Package ‘calcWOI’

March 28, 2020

Type Package

Title Calculates the Wavelet-Based Organization Index

Version 1.0.3

Date 2020-03-24

Author Sebastian Brune, Sebastian Buschow, Florian Kapp, Petra Friederichs

Maintainer Sebastian Brune <sbrune@uni-bonn.de>

Depends R (>= 3.5.0), wavethresh (>= 4.5), LS2W (>= 1.3.4), dualtrees (>= 0.1.4)

Description Calculates the wavelet-based organization index following Brune et al (2018) (<doi:10.1002/qj.3409>), the modified wavelet-based organization index and the local wavelet-based organization index of an arbitrary 2D array using Wavelet Transforms of the LS2W package by Eckley et al (2010) (<doi:10.1111/j.1467-9876.2009.00721.x>) and Eckley and Nason (2011) (<doi:10.18637/jss.v043.i03>). In Version 1.0.3 the calculation of LW is added.

License GPL-3

LazyData true

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-03-28 08:20:17 UTC

R topics documented:

`calcWOI-package` *Calculates the Wavelet-Based Organization Index*

Description

Calculates the original wavelet-based organization index, the modified wavelet-based organization index and the local wavelet-based organization index of an arbitrary 2D array. Since version 1.0.3 the function LW is added, which calculates the local wavelet-based organization index with help of Dual-tree wavelets.

Details

This package provides all functions and tools to calculate the wavelet-based organization index (Brune et al., 2018). The central function within the calcWOI package is WOI. This function calculates the original WOI, the modified WOI and the local WOI (Brune et al., 2020a). All these indexes based on wavelet transforms (DaubExPhase4) done with the function wavtra, where we use parts of the cddews function within the package LS2W to create the function wavtra and constant data like the inverse correction matrix AI and the centre of mass matrix.

Other functions in calcWOI like flatten, blowup and buildperiodic are used to ensure that the incoming field is quadratic and periodic.

The function LW allows the calculation of the revised LWOI, namely LW, of an arbitrary array. Details on the new LW will be published in Brune et al (2020b).

Author(s)

Sebastian Brune, Sebastian Buschow, Florian Kapp, Petra Friederichs. Maintainer: Sebastian Brune <sbrune@uni-bonn.de>

References

Eckley, I.A., Nason, G.P. and Treloar, R.L. (2010) Locally stationary wavelet fields with application to the modelling and analysis of image texture. Journal of the Royal Statistical Society (Series C), 59, 595-616.

Eckley, I.A. and Nason, G.P. (2011). LS2W: Implementing the Locally Stationary 2D Wavelet Process Approach in R, Journal of Statistical Software, 43(3), 1-23.

Brune, S., Kapp, F., & Friederichs, P. (2018). A wavelet-based analysis of convective organization in ICON large-eddy simulations. Quarterly Journal of the Royal Meteorological Society, 144(717), 2812-2829.

Brune, S., Buschow, S., & Friederichs, P. (2020a). Observations and high-resolution simulations of convective precipitation organization over the tropical Atlantic. Quarterly Journal of the Royal Meteorological Society.

Brune, S., Buschow, S., & Friederichs, P. (2020b). The Local Wavelet-based Organization Index - Quantification, Localization and Classification of Convective Organization from Radar and Satellite Data. In preparation.

Examples

```
# Calculate WOI, modified WOI and LWOI for a random precipitation
# field using an 230x200 array with
# random positive numbers
x <- array(5 + rnorm(230*200), dim = c(230, 200))
s <- c(1,2)
l <- c(3,4)
thres <- 0.1
flat <- 5
WOIres <- WOI (x = x, s = s, l = l, thres = thres, flat = flat,
verbose = TRUE)
# original WOI (Brune et al., 2018)
WOIorig <- WOIres$WOIorig
print(paste("Original WOI:", WOIorig))
# modified WOI
WOI <- WOIres$WOI
print(paste("Modified WOI:", WOI))
# local WOI
LWOI <- WOIres$LWOI
par(mfrow = c(2, 2))
image(LWOI, main = "LWOI")

# Calculate the three components of the
# local wavelet-based organization index with Dual-tree wavelets
LW <- LW(x, thres = thres, Nx = 2^ceiling(log2(max(dim(x)))),
  Ny = 2^ceiling(log2(max(dim(x)))), boundaries = "pad")
image(LW$LWsc, main = "LWsc", zlim = 0:1)
image(LW$LWin, main = "LWin", zlim = 0:1)
image(LW$LWai, main = "LWai", zlim = 0:1)
```

AICEN *File with Inverse A matrix and centre of mass matrix*

Description

This file includes the lists AIx and CENx for x = 16, 32, 64, 128, 256, 512, 1024, 2048 or 4096. AIx is the inverse matrix calculated by Eckley et al. (2010). CENx includes the centre of mass for DaubExPhase4, calculated with help of the LS2W package by Eckley and Nason (2011).

Usage

```
data(AICEN)
```

Format

List of 18 elements (9 AIx and 9 CENx). CENx include lists for x and y direction.

Author(s)

Sebastian Buschow, Sebastian Brune

References

Eckley, I.A., Nason, G.P. and Treloar, R.L. (2010) Locally stationary wavelet fields with application to the modelling and analysis of image texture. Journal of the Royal Statistical Society (Series C), 59, 595-616.

Eckley, I.A. and Nason, G.P. (2011). LS2W: Implementing the Locally Stationary 2D Wavelet Process Approach in R, Journal of Statistical Software, 43(3), 1-23.

Examples

```
data(AICEN)
image(AICEN$AI256)
str(AICEN)
```

---

blowup — *Adds zeros around 2D array*

Description

This function adds zeros around a 2D array to get a M x M field.

Usage

```
blowup(x, M, number = 0)
```

Arguments

`x`	2D array, which should be blown up to a M x M array.
`M`	Dimension of new array including zeros at boundaries. M should be larger than max(dim(x)).
`number`	Number, that should be added to the incoming array. Default is 0.

Value

Returns the M x M array with the original field in the centre.

Author(s)

Florian Kapp

Examples

```
# Add zeros around a 3x3 matrix to generate a 8x8 matrix.
x <- matrix(1:9, nrow = 3)
new <- blowup(x = x, M = 8, number = 0)
new
```

`buildperiodic` *Builds periodic array by mirroring at side and top*

Description

This function generates periodic boundaries by mirroring at side and top. The resulting array is four times larger than the incoming array.

Usage

```
buildperiodic(x)
```

Arguments

`x`	2D quadratic array.

Value

Returns a 2D array, which is four times larger than the incoming array.

Author(s)

Sebastian Brune

Examples

```
x <- matrix(1:12, nrow = 4)
out <- buildperiodic(x = x)
out
```

`flatten` *Reduces the boundary gradients*

Description

This function smoothes the boundaries with a linear filter.

Usage

```
flatten(x, filter)
```

Arguments

`x`	2D array, which boundaries should be smoothed.
`filter`	The smoothing vector with increasing elements from 0...1. The length of the vector corresponds to the number of smoothed points at each side.

Value

Returns the incoming field with smoothed boundaries.

Author(s)

Florian Kapp

Examples

```
# Smooth outer 25 grid points
x <- array(10, dim = c(100, 200))
xflat <- flatten(x = x, filter = seq(0, 1, , 25))
par(mfrow = c(1, 2))
image(x, main = "Original")
image(xflat, main = "Smoothed Bound")
```

LW — *Calculates the wavelet-based organization index with the dualtree wavelet transform*

Description

This function calculates the locally wavelet-based organization index (LW) as defined in Brune et al. (2020) based on the dualtree complex wavelet transform.

Usage

```
LW(x, thres = 0.1, Nx = 2^ceiling(log2(max(dim(x)))),
  Ny = 2^ceiling(log2(max(dim(x)))), boundaries = "pad", verbose = FALSE)
```

Arguments

`x`	2D numeric array with dimensions larger than 16 x 16. dim(x)[1] has not to be equal to dim(x)[1], but NA and/or NaN are not allowed.
`thres`	0 or a positive number. Threshold for rain rate. Default is 0.1 mm/h, because we calculate LWOI only for grid points, where rain rate is >= 0.1 mm/h. For brightness temperatures we use 245 K.
`Nx`	The number of grid points in x direction of the array, which is put into the dualtree wavelet transform. The default is the log2 of maximum dimension of the ingoing array. Nx must be equal or larger than the first dimension of x.
`Ny`	The number of grid points in y direction of the array, which is put into the dualtree wavelet transform. The default is the log2 of maximum dimension of the ingoing array. Nx must be equal or larger than the second dimension of x.
`boundaries`	Handles the boundary conditions, either "pad", "mirror" or "periodic".
`verbose`	Default FALSE. Set TRUE for print statements.

Value

This functions returns a list with follwing elements:

`LWsc`	LWOI scale calculated with dualtrees. The array is masked with thres. The dimension of LWsc is equal to the dimension of x, if the boundaries are only padded with zeros.
`LWin`	LWOI intensity calculated with dualtrees. The array is masked with thres. The dimension of LWsc is equal to the dimension of x, if the boundaries are only padded with zeros.
`LWai`	LWOI anisotropy calculated with dualtrees. The array is masked with thres. The dimension of LWsc is equal to the dimension of x, if the boundaries are only padded with zeros.
`LWuu`	LWOI u component calculated with dualtrees. The array is masked with thres. The dimension of LWsc is equal to the dimension of x, if the boundaries are only padded with zeros.
`LWvv`	LWOI v component with dualtrees. The array is masked with thres. The dimension of LWsc is equal to the dimension of x, if the boundaries are only padded with zeros.
`angle`	Angle of spectrum. The array is masked with thres. The dimension of angle is equal to the dimension of x, if the boundaries are only padded with zeros. Angle ranges between 0 degree and 180 degree.
`thres`	Threshold of LW calculation.
`mask`	Mask defined by thres.
`x`	Ingoing array.
`ts`	Computation time in seconds.

Warning

The input array must be numeric without NA or NaN and maximal of size 1024 x 1024.

Note

This function calculates the locally wavelet-based organization on the basis of dualtree wavelet spectra. To calculate the WOI, modified WOI or LWOI as presented in Brune et al (2020), use the function WOI.

Author(s)

Sebastian Brune

References

Brune, S., Kapp, F., & Friederichs, P. (2018). A wavelet-based analysis of convective organization in ICON large-eddy simulations. Quarterly Journal of the Royal Meteorological Society, 144(717), 2812-2829. Brune, S., Buschow, S., & Friederichs, P. (2020). Observations and high-resolution simulations of convective precipitation organization over the tropical Atlantic. Quarterly Journal

of the Royal Meteorological Society. Brune, S., Buschow, S., & Friederichs, P. (2020). The Local Wavelet-based Organization Index - Quantification, Localization and Classification of Convective Organization from Radar and Satellite Data.

Examples

```
# Random array of dim 230 x 200
x <- array((rnorm(230*200)), dim = c (230, 200))
thres <- 0.1
LWres <- LW(x, thres = thres, Nx = 2^ceiling(log2(max(dim(x)))),
  Ny = 2^ceiling(log2(max(dim(x)))), boundaries = "pad")
print(paste("The LWOI calculation took", LWres$ts, "seconds."))
# plot data
par(mfrow = c(3, 2))
# original data
image(LWres$x, main = "Original data", zlim = 0:1)
# scale component
image(LWres$LWsc, main = "Scale", zlim = 0:1)
# intensity component
image(LWres$LWin, main = "Intensity", zlim = 0:1)
# anisotropy component
image(LWres$LWai, main = "Anisotropy", zlim = 0:1)
# u component
image(LWres$LWuu, main = "u direction", zlim = 0:1)
# v component
image(LWres$LWvv, main = "v direction", zlim = 0:1)
```

---

shiftmat *Shifts the elements of a matrix*

Description

This function shifts the elements of an array to the right and the top.

Usage

```
shiftmat(x, dx = 0, dy = 0)
```

Arguments

x	2D array.
dx	Integer number. Number of grid points to shift the array to the north. Should be smaller than dim(x). Default is 0.
dy	Integer number. Number of grid points to shift the array to the west. Should be smaller than dim(x). Default is 0.

Value

Returns an array with shifted elements.

Author(s)

Sebastian Brune

Examples

```
# shift the matrix dx = 1 and dy = 2 grid points
x <- array(1:48, dim = c(6, 8))
xshift <- shiftmat(x = x, dx = 1, dy = 2)
```

---

wavtra *Performs the wavelet transform*

Description

This function uses parts the wavelet transform of Eckely et al. (2010). We use the DaubExPhase4 wavelet for all calculations. The inverse A matrix is loaded from constants.rda. The resulting value of each transform is written to the centre of mass of the spectrum, which is also saved in constants.rda due to computation time.

Usage

```
wavtra(x)
```

Arguments

`x`	2D array of dimension 2^n x 2^n with n = 4, 5, ... or 12. Periodic boundaries are assumed.

Value

Returns a 3D array with 2^n x 2^n x 3*n wavelet coefficients. The third dimension includes the wavelet coefficients of North-South scales 1-n, East-West scales 1-n and Diagonal scales 1-n.

Author(s)

Sebastian Brune, Sebastian Buschow

References

Eckley, I.A., Nason, G.P. and Treloar, R.L. (2010) Locally stationary wavelet fields with application to the modelling and analysis of image texture. Journal of the Royal Statistical Society (Series C), 59, 595-616.

Eckley, I.A. and Nason, G.P. (2011). LS2W: Implementing the Locally Stationary 2D Wavelet Process Approach in R, Journal of Statistical Software, 43(3), 1-23.

Examples

```
x <- array(1:(2^12), dim = c(2^6, 2^6))
print(dim(x))
```

WOI *Calculates WOI, modified WOI and LWOI*

Description

This function calculates the wavelet-based organization index (WOI) as defined in Brune et al. (2018), a modified version of WOI and the local WOI using DaubExPhase4 wavelet.

Usage

```
WOI(x = x, s = c(1, 3), l = c(4, 7), thres = 0.1, flat = 25,
    verbose = FALSE, periodic = FALSE)
```

Arguments

`x`	2D numeric array with dimensions larger than 16 x 16. dim(x)[1] has not to be equal to dim(x)[1], but NA and/or NaN are not allowed.
`s`	Vector (length 2) of smallest small convective scale and largest small convective scale. Default: s = c(1, 3).
`l`	Vector (length 2) of smallest large convective scale and largest large convective scale. Default: l = c(4, 7).
`thres`	0 or a positive number. Threshold for rain rate. Default is 0.1 mm/h, because we calculate LWOI only for grid points, where rain rate is >= 0.1 mm/h.
`flat`	The number of grid points at each side, which should be smoothed lineraly. Default is 25. For quadratic arrays with dimension 2^n x 2^n boundaries are not smoothed and flat is a dummy variable.
`verbose`	If TRUE, the function prints progress statements and calculation time. Default is FALSE.
`periodic`	If TRUE, the field x is already of size 2^n x 2^n and has perdiodic boundaries. Default is FALSE.

Value

This functions returns a list with follwing elements:

`WOI1orig`	Original WOI1 (but calculated with DaubExPhase4).
`WOI2orig`	Original WOI2 (but calculated with DaubExPhase4).
`WOI3orig`	Original WOI3 (but calculated with DaubExPhase4).
`WOIorig`	Original WOI (but calculated with DaubExPhase4).
`WOI1`	Modified WOI1.
`WOI2`	Modified WOI2.
`WOI3`	Modified WOI3.
`WOI`	Modified WOI.

LWOI1	2D array of LWOI1.
LWOI2	2D array of LWOI2.
LWOI3	2D array of LWOI3.
LWOI	2D array of LWOI.
s	Smallest and largest small convective scale.
l	Smallest and largest large convective scale.
flat	The number of grid points at each side, which are smoothed.
quad	TRUE or FALSE, if the ingoing array is quadratic and 2^n x 2^n.
thres	Threshold of LWOI calculation.
RR	Ingoing array.
ts	Computation time in seconds.

Warning

This function calculates WOI/LWOI only for arrays up to 2048 x 2048. Minimum size is 16 x 16.

Note

This function preprocesses the ingoing array. If x is quadratic with dimension 2^n, the function generates periodic boundaries by mirroring. If x is not 2^n x 2^n, the boundaries are smoothed regarding flat and 0 are added. In this case, the dimensions of the LWOI arrays are 2*n smaller than dim(x).

Author(s)

Sebastian Brune

References

Brune, S., Kapp, F., & Friederichs, P. (2018). A wavelet-based analysis of convective organization in ICON large-eddy simulations. Quarterly Journal of the Royal Meteorological Society, 144(717), 2812-2829.

Examples

```
# Random array of dim 350 x 300
x <- array((rnorm(350*300)), dim = c (350, 300))
s <- c(1, 2)
l <- c(3, 4)
thres <- 0.1
flat <- 25
WOIres <- WOI(x = x, s = s, l = l, thres = thres, flat = flat,
    verbose = TRUE)
# original WOI (Brune et al., 2018)
WOIorig <- WOIres$WOIorig
print(paste("Original WOI:", WOIorig))
# modified WOI
WOI <- WOIres$WOI
```

```
print(paste("Modified WOI:", WOI))
# local WOI
LWOI <- WOIres$LWOI
par(mfrow = c(1, 2))
image(WOIres$RR, main = "Rain")
image(LWOI, main = "LWOI")
```

Index