Package 'dualtrees'

October 28, 2019

Title Decimated and Undecimated 2D complex dual-tree wavelet transform

Version 0.0.1

Description What the package does (one paragraph).

Depends R (>= 3.5.0)

License What license is it under?

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

R topics documented:

A_b — *Bias correction matrices for the "b"-wavelets*

Description

Matrices supposedly needed in order to eliminate the effect of "spectral leakage" for the local dtcwt-spectra. Used by biascor(...).

Usage

```
A_b
```

Format

A list with entries N512, N256, N128, N64, N32, each containing the bias correction matrix of appropriate size

Source

Calculated by hand via /user/s6sebusc/wavelets_verification/general_scripts/Amats_cdtwt.r

Examples

```
image( A_b$N512 )
```

A_b_bp — *Bias correction matrices for the "b_bp"-wavelets*

Description

Matrices supposedly needed in order to eliminate the effect of "spectral leakage" for the local dtcwt-spectra. Used by biascor(...).

Usage

```
A_b_bp
```

Format

A list with entries N512, N256, N128, N64, N32, each containing the bias correction matrix of appropriate size

Source

Calculated by hand via /user/s6sebusc/wavelets_verification/general_scripts/Amats_cdtwt.r

Examples

```
image( A_b_bp$N512 )
```

biascor *spectral bias correction, implemented in FORTRAN*

Description

spectral bias correction, implemented in FORTRAN

Usage

```
biascor(en, a)
```

blossom *Two meteorologists in front of cherry blossoms*

Description

A very beautiful image.

Usage

```
blossom
```

Format

A 512x512 matrix of gray-scale values

Source

real life

Examples

```
image(blossom, col=gray.colors(32,0,1))
```

boys *Two stromchasers in the sun*

Description

Another classic image.

Usage

```
boys
```

Format

A 256x256 matrix of gray-scale values

Source

real life

Examples

```
image(boys, col=gray.colors(32,0,1))
```

---

c2q *Transform six fields of complex coefficients back into four trees.*

Description

This function takes the the six directional complex daughter wavelet coefficients and re-constructs the three combinations of high- and low passes from the four trees (ab, ba, aa, bb).

Usage

```
c2q(comp)
```

Arguments

comp	complex array of dimnesions nx, ny, 6

Value

a list of low- and high-pass components from the four trees, names LoaHia, LobHib, etc.

Examples

```
c2q( comp )
```

---

cen2uv *transform angle and anisotropy into vector components for plotting*

Description

transform angle and anisotropy into vector components for plotting

Usage

```
cen2uv(cen)
```

decimate *delete every second row of a matrix*

Description

delete every second row of a matrix

Usage

```
decimate(mat, odd = FALSE, dec = TRUE)
```

dt2cen *get the centre of the DT-spectrum*

Description

get the centre of the DT-spectrum

Usage

```
dt2cen(pyr)
```

dtcwt *The 2D forward dualtree complex wavelet transform*

Description

This function performs the dualtree complex wavelet analysis, either with or withour decimation

Usage

```
dtcwt(mat, fb1 = near_sym_b, fb2 = qshift_b, J = NULL, dec = TRUE,
  mode = NULL, verbose = TRUE, boundaries = "periodic")
```

Arguments

mat	the real matrix we wish to transform
fb1	A list of analysis filter coefficients for the first level. Currently only near_sym_b and near_sym_b_bp are implemented
fb2	A list of analysis filter coefficients for all following levels. Currently only qshift_b and qshift_b_bp are implemented
J	number of levels for the decomposition. Defaults to `log2( min(Nx,Ny) )` in the decimated case and `log2( min(Nx,Ny) ) - 3` otherwise
dec	whether or not the decimated transform is desired
mode	how to perform the convolutions, either "direct" (default if dec=TRUE) or "FFT" (default if dec=FALSE)
verbose	if TRUE, the function tells you which level it is working on
boundaries	how to handle the internal boundary conditions of the convolutions, has no effect if mode="direct"

Details

This is the 2D complex dualtree wavelet transform as described by Selesnick et al 2005. It consists of four discerete wavelet transform trees, generated from two filter banks a and b by applying one set of filters to the rows and the same ot the other to the columns. In the decimated case (dec=TRUE), each convolution is followed by a downsampling, meaining that the size of the six coefficient fields is cut in half at each level. In this case, it is supposedly efficient to use direct convolutions (mode="direct"), the boundary conditions of which are steered by the boundaries-argument. If dec=FALSE, direct convolutions may be slow and you should use mode="FFT". In that case, you need to handle the boundary conditions externally (enter a nice 2^N x 2^M matrix) and the maximum level J is smaller than log2(N) due to the construction of the filters via an 'algorithme a trous'.

Value

if dec=TRUE a list of complex coefficient fields, otherwise a complex J * Nx * Ny * 6 array.

Note

Periodic and reflective boundaries are both implemented for the decimated case, but only the periodic boundaries are actually invertible at this point.

References

Selesnick, I.W., R.G. Baraniuk, and N.C. Kingsbury. "The Dual-Tree Complex Wavelet Transform." IEEE Signal Processing Magazine 22, no. 6 (November 2005): 123–51. https://doi.org/10.1109/MSP.2005.1550194.

See Also

idtcwt

Examples

```
dt <- dtcwt( boys )
par( mfrow=c(2,3), mar=rep(2,4) )
for( j in 1:6 ){
    image( boys, col=grey.colors(32,0,1) )
    contour( Mod( dt[[3]][ ,,j ] )**2, add=TRUE, col="green" )
}
```

---

fld2dt *transform a field, handle boundary conditions, return energy*

Description

transform a field, handle boundary conditions, return energy

Usage

```
fld2dt(fld, Nx = NULL, Ny = NULL, J = NULL, mode = NULL,
  correct = NULL, verbose = FALSE, boundary = "pad",
  fb1 = near_sym_b_bp, fb2 = qshift_b_bp)
```

get_en *get energy from the dualtree transform*

Description

get energy from the dualtree transform

Usage

```
get_en(pyr, correct = "fast")
```

holes *insert holes into a filter?*

Description

insert holes into a filter?

Usage

```
holes(fil, second = TRUE)
```

idtcwt *The 2D inverse dualtree complex wavelet transform*

Description

Reconstructs an image from the pyramid of complex directional wavelet coefficients.

Usage

```
idtcwt(pyr, fb1 = near_sym_b, fb2 = qshift_b, verbose = TRUE,
  boundaries = "periodic")
```

Arguments

pyr	a list containing arrays of complex coefficients for each level of the decomposition, produced by `dtcwt( ..., dec=TRUE )`.
fb1	the filter bank for the first level
fb2	the filter bank for all following levels
verbose	if true, the function will say a few words while doing its thing.
boundaries	how to handle the boundary conditions, should be the same as for the decomposition.

Details

This function re-arranges the six complex daughter coefficients back into the four trees, convolves them with the synthesis wavelets and adds everything up to recover an image. For the `near_sym_b` and `qshift_b` filter banks, this reconstrcution should be basically perfect. In the case of the the `b_bp` filters, non-negligible artifacts appear near +-45° edges.

Value

a real array of size `2N x 2M` where `dim( pyr[[1]] ) = (M,N,6)` .

Note

At present, only boundaries="periodic" actually works :(

References

Selesnick, I.W., R.G. Baraniuk, and N.C. Kingsbury. "The Dual-Tree Complex Wavelet Transform." IEEE Signal Processing Magazine 22, no. 6 (November 2005): 123–51. https://doi.org/10.1109/MSP.2005.1550194.

See Also

`dtcwt`

Examples

```
py <- dtcwt( boys )
boys_i <- idtcwt( py )
image( boys - boys_i )
```

`make_square` *Padded boundary conditions*

Description

Padded boundary conditions

Usage

```
make_square(picture, N, Ny = N, value = min(picture, na.rm = TRUE))
```

my_conv *Column-convolutions*

Description

This function convolves the columns of a matrix mat with a filter fil.

Usage

```
my_conv(mat, fil, dec = TRUE, mode = "direct", odd = FALSE,
  boundaries = "periodic")
```

Arguments

mat	a matrix
fil	the filter to convolve the columns with
dec	if TRUE, every second row is discarded after the convolution
mode	how to actually do the convolutions, must be either "direct" or "FFT"
odd	if TRUE, the first row is discarded, otherwise the second row is.
how	to handle the boundaries, does nothing if mode="FFT"

Details

This functions does all of the actual computations inside the wavelet transform. The direct mode uses filter(...) and can handle any field size you like. It is supposedly faster when the filters are short, i.e., in the decimated case. The FFT-version really only works when the input dimensions are whole powers of two and the filter is not longer than the columns of the matrix.

Examples

```
dboysdy <- my_conv( boys, c(-1,1), dec=FALSE )
dboysdx <- t( my_conv( t(boys), c(-1,1), dec=FALSE ) )
par( mfrow=c(1,2) )
image( dboysdx, col=gray.colors(32) )
image( dboysdy, col=gray.colors(32) )
```

near_sym_b *A q-shift filter for the second to last levels*

Description

Data from a QTL experiment on gravitropism in

Usage

```
data(qshift_b)
```

Format

A list of high- and low-pass filters for analysis and synthesis

Source

dtcwt python package

Examples

```
data(qshift_b)
```

---

near_sym_b_bp *A q-shift filter for the second to last levels*

Description

Data from a QTL experiment on gravitropism in

Usage

```
data(qshift_b)
```

Format

A list of high- and low-pass filters for analysis and synthesis

Source

dtcwt python package

Examples

```
data(qshift_b)
```

---

period_bc *Periodic boundary conditions*

Description

Periodic boundary conditions

Usage

```
period_bc(x, N, Ny = N)
```

---

put_in_mirror *Reflective boundary conditions*

Description

Reflective boundary conditions

Usage

```
put_in_mirror(x, N, Ny = N)
```

q2c *Transform data from the four trees to six fields of complex coefficients.*

Description

This function takes the four combinations of high- and low passes from the four trees (ab, ba, aa, bb) and re-arranges them into the six directional complex daughter wavelets.

Usage

```
q2c(q)
```

Arguments

q	a list of wavelet coefficients named LoaHia, LobHib, HiaLoa, ...

Value

a complex array of size nx, ny, 6

Examples

```
q2c( q )
```

qshift_b *A q-shift filter for the second to last levels*

Description

Data from a QTL experiment on gravitropism in

Usage

```
data(qshift_b)
```

Format

A list of high- and low-pass filters for analysis and synthesis

Source

dtcwt python package

Examples

```
data(qshift_b)
```

qshift_b_bp — *A q-shift filter for the second to last levels*

Description

Data from a QTL experiment on gravitropism in

Usage

```
data(qshift_b)
```

Format

A list of high- and low-pass filters for analysis and synthesis

Source

dtcwt python package

Examples

```
data(qshift_b)
```

shift1 — *shift a matrix forward or backward by one row*

Description

shift a matrix forward or backward by one row

Usage

```
shift1(x, forward = TRUE)
```

upsample — *add rows with zeroes to a matrix*

Description

add rows with zeroes to a matrix

Usage

```
upsample(mat, odd = TRUE)
```

Index