1-Regularized Least Squares," *IEEE Journal on Selected Topics in Signal Processing*, 1, 606–617.

Kleiner, K., and Montgomery, M. (1994), "Forest Stand Susceptibility to the Gypsy-Moth (Lepidoptera, Lymantriidae)—Species and Site Effects on Foliage Quality to Larvae," *Environmental Entomology*, 23, 699–711.

Kyung, M., Gill, J., Ghosh, M., and Casella, G. (2009), "Penalized Regression, Standard Errors, and Bayesian Lassos," Online Manuscript.

Lovett, G. M., Christenson, L. M., Groffman, P. M., Jones, C. G., Hart, J. E., and Mitchell, M. J. (2002), "Insect Defoliation and Nitrogen Cycling in Forests," *Bioscience*, 52, 335–341.

Mallat, S. G., and Zhang, Z. (1993), "Matching Pursuits with Time-Frequency Dictionaries," *IEEE Transactions on Signal Processing*, 41, 3397–3415. See also IEEE Transactions on Acoustics, Speech, and Signal Processing.

McNeil, B., de Beurs, K., Eshleman, K., Foster, J., and Townsend, P. (2007), "Maintenance of Ecosystem Nitrogen Limitation by Ephemeral Forest Disturbance: An Assessment Using Modis, Hyperion, and Landsat ETM," *Geophysical Research Letters*, 34, 73–102.

Netravali, A. N., and Haskell, B. G. (1995), *Digital Pictures: Representation, Compression and Standards* (2nd ed.), New York: Plenum.

Osborne, M. R., Presnell, B., and Turlach, B. A. (2000), "On the LASSO and Its Dual," *Journal of Computational and Graphical Statistics*, 9, 319–337.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. (1993), "Orthogonal Matching Pursuit: Recursive Function Approximation With Applications to Wavelet Decomposition," in *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, Vol. 1, pp. 40–44.

Pötscher, B. M., and Leeb, H. (2007), "On the Distribution of Penalized Maximum Likelihood Estimators: The LASSO, SCAD, and Thresholding," MPRA Paper 5615, University Library of Munich, Germany, URL *http://ideas.repec.org/p/pra/mprapa/5615.html*.

Ramsay, J., and Silverman, B. (2005), *Functional Data Analysis* (2nd ed.), Berlin: Springer.

Ratcliffe, S., Leader, L., and Heller, G. (2002), "Functional Data Analysis With Application to Periodically Stimulated Foetal Heart Rate Data. I: Functional Regression," *Statistics in Medicine*, 21, 1103–1114.

Smith, M., and Fahrmeir, L. (2007), "Spatial Bayesian Variable Selection With Application to Functional Magnetic Resonance Imaging," *Journal of the American Statistical Association*, 102, 417–431.

Stern, H., and Cressie, N. (2000), "Posterior Predictive Model Checks for Disease Mapping Models," *Statistics in Medicine*, 19, 2377–2397.

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the LASSO," *The Annals of Statistics*, 58, 267–288.

Townsend, P. A., Eshleman, K. N., and Welcker, C. (2004), "Remote Sensing of Gypsy Moth Defoliation to Assess Variations in Stream Nitrogen Concentrations," *Ecological Applications*, 14, 504–516.

Tropp, J. A., and Gilbert, A. C. (2007), "Signal Recovery From Random Measurements via Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory*, 53, 4655–4666.

Unser, M., Aldroubi, A., and Eden, M. (1991), "Fast b-Spline Transforms for Continuous Image Representation and Interpolation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, 277–285.

Wahba, G., and Cressie, N. (1990), "Comments on Cressie and reply to Wahba," *American Statistician*, 44, 255–258.

Walker, J. S. (1999), *A Primer on Wavelets and Their Scientific Applications*, Boca Raton: CRC Press.

Wall, M. M. (2004), "A Close Look at the Spatial Structure Implied by the Car and Sar Models," *Journal of Statistical Planning and Inference*, 121.

West, M., Harrison, P., and Migon, H. (1985), "Dynamic Generalized Linear Models and Bayesian Forecasting," *Journal of the American Statistical Association*, 80, 73–83.

Yamanishi, Y., and Tanaka, Y. (2003), "Geographically Weighted Functional Multiple Regression Analysis: A Numerical Investigation," *Journal of the Japanese Society of Computational Statistics*, 15, 307–317.

Zou, H. (2006), "The Adaptive LASSO and Its Oracle Properties," *Journal of the American Statistical Association*, 101, 1418–1429.

Zou, H., Hastie, T., and Tibshirani, R. (2007), "On the Degrees of Freedom of the LASSO," *The Annals of Statistics*, 35, 2173–2192.