Journal of Statistical Software

July 2011, Volume 43, Issue 3. *http://www.jstatsoft.org/*

LS2W: Implementing the Locally Stationary 2D Wavelet Process Approach in R

Idris A. Eckley
Lancaster University

Guy P. Nason
University of Bristol

Abstract

Locally stationary process representations have recently been proposed and applied to both time series and image analysis applications. This article describes an implementation of the locally stationary two-dimensional wavelet process approach in R. This package permits construction of estimates of spatially localized spectra and localized autocovariance which can be used to characterize structure within images.

Keywords: random field, locally stationary, local autocovariance, LS2W, texture analysis, non-decimated wavelets, R.

1. Introduction

Wavelet techniques have established themselves in the statistics literature as a popular approach for nonparametric curve estimation and for the modelling, analysis and forecasting of time series. Given the (typically) inherent non-stationary nature of time series, the modelling of locally stationary processes has received increased attention in recent years. Nason, von Sachs, and Kroisandt (2000, henceforth referred to as NvSK) provide an early contribution to this field, introducing a class of locally stationary wavelet (LSW) processes together with an associated estimation scheme. Other recent contributions include Fryzlewicz and Nason (2006), who propose a new technique for estimating the evolutionary wavelet spectrum based on the Haar-Fisz transform. Van Bellegem and von Sachs (2008) extend the class of LSW processes to contain processes whose spectral density can change suddenly in time. Van Bellegem and Dahlhaus (2006) propose an estimation approach to fit time-varying models to an alternative clasfigs of locally stationary processes using the Whittle likelihood, whilst Dahlhaus and Polonik (2009) apply empirical process methodology to locally stationary time series processes.

This article introduces the **LS2W** package which provides an implementation of the modelling approach proposed by Eckley, Nason, and Treloar (2010) for locally stationary spatial covariance structure of (regular) lattice processes, such as images. The approach, which is an extension of NvSK's time series modelling paradigm, is in part motivated by a frequently highlighted aspect of image texture, namely that texture possesses structure on many different scales. Moreover, several researchers have highlighted that the human and mammalian visual systems process images in a multiscale manner, preserving both local and global information (see for example Daugman 1990 or Field 1999). Thus there is a compelling argument for the development of a multiscale model of image texture as proposed by Eckley *et al.* (2010).

The **LS2W** package has been developed in R (R Development Core Team 2011) and makes use of several functions within the **wavethresh** package developed by Nason (2010). Both packages are available from the Comprehensive R Archive Network (CRAN) at `http://CRAN.R-project.org/`.

Our article is organized as follows: Section 2 provides a brief introduction to wavelets, paying particular attention to the discrete wavelets which form the central building block of the locally stationary two-dimensional wavelet (LS2W) approach. The modelling approach is introduced in Sections 3 and Section 4 provides a tutorial on the **LS2W** package whilst describing various elements of the estimation scheme established by Eckley *et al.* (2010). Finally a case study involving texture discrimination is provided in Section 5.

2. A brief introduction to wavelets

Wavelet research has undergone a rapid expansion in terms of volume and complexity in the last 20 years although the basic concepts are extremely simple. We provide a very brief introduction to wavelets here. Those interested in learning more can consult any number of useful texts including Daubechies (1992), Mallat (1999), Vidakovic (1999) or Nason (2008).

For functions, $f(x)$, on the real line, the idea is that there exists a basis of functions, $\{\psi_{jk}(x)\}_{j,k\in\mathbb{Z}}$, so that

$$f(x) = \sum_{j,k} d_{j,k}\psi_{jk}(x).$$

If the basis functions are orthogonal then the coefficients can be obtained from the function f in the usual way, i.e.,

$$d_{jk} = \int f(x)\psi_{jk}(x)\,dx, \qquad (1)$$

where $j, k \in \mathbb{Z}$. Wavelets provide a multiscale analysis of functions, f, because the basis functions are all scalings of one function, $\psi(x)$, called the *mother wavelet*. Such scalings are defined by the following relation:

$$\psi_{jk}(x) = 2^{j/2}\psi(2^j x - k),$$

for all $x \in \mathbb{R}$ and $j, k \in \mathbb{Z}$. The mother wavelet is chosen to have several important properties such as fast decay (often compact support), oscillation, zero integral and not least that the wavelets form an (orthonormal) basis. With this construction it can be seen that the coefficient d_{jk} in (1) conveys information about the function f at a scale proportional to 2^j and location near to $2^{-j}k$. In other words, the set of wavelet coefficients of a function characterize the function's behavior at different scales and locations.

As well as a mother wavelet one can define a father wavelet, $\phi(x)$, which is similar to a kernel function such as that used in kernel density estimation or regression. Whilst wavelet coefficients provide information about the local oscillatory behavior of a function the father coefficients encapsulate information about the multiscale mean behavior of that function. The father wavelets satisfy a multiscale relation, called the dilation equation:

$$\phi(x) = \sum_k h_k \phi(2x - k),$$

for all $x \in \mathbb{R}$. The mother wavelet $\psi(x)$ satisfies a similar equation:

$$\psi(x) = \sum_k g_k \psi(2x - k) \ \forall x \in \mathbb{R}.$$

More details and illustrations of various wavelets can be found in Daubechies (1992), Mallat (1999), Vidakovic (1999) or Nason (2008). The whole set of mother/father wavelets can be characterized by a suitable choice of h_k and g_k. There are many families of wavelets but a particularly useful and famous set, compactly supported, was developed by Daubechies (1988) which we will use extensively here. Daubechies' construction produced sets of h_k, g_k known as quadrature mirror filters.

The relationship between the mother and father wavelets lies as the heart of the wavelet transform. For a thorough discussion we refer the reader to Nason (2008). Of particular relevance to the work in this paper, both the mother and father wavelet are required to construct the (separable) two-dimensional wavelet transform which we use as the basic building block in our locally stationary modelling framework. In the next section we describe this two-dimensional wavelet transform.

2.1. Discrete wavelets

In many real-life situations data appear not as continuous mathematical functions but discrete sequences. Hence, a discrete wavelet transform, with "discrete wavelets" exists to provide discrete wavelet coefficients. One of the appealing features of such a transform is the existence of an extremely fast and efficient transform algorithm, such as that proposed by Mallat (1989). Mallat's discrete wavelet transform achieves a computational effort which is linear in the number of data points.

Let $\{h_k\}$ and $\{g_k\}$ be the low and high pass quadrature mirror filters that are used in the construction of the Daubechies (1992) compactly supported continuous time wavelets. NvSK define the associated *discrete father wavelet filters*, $\phi_j = (\phi_{j,0}, \ldots, \phi_{j,(L_j-1)})$ which are vectors of length L_j for integer scales $j > 0$ and obtained using the formulae:

$$\phi_{1,n} = h_n, \quad \text{for } n = 0, \ldots, L_1 - 1, \tag{2}$$

$$\phi_{(j+1),n} = \sum_k h_{n-2k} \phi_{j,k}, \quad \text{for } n = 0, \ldots, L_{j+1} - 1, \tag{3}$$

where $L_j = (2^j - 1)(N_h - 1) + 1$, N_h is the number of non-zero elements of $\{h_k\}$ and δ_{0k} is the Kronecker delta. This definition for discrete father wavelet filters is essentially the same as for mother wavelets, ψ_j, given in NvSK, except that the h_{n-2k} in (2) is replaced by g_{n-2k}. Also, here we adopt *positive* scales rather than use the notationally inconvenient

$j < 0$ as in NvSK. As an example, the discrete Haar father wavelet filters at scales 1 and 2 are $\phi_1 = (h_0, h_1) = (1/\sqrt{2})(1, 1)$ and $\phi_2 = (h_0^2, h_1 h_0, h_0 h_1, h_1^2) = \frac{1}{2}(1, 1, 1, 1)$.

Two-dimensional discrete father wavelet filters can be formed from one-dimensional mother *and* father wavelet filters by a tensor product operation as follows.

Definition 1. *Let* $\mathbf{k} = (k_1, k_2)$ *where* $k_1, k_2 \in \mathbb{Z}$. *We define the* two-dimensional discrete wavelet filters, $\{\psi_j^l\}$, *as finite matrices of dimension* L_j^2, *as follows:*

$$\psi_j^l = \left[\begin{array}{ccc} \psi^l_{j,(0,0)} & \cdots & \psi^l_{j,(0,L_j-1)} \\ \vdots & \vdots & \vdots \\ \psi^l_{j,(L_j-1,0)} & \cdots & \psi^l_{j,(L_j-1,L_j-1)} \end{array} \right] \textit{for } l = h, v \textit{ or } d.$$

Here h, v *and* d *denote the horizontal, vertical and diagonal directions. The elements of the matrix are defined by*

$$\left. \begin{array}{rl} \psi^h_{j,\mathbf{k}} &= \phi_{j,k_1}\psi_{j,k_2} \\ \psi^v_{j,\mathbf{k}} &= \psi_{j,k_1}\phi_{j,k_2} \\ \textit{and} \quad \psi^d_{j,\mathbf{k}} &= \psi_{j,k_1}\psi_{j,k_2} \end{array} \right\} \textit{for } k_1, k_2 = 0, \ldots, L_j - 1, \tag{4}$$

where $\psi_{j,k}$ *and* $\phi_{j,k}$ *are the one-dimensional discrete wavelets. Similarly, the two-dimensional discrete father wavelet filter is defined as:*

$$\phi_{j,\mathbf{k}} = \phi_{j,k_1}\phi_{j,k_2}.$$

2.2. Example: Discrete Haar wavelets

The finest scale discrete Haar wavelet in the diagonal decomposition direction is given by:

$$\psi_1^d = \psi_1 \otimes \psi_1 = \frac{1}{2}\left[\begin{array}{cc} 1 & -1 \\ -1 & 1 \end{array} \right];$$

where $\otimes$ is the tensor product. The second finest scale discrete Haar wavelet in the horizontal direction is given by:

$$\psi_2^h = \phi_2 \otimes \psi_2 = \frac{1}{4}\left[\begin{array}{cccc} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \end{array} \right].$$

For convenience, we will henceforth refer to these discrete wavelet filters as simply discrete wavelets.

As with the one-dimensional case, considered by NvSK, we can form the collection of *non-decimated* discrete wavelets by translations as follows:

$$\psi^l_{j,\mathbf{u}}(\mathbf{r}) = \psi^l_{j,\mathbf{u}-\mathbf{r}}$$

for every scale j, direction l and locations $\mathbf{u}, \mathbf{r} \in \mathbb{Z}^2$. Thus with two-dimensional non-decimated wavelets there is a wavelet at each scale, direction *and* location. For further details on non-decimated wavelets and their associated transform see Nason and Silverman (1995) or Nason (2008).

3. The LS2W approach

3.1. Motivation

Suppose that we have a random field which is defined on a regular grid, $\{X_{\mathbf{r}}\}_{\mathbf{r}\in\mathbb{Z}^2}$ for which we wish to estimate the covariance $\mathsf{COV}(X_{\mathbf{r}}, X_{\mathbf{s}}) = \gamma_{\mathbf{r},\mathbf{s}}$ where $\mathbf{r},\mathbf{s} \in \mathbb{Z}^2$. The covariance structure of such a field could take many possible forms. For example, the process could be (second-order) stationary, or intrinsically stationary (see Priestley 1981 or Cressie 1991) or in extreme cases the covariance could possess minimal controls such as $\gamma_{\mathbf{r},\mathbf{s}} = \gamma_{\mathbf{t},\mathbf{u}}$ if and only if $\mathbf{r} = \mathbf{t}$ and $\mathbf{s} = \mathbf{u}$. This last case permits the covariance to change from location to location in a highly nonstationary manner. Clearly, statistical estimation in this situation is troublesome.

The form of covariance structure that Eckley *et al.* (2010) consider lies somewhere between the two extremes of stationarity and the highly nonstationary form. They permit the covariance structure to change slowly as a function of location and hence the covariance structure around a particular location, $\mathbf{r}$, may be estimated by pooling information from data close to $\mathbf{r}$. Fields which exhibit this slowly varying structure are termed *locally stationary random fields*. Since many real-life images have a locally stationary structure operating at several different scales, Eckley *et al.* (2010) adopt wavelets as the basic building block for such locally stationary processes.

3.2. The model

Armed with the non-decimated discrete wavelet building blocks of Section 2.1, we now introduce the wavelet model for random fields proposed by Eckley *et al.* (2010).

Definition 2. *Let* $\mathbf{R} = (R, S)$ *where* $R = 2^m, S = 2^n \geq 1$ *for* $m, n \in \mathbb{N}$ *and set* $J(R,S) \equiv \log_2\{\min(R,S)\}$. *Further, let* $\mathbf{r} = (r,s)$ *and* $\mathbf{u} = (u,v)$ *for* $\mathbf{r},\mathbf{u} \in [0, R-1] \times [0, S-1]$. *Then a class of* locally stationary two-dimensional wavelet processes *(LS2W) is defined to be a sequence of stochastic processes which lie on a regular grid denoted by*

$$\{X_{\mathbf{r};\mathbf{R}}\}_{\mathbf{r}\in[0,R-1]\times[0,S-1]}.$$

Such processes have the following representation in the mean-square sense:

$$X_{\mathbf{r};\mathbf{R}} = \sum_{l} \sum_{j=1}^{\infty} \sum_{\mathbf{u}} w^l_{j,\mathbf{u};\mathbf{R}} \psi^l_{j,\mathbf{u}}(\mathbf{r}) \xi^l_{j,\mathbf{u}}, \tag{5}$$

where the sum over l *is over decomposition directions* v, h *and* d, $\{\psi^l_{j,\mathbf{u}}(\mathbf{r})\}$ *is a collection of discrete non-decimated two-dimensional wavelets and* $\xi^l_{j,\mathbf{u}}$ *is a mean zero, random orthonormal increment process satisfying* $\mathsf{E}\left(\xi^l_{j,\mathbf{k}}\xi^p_{m,\mathbf{n}}\right) = \delta_{j,m}\delta_{\mathbf{k},\mathbf{n}}\delta_{l,p}$, *(i.e., is uncorrelated).*

Eckley *et al.* (2010) also impose certain technical assumptions on the LS2W processes. These provide control mechanisms over the degree of non-stationary structure allowed within such models, imposing zero mean and the finiteness of the model.

3.3. Example: Haar moving average fields

As an example we use Haar wavelets to construct LS2W fields. We begin by defining the canonical set of fields from which more complicated processes can be constructed.

Definition 3. *Let $c \in \mathbb{R}$ and $\mathbf{r} = (r, s)$. We define a Haar MA field of order j_0, in direction l_0, to be the LS2W process $X_{\mathbf{r}}^{j_0,l_0}$ generated by the Haar 2D non-decimated discrete wavelets with the following condition on the amplitude:*

$$w_{j,\mathbf{u}}^{l} = \begin{cases} c & \textit{for } j = j_0, l = l_0, \\ 0 & \textit{otherwise.} \end{cases}$$

For example, setting $c = \sigma$ in the definition for $j_0 = 1$ and $l_0 = d$ (diagonal direction), using Haar wavelets and setting the orthonormal increment sequence $\xi_{1,\mathbf{u}}^{d} = \epsilon_{\mathbf{u}}$ where $\{\epsilon_{\mathbf{u}}\}$ is a purely random process with mean zero and variance 1 gives

$$X_{\mathbf{r}}^{1,d} = \sigma \sum_{\mathbf{u}} \psi_{1,\mathbf{u}-\mathbf{r}}^{d} \xi_{1,\mathbf{u}}^{d} = \sigma(\epsilon_{r,s} - \epsilon_{r,s+1} - \epsilon_{r+1,s} + \epsilon_{r+1,s+1})/2. \tag{6}$$

Haar MA fields are special cases of the moving average fields due to Haining (1978) – see Moore 1988 or Cressie 1991 for further details.

A realization of the $X_{\mathbf{r}}^{1,d}$ Haar moving average field is shown in Figure 1 (left). This can be generated using the **LS2W** function `Haar2MA.diag` as follows:

```
R> Haar2MA.diag(n = 512, sd = 1, order = 1)
```

The inputs to this function are

- `n`: This determines the size of the simulated image (512 × 512).
- `sd`: The standard deviation of the orthonormal sequence which underlies the process.
- `order`: This defines the order of the two-dimensional moving average process to be simulated.

As the realization is constructed from the finest scale wavelets the texture detail is fine and also has a "diagonal" orientation because of the direction of the underlying wavelets. Figure 1 (right) shows broader detail from scale $j_0 = 3$ horizontal ($l_0 = h$) and scale $j_0 = 2$ vertical wavelets ($l_0 = v$): a realization from the addition of two Haar MA fields with $w_{3,\mathbf{u}}^{h} = w_{2,\mathbf{u}}^{v} = \sigma$ (with all other w zero). This realization was generated by combining the output obtained from the `Haar2MA.vert` and `Haar2MA.horiz` functions, which simulate realizations containing vertical and horizontal orientations respectively.

3.4. Local wavelet spectra

Recall from time series analysis, that if a series $\{X_t\}_{t\in\mathbb{Z}}$ is a stationary stochastic process, then it admits the following representation:

$$X_t = \int_{-\pi}^{\pi} A(\omega) \exp(i\omega t) d\xi(\omega), \tag{7}$$

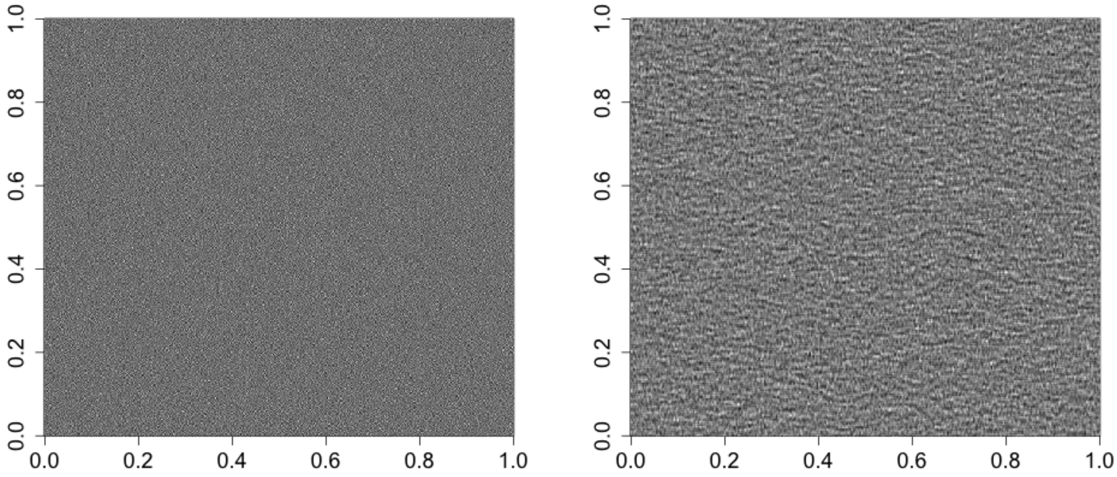


Figure 1: Haar moving average fields. Left: 2D Haar moving average field of order 1 with diagonal detail only. Right: A LS2W process with contributions at scales 2 and 3 in the vertical and horizontal directions respectively.

where $A(\omega)$ denotes the amplitude of the process and $d\xi(\omega)$ is an orthonormal increments process (see Priestley 1981 for further details). Here the amplitude $A(\omega)$ controls the magnitude of the sinusoidal oscillation at frequency ω. The traditional way of presenting the amplitude is the spectrum: $f(\omega) = |A(\omega)|^2$. Equivalent structures also exist in the stationary random field setting. It is natural therefore to consider what the LS2W analogue of such a quantity might be.

To this end, we introduce the *local wavelet spectrum* for a LS2W process $X_{\mathbf{r}}$:

$$S_j^l(\mathbf{z}) = S_j^l(z_1, z_2) = \lim_{\min\{R,S\}\to\infty} |w^l_{j,[z_1R],[z_2S]}|^2 \qquad (8)$$

for $\mathbf{z} \in (0,1)^2$, $j \in 1,\ldots,J$, and $l \in \{h, v \text{ or } d\}$.

In other words, the LWS delivers a location-direction/scale decomposition of structure (i.e., variance) within a LS2W process. Note that in this definition we use a quantity $\mathbf{z}$, known as *rescaled location*. Such rescaling is common within the locally stationary literature. For further details in the locally stationary Fourier, locally stationary wavelet or LS2W setting we refer the reader to Dahlhaus (1997), Nason *et al.* (2000) or Eckley (2001) respectively.

3.5. Example

Let us reconsider the stationary, Haar MA field, example considered in Section 3.3. For a process of the form $X_{\mathbf{r}}^{1,d}$, it is easily verified that the LWS takes the form:

$$\begin{aligned} S_j^l(\mathbf{z}) &= |w^l_{j,([z_1R],[z_2S])}|^2; \\ &= \begin{cases} \sigma^2 & \text{for } j = 1 \text{ and } l = d \\ 0 & \text{otherwise;} \end{cases} \\ &= \sigma^2\delta_{j,1}\delta_{l,d} \qquad \forall\ \mathbf{z} \in (0,1)^2. \end{aligned}$$

3.6. Local autocovariance and autocorrelation wavelets

For stationary processes it is well-known that the covariance is the Fourier transform of the spectrum. For completeness, we highlight that an equivalent relationship also exists in the LS2W setting. Prior to this we introduce autocorrelation wavelets. These will prove useful in defining the wavelet-based local covariance.

We define two-dimensional autocorrelation wavelets as follows:

Definition 4. *Let $j \in \mathbb{N}$, $l \in \{v, h, d\}$ and $\boldsymbol{\tau} \in \mathbb{Z}^2$. Then the* autocorrelation (ac) wavelet *of a 2D discrete wavelet family $\{\psi^l_{j,\mathbf{k}}\}$ is given by*

$$\Psi^l_j(\boldsymbol{\tau}) = \sum_{\mathbf{v}\in\mathbb{Z}^2} \psi^l_{j,\mathbf{v}}(\mathbf{0})\psi^l_{j,\mathbf{v}}(\boldsymbol{\tau}). \tag{9}$$

Note that the 2D ac wavelets inherit a separable form from the discrete wavelets of equation (4), i.e., in the horizontal, vertical and diagonal directions:

$$\Psi^h_j(\boldsymbol{\tau}) = \Phi_j(\tau_1)\Psi_j(\tau_2),\ \Psi^v_j(\boldsymbol{\tau}) = \Psi_j(\tau_1)\Phi_j(\tau_2),\ \Psi^d_j(\boldsymbol{\tau}) = \Psi_j(\tau_1)\Psi_j(\tau_2), \tag{10}$$

where $\boldsymbol{\tau} = (\tau_1, \tau_2)$, and Ψ_j, Φ_j are the 1D discrete ac wavelet and father wavelets from NvSK.

One-dimensional discrete ac wavelets can be generated using the **LS2W** package by calling the `PsiJ` and `PhiJ` functions as appropriate. For example

```
R> MyPhi <- PhiJ(J = -4, filter.number = 1, family = "DaubExPhase")
R> plot(MyPhi[[4]], xlab = "Index", ylab = "Haar AC Scaling function")
R> MyPsi <- PsiJ(J = -4, filter.number = 1, family = "DaubExPhase")
R> plot(MyPsi[[4]], xlab = "Index", ylab = "Haar AC Wavelet")
```

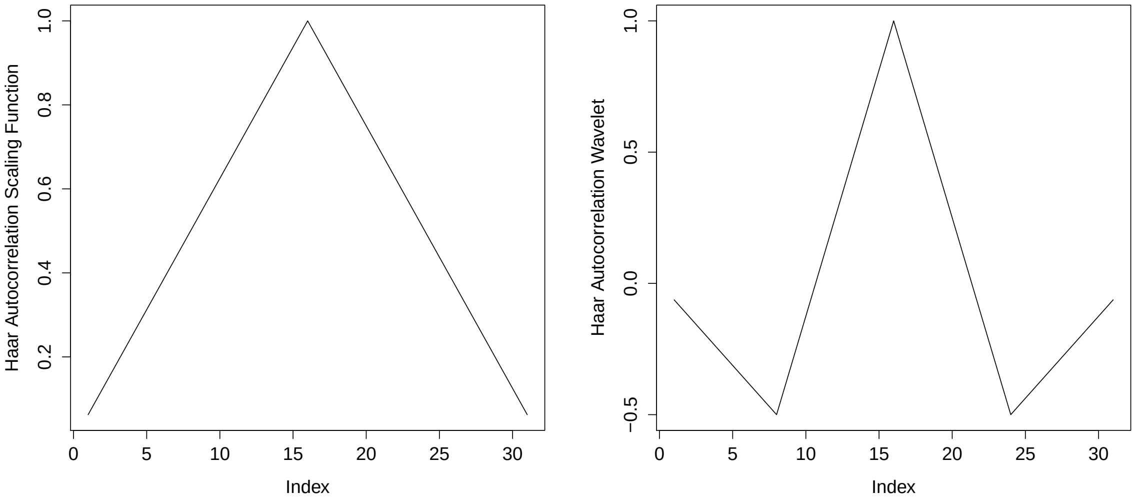


Figure 2: Examples of one-dimensional autocorrelation wavelets. Left: Scale 4 Haar ac father wavelet. Right: Scale 4 Haar ac wavelet.

produces the fourth scale ac father and ac wavelet (see Figure 2), respectively. The inputs used within the above function calls are:

- `J`: The scale of the ac (father) wavelet generated. Note that to follow existing conventions within **wavethresh**, we have adopted a negative scale within the **LS2W** code.
- `filter.number`: Determines the regularity of the wavelet.
- `family`: Determines the family of wavelets used (either `DaubExPhase` or `DaubLeAsymm`).

For details of other optional arguments, please refer to the associated R help pages.

The 2D discrete ac father wavelet is similarly given by $\Phi_j(\boldsymbol{\tau}) = \Phi_j(\tau_1)\Phi_j(\tau_2)$. Details of the related **LS2W** function are provided in Section 4 where we shall see that autocorrelation wavelets play an important role in the estimation of the LWS. However, before this, we introduce the wavelet local covariance.

Definition 5. *The* wavelet local covariance *(LCV), $C(\mathbf{z}, \boldsymbol{\tau})$, of a given LS2W process with LWS $\{S_j^l(\mathbf{z})\}$ is defined to be*

$$C(\mathbf{z}, \boldsymbol{\tau}) = \sum_l \sum_{j=1}^{\infty} S_j^l(\mathbf{z}) \Psi_j^l(\tau), \tag{11}$$

where $\boldsymbol{\tau} \in \mathbb{Z}^2$ and $\mathbf{z} \in (0,1)^2$.

To date, there has been limited research into the behavior and estimation of the wavelet LCV. More details can be found in Section 2.6 of Eckley *et al.* (2010), who describe the asymptotic behavior of the LCV and establish the existence of an "inverse formula".

4. The LS2W package

The **LS2W** package implements the estimation scheme described in previous sections. It makes use of various algorithms contained within the **wavethresh** package. Below we provide brief descriptions of the main functions contained within the package:

- `PhiJ`: Computes discrete autocorrelation father wavelets. These are automatically stored within the R session, using a naming convention governed by `Phi1Dname`.
- `PsiJ`: Computes discrete autocorrelation mother wavelets. These are automatically stored within the R session, using a naming convention governed by `Psi1Dname`.
- `D2ACW`: Computes two-dimensional autocorrelation wavelets. These are autmoatically stored within the R session, using a naming convention governed by `Psi2Dname`.
- `D2autoplot`: Can be used to generate images of two-dimensional discrete autocorrelation wavelets.
- `D2Amat`: Computes the inner product matrix of two-dimensional discrete autocorrelation wavelets. These objects are automatically stored within a session following a naming convention governed by `A2name`.

- `cddews`: Computes the local wavelet spectrum estimate as described in Eckley *et al.* (2010), returning an object of class `cddews`.
- `specplot`: Plots the local wavelet periodogram associated with a cddews object.
- `Haar2MA.diag`: Generates a two-dimentional Haar MA object (diagonal direction) of specified size and order. The functions `Haar2MA.vert, Haar2MA.horiz` Can be used to construct equivalent realizations for process in the vertical and horizontal direction respectively whilst `HaarMontage` generates a realization of a concatenated two-dimensional Haar MA process (as per Figure 3).

The package can effectively be segmented into three elements:

1. Simulation functions which generate certain forms of LS2W processes (`Haar2MA.diag`, `Haar2MA.vert`, `Haar2MA.horiz`, `HaarMontage`).
2. Functions which generate key two-dimensional wavelet-based constructs: E.g. the discrete autocorrelation wavelets and the inner product matrix of discrete autocorrelation wavelets (`PhiJ, PsiJ`, `D2ACW`, `D2autoplot`, `D2Amat`).
3. Functions which implement the estimation scheme (`cddews`, `specplot`, `summary.cddews` and `print.cddews`).

Below we illustrate the use of the **LS2W** package by way of a simulated example, based on non-stationary textures such as those generated by `HaarMontage`. In Section 4.1 we describe the simulation functions. Section 3.6 describes the functions which generate the key two-dimensional wavelet elements used in the analysis of LS2W processes, whilst Section 4.3 describes the functions which are associated with the estimation scheme.

In essence, `cddews` is the central function within the package. This executes the LS2W estimation algorithm, drawing on the other functions to calculate the autocorrelation (ac) wavelets and their inner product matrix.

4.1. Simulating LS2W processes

To begin with, let us simulate a realization from such a process where contributions come from the diagonal components only:

```
R> library("LS2W")
R> monty <- HaarMontage(direction = "diagonal")
R> image(monty, col = gray(c(0:255)/255))
```

An example of such a realization is displayed in Figure 3. Note in particular how the second-order structure appears stationary within individual quadrants, but is clearly different from one quadrant to another. Explicitly, the underlying structure for this process is $X_{\mathbf{r}} = \sum_{j=1}^{4} \sum_{\mathbf{u}} w^d_{j,\mathbf{u}} \psi^d_{j,\mathbf{u}}(\mathbf{r}) \xi^d_{j,\mathbf{u}}$, where $\{\psi^l_{j,\mathbf{u}}\}$ is the set of 2D Haar non-decimated discrete wavelets, and

$$w^d_{j,[2^J\mathbf{z}]} = \begin{cases} \sigma & \text{for } j=1, \mathbf{z} \in (0,1/2) \times (0,1/2); \\ \sigma & \text{for } j=2, \mathbf{z} \in (1/2,1) \times (0,1/2); \\ \sigma & \text{for } j=3, \mathbf{z} \in (0,1/2) \times (1/2,1); \\ \sigma & \text{for } j=4, \mathbf{z} \in (1/2,1) \times (1/2,1); \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

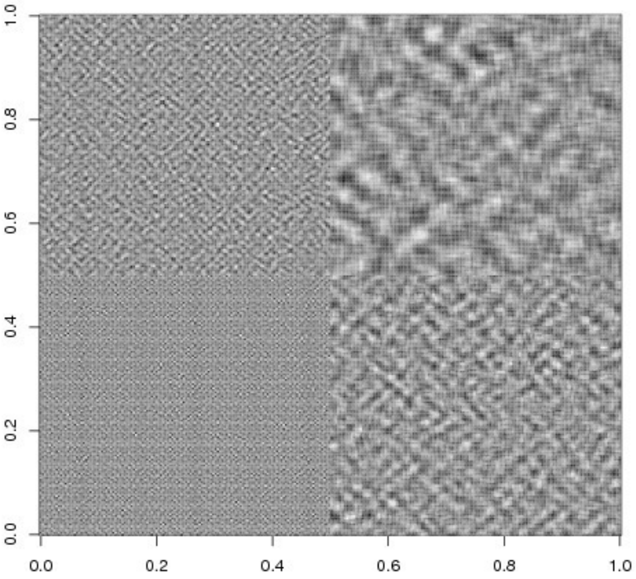


Figure 3: Simulation of a `HaarMontage` object.

In other words the `HaarMontage` code generates a simulation of concatenated HaarMA processes (see Section 3.3 for details). By construction, such images are of size 256 × 256.

4.2. Two-dimensional autocorrelation wavelet generation

The concept of an ac wavelet underpins the LS2W estimation scheme. As described in Section 3.6, the one-dimensional mother and father ac wavelets, Ψ_j and Φ_j, can be constructed using the functions `PsiJ` and `PhiJ` respectively. Recall also, from Section 2.1, that we have adopted a separable form for the two dimensional discrete wavelets. As we have previously described, the autocorrelation of these discrete wavelets is required for the correction of the (biased) raw wavelet periodograms. Consequently, the two dimensional autocorrelation wavelet also inherits this separability (see Eckley and Nason 2005 for further details).

To see the form of one of the autocorrelation wavelets associated with the estimation of such LS2W processes, we can use the `D2autoplot` function. The main arguments which can be used within this function are:

- `J`: The scale of the two-dimensional ac wavelet to be generated. Again we follow **wavethresh**'s convention by using negative numbers for this argument (`-1`: Fine, `-J`: Coarse).
- `filter.number`: Determines the regularity of the wavelet.
- `family`: Determines the family of wavelets used (`DaubExPhase` or `DaubLeAsymm`).
- `direction`: Determines whether the horizontal (`1`), vertical (`2`) or diagonal (`3`) autocorrelation wavelet is generated.

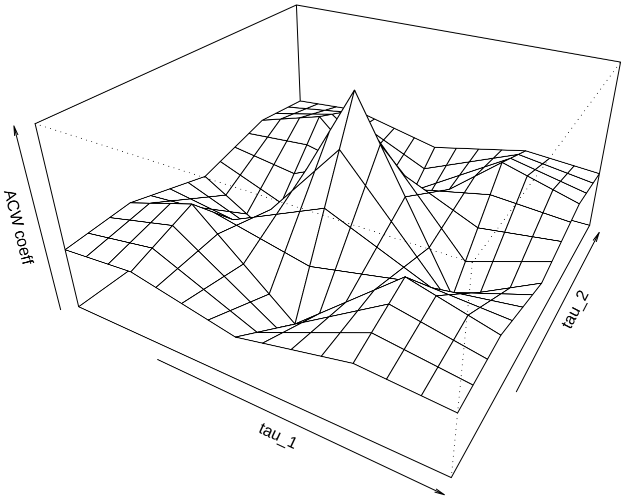


Figure 4: Example of two-dimensional Haar ac wavelet ($J = -3$).

- `main`: An overall title for the plot.
- `scaling`: Permits the scaling of the x and y-axes (for Haar ac wavelets only). If any other wavelet family is used, then set `scaling = "other"`.

Hence to generate the level 3 Haar ac wavelet in the diagonal direction (see Figure 4) we simply issue the following command:

```
R> D2autoplot(J = -3, filter.number = 1, family = "DaubExPhase",
+    direction = 3, main = " ", scaling = "Haar")
```

Note, in particular that `D2autoplot` calls `D2ACW` to compute the 2D discrete autocorrelation wavelets. The construction method employed within `D2ACW` is a brute force approach – a more elegant solution would be based on the recursive schemes as described in Eckley and Nason (2005). The routine returns only the values of the discrete ac wavelets, not their spatial positions. Each discrete autocorrelation wavelet is compactly supported. This support is determined from the discrete wavelets upon which these autocorrelations are based.

As a consequence of running `D2autoplot` a list containing $3J$ ac wavelet components is returned. The list elements are numbered from 1 to $3J$. The first J components contain the vertical autocorrelation wavelet coefficients, the second set of J components contains the horizontal autocorrelation wavelet coefficients (scales $1, \ldots, J$) and the last J components constitute the diagonal autocorrelation wavelet coefficients. Note that these 2D autocorrelation wavelets are stored as matrices. The central element of the matrix refers to lag 0. To minimize future computational cost, these ac wavelet lists are automatically saved within the workspace using a naming convention contained within `Psi2Dname`. Each save object has three defining characteristics: Its order, filter.number and family. Each of these three characteristics are concatenated together to form the name returned by `Psi2Dname`.

As we shall see in Section 4.3 the inner product matrix of discrete autocorrelation wavelets plays a crucial role in the estimation of LS2W processes. The inner product matrix A matrix of order $(3J) \times (3J)$ containing the elements $A_{j,l}$ defined in Eckley *et al.* (2010). Each element is the sum over all lags of the product of the matrix coefficients of a 2D DACW matrix at level j_1 in direction l_1 with that of another (not necessarily different) matrix of DACW coefficients at level j_2 in direction l_2. The structure of this matrix is as follows: The rows and columns of the matrix are labeled $1, \ldots, 3J$ in accordance with the notation of Eckley *et al.* (2010). When `switch = "direction"` the matrix has the following structure:

- Levels $1, \ldots, J$ correspond to the different levels of the decomposition in the vertical direction. $1 =$ fine and $J =$ coarse scale.
- Levels $J + 1, \ldots, 2J$ correspond to the different levels in the horizontal direction.
- Levels $2J + 1, \ldots, 3J$ correspond to the different directions in the diagonal direction.

When `switch = "level"`, the row and column elements cycle as follows: level 1 vertical, level 1 horizontal, level 1 diagonal, level 2 vertical, etc.

Within **LS2W** an inner product matrix of two-dimensional autocorrelation wavelets can be constructed using the function `D2Amat`. For example

```
R> D2Amat(J = -3, filter.number = 1, family = "DaubExPhase",
+    switch = "direction")
```

results in the following output

```
        0     -1       -2     -3     -4       -5      -6     -7     -8
0  2.2500 1.3125  0.70312 0.2500 0.3125  0.20312 0.75000 0.9375 0.6094
-1 1.3125 4.8125  3.79687 0.3125 0.5625  0.79687 0.18750 1.3125 2.3906
-2 0.7031 3.7969 15.45312 0.2031 0.7969  1.89062 0.04687 0.4219 3.9531
-3 0.2500 0.3125  0.20312 2.2500 1.3125  0.70312 0.75000 0.9375 0.6094
-4 0.3125 0.5625  0.79687 1.3125 4.8125  3.79687 0.18750 1.3125 2.3906
-5 0.2031 0.7969  1.89062 0.7031 3.7969 15.45312 0.04687 0.4219 3.9531
-6 0.7500 0.1875  0.04687 0.7500 0.1875  0.04687 2.25000 0.5625 0.1406
-7 0.9375 1.3125  0.42187 0.9375 1.3125  0.42187 0.56250 3.0625 1.2656
-8 0.6094 2.3906  3.95312 0.6094 2.3906  3.95312 0.14062 1.2656 8.2656
```

We have described the above functions for completeness. However it is important to note that it is not necessary to invoke the above functions when estimating the LWS structure of a two-dimensional structure. All of the above functions are called automatically when the function `cddews`, which we describe in the next section, is invoked.

4.3. The estimation scheme

We now summarize the main aspects of the estimation scheme proposed by Eckley *et al.* (2010). To start, we define our 'transformed' data: the empirical wavelet coefficients of the process:

Definition 6. *Let $\{X_r\}$ be a LS2W process. Then the* empirical wavelet coefficients *of the process are given by* $d^l_{j,\mathbf{u}} \equiv \sum_{\mathbf{r}} X_{\mathbf{r}} \psi^l_{j,\mathbf{u}}(\mathbf{r})$.

Input:

- A dataset consisting of a square greyscale image matrix of dimension $2^J \times 2^J$, X.
- A wavelet used to decompose the image object, $\psi_{2d,\text{decompose}}$.
- A second-stage wavelet used to smooth the spectral estimate, $\psi_{2d,\text{smooth}}$.

Estimation:

1. Calculate the non-decimated wavelet transform of X with respect to $\psi_{2d,\text{decomp}}$.
2. Square the non-decimated detail coefficients to obtain the raw periodogram.
3. Smooth the raw periodogram using the second-stage orthonormal wavelet basis $\psi_{2d,\text{smooth}}$
4. Calculate the inner product matrix of discrete autocorrelation wavelets based on $\psi_{2d,\text{decomp}}$ (A).
5. Correct the smoothed raw periodogram by applying the inverse of the inner product matrix (A^{-1}).

Output:

- The $3J \times 2^J \times 2^J$ array of the smoothed, corrected LWS estimate.

Table 1: The generic LSW estimation algorithm.

Recall from stationary theory that an (inconsistent) estimate of the spectral density function is given by the squared absolute value of the Fourier transform. As in NvSK, the estimator which we propose for the LWS is founded upon the collection of squared empirical wavelet coefficients – the local wavelet periodogram. Using empirical wavelet coefficients it is possible to introduce an equivalent estimator for the local wavelet spectrum. In this case, the Fourier transform is replaced by the collection of squared empirical wavelet coefficients.

Definition 7. *The* ***local wavelet periodogram*** *(LWP) of a LS2W process* $\{X_{\mathbf{r}}\}$ *is defined as*

$$I^l_{j,\mathbf{u}} \equiv |d^l_{j,\mathbf{u}}|^2. \tag{13}$$

Following NvSK, Eckley *et al.* (2010) demonstrate that this particular estimator is biased (see Theorem 2 of their paper). However, if we denote the vector of raw LWPs as $\mathbf{I}(\mathbf{z}) = \left\{I_{\eta,[\mathbf{zR}]}\right\}$, then the form of the bias suggests the following transformation of the spectra:

$$\mathbf{L}(\mathbf{z}) = A^{-1}\mathbf{I}(\mathbf{z}), \tag{14}$$

where $A = (A_{\eta,\nu})_{\eta,\nu\geq 1}$ is an operator defined by $A_{\eta,\nu} \leq \Psi_\eta, \Psi_\nu \geq \sum_{\boldsymbol{\tau}} \Psi_\eta(\boldsymbol{\tau})\Psi_\nu(\boldsymbol{\tau})$. This correction produces an asymptotically unbiased estimate of the LWS, however we should note that these estimates have an asymptotically non-vanishing variance, i.e., this bias-corrected estimator is inconsistent. Consequently estimates of the LWS must be smoothed to obtain consistency. In practice, this can be achieved by smoothing the raw (uncorrected) periodogram, using an orthonormal second stage wavelet basis $\{\tilde{\psi}_{l,m}\}$, and then inverting the smoothed transformation to obtain the smoothed estimate, $\tilde{L}_{\eta}(\mathbf{z})$. See Eckley *et al.* (2010) for further details.

Pseudo-code for the above estimation scheme is provided in Table 1. Within **LS2W** this estimation scheme described is contained within the function `cddews`. The main arguments for `cddews` are

- `data`: The image which you want to analyse.
- `filter.number`: The index of the wavelet used in the analysis of the time series (i.e., the wavelet basis functions used to model the time series). For Daubechies compactly supported wavelets the filter number is the number of vanishing moments.
- `switch`: This specifies the order of the corrected spectrum. Two options are available: The default is `switch = "direction"` which structures the matrix by scale within each decomposition direction. Thus, the ordering goes as follows $(-1, V), (-2, V), (-3, V), \ldots$. The alternative is `switch = "level"` structures the matrix by direction within each scale. Thus the ordering is as follows $(-1, V), (-1, H), (-1, D), (-2, V), (-2, H), \ldots$. For further details, see Eckley *et al.* (2010).
- `correct`: Eckley *et al.* (2010) have demonstrated that, as a consequence of the inherent redundancy of the non-decimated wavelet transform, the raw wavelet spectrum is biased. However, an asymptotically unbiased estimator may be obtained by applying the inverse of the inner product matrix of discrete autocorrelation wavelets. This logical argument permits the user to decide whether or not to correct for this inherent bias. By default, this is set to `TRUE`.
- `verbose`: A logical variable which allows certain informative messages to be printed on screen. The default setting for this variable is `FALSE`.
- `smooth`: A logical variable which allows the user to specify whether or not the resulting LWP should be smoothed. It is advised that this option be set to `TRUE` in order that consistent estimates are obtained.
- `sm.filter.number`: Selects the index number of the wavelet that smooths each scale of the wavelet periodogram. A default value is provided for this variable.
- `sm.family`: Selects the wavelet family that smooths each scale of the wavelet periodogram. A default value is provided for this variable.

Example: Returning to the Haar Montage example which we considered in Section 4.1, the final and key step of the analysis is to estimate the LWS structure for our data, the simulated non-stationary process, `monty`:

```
R> monty.cddews <- cddews(monty, filter.number = 1, family = "DaubExPhase")
```

By default, the above function will correct and smooth the LWP to obtain an (asymptotically) unbiased and consistent estimate of the LWS. The resulting object `monty.cddews` is an example of a `cddews.object` and has class `cddews`. The `cddews` class objects are returned as lists. Two methods are available for this object class : `summary` and `print`.

```
R> summary(monty.cddews)

Locally stationary two-dimensional wavelet decomposition structure
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Levels:  8
dimension of original image was:  256 x 256 pixels.
Filter family used:  DaubExPhase Filter index (N): 1
Structure adopted: direction
Date:  Fri Jul 15 15:56:28 2011
```

tells us that the dimension of the original image was 256 × 256, the analysis was performed using the N = 1 Daubechies compactly supported wavelet from the DaubExPhase family (i.e., Haar), and that the analysis output is structured by decomposition direction.

To discover what the monty.cddews object contains, we can use the print method:

```
R> print(monty.cddews)

Class 'cddews' : corrected directional dependent wavelet spectrum:
       ~~~~~~  : List with 14 components with names
              S datadim filter.number family structure nlevels correct smooth
              sm.filter.number sm.family levels type policy date

The spectrum of this image was corrected (IP matrix).
$S is a large array of data
The spectra have been smoothed to obtain consistency.
```

Amongst the 14 listed components, monty.cddews contains the following elements:

- S: The LWS estimate of the input data. This is a large array, the first dimension refers to a specific scale-direction pair. The next dimension refers to the rows of the spectral image, whilst the third element refers to the columns of the image.
- datadim: The dimension of the original image.
- filter.number: The index of the wavelet used in the analysis of the image. For Daubechies compactly supported wavelets the filter number is the number of vanishing moments.
- family: The wavelet family used in the analysis of the image (i.e., the wavelet family used in the modelling).
- structure: Explains the structure of the inner product matrix and S. It can only take two values, direction and scale.
- nlevels: The number of levels in the decomposition.
- correct: TRUE or FALSE, depending on whether the user corrected for the bias.
- Smooth: TRUE or FALSE, depending on whether the LWP has been smoothed.
- policy: The smoothing policy adopted.

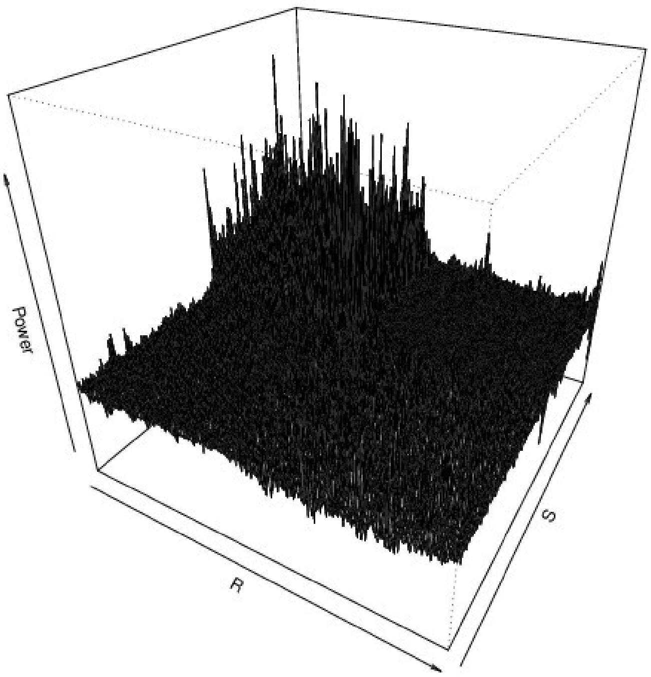


Figure 5: Estimate of the LWS for the second-finest scale structure in the diagonal decomposition direction for the HaarMontage simulation.

Finally, the function `specplot` can be used to display the collection of LWS estimates returned by the `cddews` function. This function requires the specification of several arguments:

- `cddews`:An object of class cddews must be supplied to the function.
- `scaling`: Two scaling options are available. The default setting is to scale `"by.level"` – an option which is useful if you wish to compare coefficients within a resolution level. The alternative setting is `"global"` scaling, whereby one scale factor is chosen for all plots. This factor depends on the largest coefficient which is to be included in the suite of plots.
- `arrangement`: Allows the user to specify the number of spectral plots which are to appear on any given page. The default is `c(3,3)`.
- `page`: A logical argument which allows the user to request that they be prompted when a new page of plots appears
- `dataname`: A name for the image whose LWP is being displayed. This will appear as part of the title associated with each plot.
- `verbose`: A logical argument which, if set to `TRUE` , prints certain informative statements to screen during execution.
- `display`: Two display methods are available. Using the option `display = "persp"` displays a 3-dimensional plot of the LWP, using the routine persp. Alternatively setting `display = "image"` displays the LWP as a collection of images.

- `reset`: If set to `TRUE`, this restores the plot settings to their default configuration (i.e., `par(mfrow = c(1, 1))`). If `FALSE`, then the current settings will remain in operation.
- `wtitle`: A logical variable which dictates whether a common title is displayed on all spectral plots.

For example

```
R> specplot(monty.cddews, display = "persp")
```

returns the entire collection of local spectral estimates, one level at a time. Of the two display modes available, in practice, we have found that `"persp"` provides the most visually interpretable output.

An example of the LWS estimate for the second-finest scale structure in the diagonal decomposition direction is displayed in Figure 5. This is one of the 24 spectral images generated for the `monty` texture. Note how negligible power exists in three quadrants, with power restricted to one quadrant alone. This is precisely the spectral form which we would expect for such a process.

5. Case study: Texture analysis

As discussed earlier, texture analysis is one possible area where the LS2W approach can prove useful. Texture can broadly be considered to be the visual character of an image region whose structure is, in some sense, regular. Figure 6 provides some examples of textured images: Fabric, hair and a creased material. Clearly the eye is able to distinguish between these images, but there is considerable interest in being able to automatically discriminate/classify such images (see Eckley *et al.* 2010 and references therein for an introduction to the field). It is therefore natural to ask, can **LS2W** be used to discriminate between the three texture types?

To investigate the discriminatory ability of the LS2W approach, 25 sub-images of size 64×64 pixels were randomly sampled from each image. Let us start by loading data from **LS2W**:

```
R> data("textures", package = "LS2W")
```

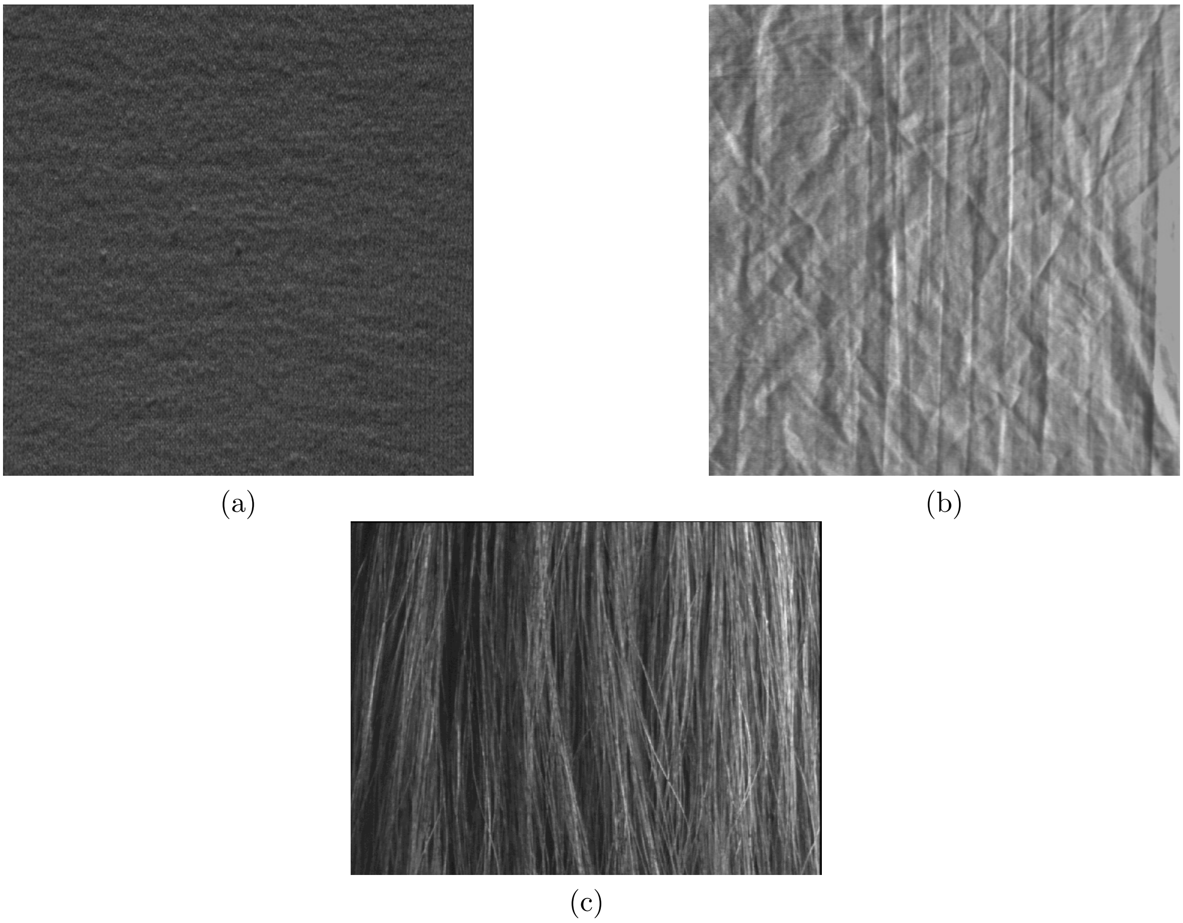
Three new objects now enter the workspace: `A`, `B` and `C`. Each of these objects contains the data for a greyscale image (as displayed in Figure 6). To view, e.g., image A in grayscale simply type:

```
R> image(A, col = gray(c(0:255)/255))
```

The textured images may contain a non-zero mean. To remove any underlying mean structure we use the `medpolish` function within the **stats** package.

```
R> A.mp <- medpolish(A)$residuals
R> B.mp <- medpolish(B)$residuals
R> C.mp <- medpolish(C)$residuals
```

Next, let us randomly sample sub-images from each texture type and generate a feature vector for each sub-image. For a given image, the following simple function (i) randomly samples (x, y) start co-ordinates within the given image for each sub-image, then (ii) estimates the LWS for these before (iii) generating a simple feature vector. This function is available in **LS2W**.



(a)

(b)

(c)

Figure 6: Examples of textured images and their pixel dimensions (a) fabric (1024 × 1024); (b) creased material (512 × 512) and (c) hair (459 × 612).

```
R> sample.stats <- function(x, n = 25, size = 64)
+  {
+    xcoords <- sample.int(nrow(x) - size, n)
+    ycoords <- sample.int(nrow(y) - size, n)
+    J <- log(size, 2)
+    feature.matrix <- matrix(0,nrow = n, ncol = 3 * J)
+    for(i in 1:n) {
+      sam <- x[xcoord[i]:(xcoord[i] + size - 1),
+        ycoord[i]:(ycoord[i] + size - 1)]
+      sam.ls2w <- cddews(sam, filter.number = 1,
+        family = "DaubExPhase", levels = 3:5)
+      feature.matrix[i,] <- apply(sam.ls2w$S, 1, sum)
+    }
+    return(feature.matrix)
+  }
```

The arguments for the function are (i) `x`, the image being analysed; (ii) `n`, the number of sub-images being sampled from `x` and (iii) `size`, the dimension of the square sub-images

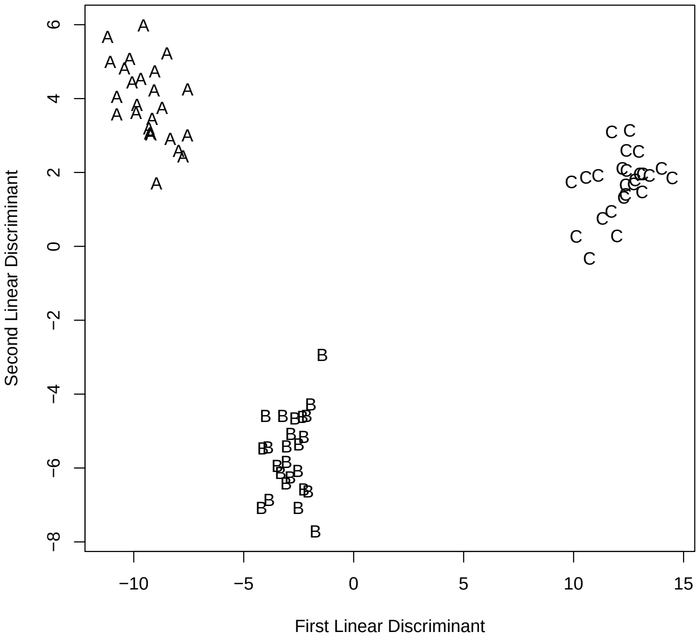


Figure 7: Linear discriminant analysis plot for texture images features derived from the LS2W model. A = fabric, B = creased material, C = hair.

sampled from `x`. Please note this code is available in the files which accompany this paper. When invoked, the function `sample.stats` returns the collection of feature vectors obtained for each sample obtained from the image of interest (`x`).

The feature vector which we adopt consists of $3J$ elements, each of which represents the test statistic: $\mathbf{t}(T_I) = \sum_{\mathbf{z}} \tilde{\mathbf{L}}(\mathbf{z}) = \sum_{\mathbf{z}} A_J^{-1}\tilde{\mathbf{I}}([\mathbf{zR}])$. Here $\tilde{\mathbf{I}}$ denotes the smoothed (uncorrected) local wavelet periodogram. Thus each element of the feature vector provides a measure of the contribution made to the overall local variance structure at scale j within direction l.

To start our example, let us generate the feature vectors for 25 subsamples, each of size 64×64 from images `A`, `B` and `C`, together with a vector of labels identifying each sample's texture:

```
R> A.stats <- sample.stats(A.mp, 25, 64)
R> B.stats <- sample.stats(B.mp, 25, 64)
R> C.stats <- sample.stats(C.mp, 25, 64)
R> all.stats <- rbind(A.stats, B.stats, C.stats)
R> imlabels <- c(rep("A", 25), rep("B", 25), rep("C", 25))
```

As an exploratory tool to assess the potential of these LS2W-based measures, we consider a discriminant analysis of the features using the `lda` function in **MASS** (Venables and Ripley 2002). To perform this, simply invoke the following code:

```
R> all.stats.lda <- lda(all.stats, imlabels)
R> all.stats.ld <- predict(all.stats.lda, dimen = 2)$x
```

```
R> plot(all.stats.ld, type = "n", xlab = "First Linear Discriminant",
+    ylab = "Second Linear Discriminant")
R> text(all.stats.ld, imlabels)
```

Figure 7 displays a plot of the first two linear discriminant axes for the LS2W feature set. Note in particular how one can clearly discern the three different textures. As one might expect, due to the regularity of the patterns in images A and B, the samples generated from these images fall very near one another whilst the more irregular image (C) results in more variable samples.

6. Summary

The **LS2W** package provides an implementation of the LS2W estimation scheme. This is not currently provided by other wavelet-related R packages on CRAN. **LS2W**'s core functionality consists of the construction of two-dimensional autocorrelation wavelets, the inner product matrix of autocorrelation wavelets and an estimation scheme for local wavelet spectra in two-dimensions. As such *LS2W* is useful for estimating the (non-stationary) second-order structure within textured images.

Acknowledgments

The authors would like to thank Matt Nunes for his advice during the development of the **LS2W** package. They are also grateful to the Associate Editor and two anonymous referees for providing several constructive comments which have helped improve this paper. Nason and Eckley gratefully acknowledge funding from the EPSRC SuSTaIn grant EP/D063485/1 to support the work for this paper. Eckley also acknowledges the financial support of Unilever Research.

References

Cressie NAC (1991). *Statistics for Spatial Data*. John Wiley & Sons.

Dahlhaus R (1997). "Fitting Time Series Models to Nonstationary Processes." *The Annals of Statistics*, **25**, 1–37.

Dahlhaus R, Polonik W (2009). "Empirical Spectral Processes for Locally Stationary Time Series." *Bernoulli*, **15**, 1–39.

Daubechies I (1988). "Orthonormal Bases of Compactly Supported Wavelets." *Communications on Pure and Applied Mathematics*, **XLI**, 909–966.

Daubechies I (1992). *Ten Lectures on Wavelets*. SIAM, Philadelphia.

Daugman JG (1990). "An Information-Theoretic View of Analog Representation in Striate Cortex." In EL Schwartz (ed.), *Computational Neuroscience*, pp. 403–423. MIT Press, Cambridge.

Eckley IA (2001). *Wavelet Methods for Time Series and Spatial Data*. Ph.D. thesis, University of Bristol.

Eckley IA, Nason GP (2005). "Efficient Computation of the Inner-Product Matrix of Discrete Autocorrelation Wavelets." *Statistics and Computing*, **15**, 83–92.

Eckley IA, Nason GP, Treloar RL (2010). "Locally Stationary Fields with Application to the Modelling and Analysis of Image Texture." *Journal of the Royal Statistical Society C*, **59**, 595–616.

Field DJ (1999). "Wavelets, Vision and the Statistics of Natural Scenes." *Philosophical Transactions of the Royal Society of London A*, **357**, 2527–2542.

Fryzlewicz P, Nason GP (2006). "Haar-Fisz Estimation of Evolutionary Wavelet Spectra." *Journal of the Royal Statistical Society B*, **68**, 611–634.

Haining RP (1978). "The Moving Average Model for Spatial Interaction." *Transactions of the Institute of British Geographers*, **3**, 202–225.

Mallat SG (1989). "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 674–693.

Mallat SG (1999). *A Wavelet Tour of Signal Processing*. 2nd edition. Academic Press, London.

Moore M (1988). "Spatial Linear Processes." *Communications in Statistics: Stochastic Models*, **4**, 45–75.

Nason G (2010). "**wavethresh**: Wavelets Statistics and Transforms." R package version 4.5, URL http://CRAN.R-project.org/package=wavethresh.

Nason GP (2008). *Wavelet Methods in Statistics with R*. Springer-Verlag.

Nason GP, Silverman BW (1995). "The Stationary Wavelet Transform and Some Statistical Applications." In A Antoniadis, G Oppenheim (eds.), *Wavelets and Statistics*, number 103 in Lecture Notes in Statistics, pp. 281–300. Springer-Verlag.

Nason GP, von Sachs R, Kroisandt G (2000). "Wavelet Processes and Adaptive Estimation of the Evolutionary Wavelet Spectrum." *Journal of the Royal Statistical Society B*, **62**, 271–292.

Priestley MB (1981). *Spectral Analysis and Time Series*. Academic Press, London.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Van Bellegem S, Dahlhaus R (2006). "Semiparametric Estimation by Model Selection for Locally Stationary Processes." *Journal of the Royal Statistical Society B*, **68**, 721–746.

Van Bellegem S, von Sachs R (2008). "Locally Adaptive Estimation of Evolutionary Wavelet Spectra." *The Annals of Statistics*, **36**, 1879 – 1924.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S.* 4th edition. Springer-Verlag, New York.

Vidakovic B (1999). *Statistical Modelling by Wavelets.* John Wiley & Sons.

Affiliation:

Idris A. Eckley
Department of Mathematics and Statistics
Lancaster University
LA1 4YF, United Kingdom
E-mail: `i.eckley@lancaster.ac.uk`
URL: `http://www.maths.lancs.ac.uk/~eckley/`

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/

Volume 43, Issue 3 *Submitted:* 2009-11-30
July 2011 *Accepted:* 2011-06-14