

Package ‘SurfRough’

May 7, 2026

Title Calculate Surface/Image Texture Indexes

Version 0.0.1.2

Description Methods for the computation of surface/image texture indices using a geostatistical based approach (Trevisani et al. (2023) <[doi:10.1016/j.catena.2023.106927](https://doi.org/10.1016/j.catena.2023.106927)> and Trevisani and Guth (2025) <[doi:10.3390/rs17233864](https://doi.org/10.3390/rs17233864)>). It provides various functions for the computation of surface texture indices (e.g., omnidirectional roughness and roughness anisotropy), including the ones based on the robust MAD estimator. The kernels included in the software permit also to calculate the surface/image texture indices directly from the input surface (i.e., without de-trending) using increments of order 2 and of order 4. It also provides the new radial roughness index (RRI), representing the improvement of the popular topographic roughness index (TRI). The framework can be easily extended with ad-hoc surface/image texture indices.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

BugReports <https://github.com/strevisani/SurfRough/issues>

LazyData true

Depends R (>= 3.5.0), terra

Suggests tinytest

URL <https://github.com/strevisani/SurfRough>,
<https://doi.org/10.5281/zenodo.7132160>

NeedsCompilation yes

Maintainer Sebastiano Trevisani <strevisani@iuav.it>

LinkingTo Rcpp

Author Sebastiano Trevisani [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-8436-7798>>),
Ilich Alexander [ctb] (ORCID: <<https://orcid.org/0000-0003-1758-8499>>),
Zakharko Taras [ctb] (ORCID: <<https://orcid.org/0000-0001-7601-8424>>)

Repository CRAN

Date/Publication 2026-01-27 12:30:17 UTC

Contents

anisoDir	2
anisoDirL	3
anisoR	4
anisoRL	4
CalcMeans	5
CalcMedians	5
circularDispersionGV	6
circularDispersionNV	6
circularEigenNV	7
iqrST	7
k05ck2	8
k1c	9
k1ck2	10
k1ck4	11
k2c	13
k2ck2	14
k4c	15
k6c	16
k8c	17
KernelCircular	18
KernelRectangular	19
Madscan	20
MadscanL	21
Meanscan	22
RRI	23
RRIcore	24
RRIK3	25
RRIk4	26
RRIMax	27
RRIMin	28
stdST	29
TRibi	30
Trik2	31
Trik2.numeric	32
Trik2.SpatRaster	33
Index	34

anisoDir	<i>Calculate the direction of maximum continuity considering 4 directions</i>
----------	---

Description

The input is represented by four rasters with the spatial variability index (e.g., MAD, variogram, etc.) computed in four directions (N-S, NE-SW, E-W, SE-NW)

Usage

anisoDir(N, NE, E, SE)

Arguments

N Spatial variability along N-S direction
 NE Spatial variability along NE-SW direction
 E Spatial variability along E-W direction
 SE Spatial variability along SE-NW direction

Value

A raster with the direction (in degrees, geographical) of maximum continuity

anisoDirL	<i>Calculate the direction of maximum continuity considering 4 directions</i>
-----------	---

Description

The input is represented by a list of rasters with the spatial variability index (e.g., MAD, variogram, etc.) computed in four directions (N-S, NE-SW, E-W, SE-NW)

Usage

anisoDirL(x)

Arguments

x A list of rasters with the spatial variability along 4 directions (see function anisoDir())

Value

A raster with the direction (in degrees, geographical) of maximum continuity

anisoR	<i>Calculate the index of anisotropy considering the spatial variability along 4 directions</i>
--------	---

Description

The input is represented by four rasters with the spatial variability index (e.g., MAD, variogram, etc.) computed in four directions (N-S, NE-SW, E-W, SE-NW)

Usage

```
anisoR(N, NE, E, SE)
```

Arguments

N	Spatial vairability along N-S direction
NE	Spatial vairability along NE-SW direction
E	Spatial vairability along E-W direction
SE	Spatial vairability along SE-NW direction

Value

A raster with the index of anisotropy (min=0 max=1)

anisoRL	<i>Calculate the index of anisotropy considering the spatial variability along 4 directions</i>
---------	---

Description

The input is represented by a list of rasters with the spatial variability index (e.g., MAD, variogram, etc.) computed in four directions (N-S, NE-SW, E-W, SE-NW)

Usage

```
anisoRL(x)
```

Arguments

x	A list of rasters with the spatial variability along 4 directions (see function anisoR())
---	---

Value

A raster with the index of anisotropy (min=0 max=1)

CalcMeans	<i>Calculate the mean of absolute values raised to an exponent found in a search window</i>
-----------	---

Description

With this you can compute variogram and madogram (but remember that for conventional geostatistical indices you need to divide the derived isotropic index by 2!)

Usage

```
CalcMeans(deltas, w, exponent)
```

Arguments

deltas	The values from which calculate the median of absolute values (i.e., directional differences of order K)
w	The moving window used (e.g. w=KernelCircular(3))
exponent	The exponent: increasing the exponent increase the sensitivity to outliers. Set 2 for Variogram and 1 for Madogram.

Value

A raster with the mean of absolute values in the search window

CalcMedians	<i>Calculate the median of absolute values found in a search window for each raster in a list</i>
-------------	---

Description

Calculate the median of absolute values found in a search window for each raster in a list

Usage

```
CalcMedians(deltas, w)
```

Arguments

deltas	A list of rasters with the values from which calculate the median of absolute values (e.g., directional differences of order K)
w	The moving window used (e.g. w=KernelCircular(3))

Value

A list of rasters with the median of absolute values in the search window

`circularDispersionGV` *Compute circular variance of aspect (i.e. of the gradient vector)*

Description

Compute circular variance of aspect (i.e. of the gradient vector)

Usage

```
circularDispersionGV(inraster, window)
```

Arguments

<code>inraster</code>	The DEM/image from which compute the index
<code>window</code>	The moving window adopted for computing the index

Value

The raster with the computed index

Examples

```
# Gradient vector dispersion using a circular search window of radius 3.
library(terra)
dem=rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif", sep=""))
w=KernelCircular(3)
roughGrad=circularDispersionGV(dem,w)
plot(roughGrad)
```

`circularDispersionNV` *Compute circular variance of normal vectors to surface*

Description

Compute circular variance of normal vectors to surface, using the resultant vector length

Usage

```
circularDispersionNV(inraster, window)
```

Arguments

<code>inraster</code>	The DEM/image from which compute the index
<code>window</code>	The moving window adopted for computing the index

Value

The raster with the computed index

Examples

```
#
#Normal vector dispersion using a circular search window of radius 3.
library(terra)
dem=rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif", sep=""))
w=KernelCircular(3)
roughVDR=circularDispersionNV(dem,w)
plot(roughVDR)
```

circularEigenNV	<i>Compute circular variance of normal vectors to surface</i>
-----------------	---

Description

Compute circular variance of normal vectors to surface, using the eigen values (only for testing, very slow)

Usage

```
circularEigenNV(inraster, window)
```

Arguments

inraster	The DEM from which compute the index
window	The moving window adopted for computing the index

Value

The raster with the computed index

iqrST	<i>iqrST: interquartile range in a moving window</i>
-------	--

Description

A function to compute IQR with r method type 7 in a search window. It uses the implemented rcpp version that is many times faster than using focal with IQR() base R function. It provides the same result as ArcGIS Pro. It is intended for computing roughness indices expressed as a robust (differently from standard deviation) estimate of dispersion of local surface parameters (e.g., slope, profile curvature, residual surface, etc.).

Usage

```
iqrST(x, w = 5, ...)
```

Arguments

x	A DEM/image as a SpatRaster
w	Search window (e.g., kernelCircular(3)), default 5x5 window
...	for further use

Value

the IQR of the selected property in the search window (same units of the input)

References

Trevisani, S., Guth, P.L., 2025. Surface Roughness in Geomorphometry: From Basic Metrics Toward a Coherent Framework. Remote Sensing 17, doi:10.3390/rs17233864.

Examples

```
dem= rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif", sep=""))
# iqr of slope in degrees
slope=terrain(dem, v="slope")
w=KernelCircular(3)
w
iqrSlope=iqrST(slope,w=w)
plot(iqrSlope)
```

k05ck2

basic kernels

Description

Kernels for computing directional differences for specific directions and lag distances. These have been constructed using bilinear interpolation for directions out of main axes. The kernels are intended to be used with "Terra" focal functions (i.e., convolution).

Usage

```
k05ck2
```

Format

just matrices.

Source

Sebastiano Trevisani

References

1. Trevisani, S. & Rocca, M. 2015. MAD: Robust image texture analysis for applications in high resolution geomorphometry. *Computers and Geosciences*, vol. 81, pp. 78-92.
2. Trevisani, S. Teza, G., Guth, P., 2023. A simplified geostatistical approach for characterizing key aspects of short-range roughness. *CATENA*, Volume 223, ISSN 0341-8162, <https://doi.org/10.1016/j.catena.2023.10>

Examples

```
#to see kernels (each one is a list with 4 kernels) of order 1
#These should be used with a detrended "surface"
#lag 1 pixel
k1c
#lag 2 pixels
k2c
#lag 4 pixels
k4c
#lag 6 pixels
k6c
#lag 8 pixels
k8c
#kernels of order 2 (differences of differences)
#these can be applied directly without detrending
#lag 05 pixel
k05ck2
#lag 1 pixel
k1ck2
#lag 2 pixels
k2ck2
#kernels of order 4, for filtering curvature
#these can be applied directly without detrending
k1ck4
```

k1c

basic kernels

Description

Kernels for computing directional differences for specific directions and lag distances. These have been constructed using bilinear interpolation for directions out of main axes. The kernels are intended to be used with "Terra" focal functions (i.e., convolution).

Usage

```
k1c
```

Format

just matrices.

Source

Sebastiano Trevisani

References

1. Trevisani, S. & Rocca, M. 2015. MAD: Robust image texture analysis for applications in high resolution geomorphometry. *Computers and Geosciences*, vol. 81, pp. 78-92.
2. Trevisani, S. Teza, G., Guth, P., 2023. A simplified geostatistical approach for characterizing key aspects of short-range roughness. *CATENA*, Volume 223, ISSN 0341-8162, <https://doi.org/10.1016/j.catena.2023.10>

Examples

```
#to see kernels (each one is a list with 4 kernels) of order 1
#These should be used with a detrended "surface"
#lag 1 pixel
k1c
#lag 2 pixels
k2c
#lag 4 pixels
k4c
#lag 6 pixels
k6c
#lag 8 pixels
k8c
#kernels of order 2 (differences of differences)
#these can be applied directly without detrending
#lag 05 pixel
k05ck2
#lag 1 pixel
k1ck2
#lag 2 pixels
k2ck2
#kernels of order 4, for filtering curvature
#these can be applied directly without detrending
k1ck4
```

k1ck2

basic kernels

Description

Kernels for computing directional differences for specific directions and lag distances. These have been constructed using bilinear interpolation for directions out of main axes. The kernels are intended to be used with "Terra" focal functions (i.e., convolution).

Usage

k1ck2

Format

just matrices.

Source

Sebastiano Trevisani

References

1. Trevisani, S. & Rocca, M. 2015. MAD: Robust image texture analysis for applications in high resolution geomorphometry. *Computers and Geosciences*, vol. 81, pp. 78-92.
2. Trevisani, S. Teza, G., Guth, P., 2023. A simplified geostatistical approach for characterizing key aspects of short-range roughness. *CATENA*, Volume 223, ISSN 0341-8162, <https://doi.org/10.1016/j.catena.2023.10>

Examples

```
#to see kernels (each one is a list with 4 kernels) of order 1
#These should be used with a detrended "surface"
#lag 1 pixel
k1c
#lag 2 pixels
k2c
#lag 4 pixels
k4c
#lag 6 pixels
k6c
#lag 8 pixels
k8c
#kernels of order 2 (differences of differences)
#these can be applied directly without detrending
#lag 05 pixel
k05ck2
#lag 1 pixel
k1ck2
#lag 2 pixels
k2ck2
#kernels of order 4,for filtering curvature
#these can be applied directly without detrending
k1ck4
```

k1ck4

basic kernels

Description

Kernels for computing directional differences for specific directions and lag distances. These have been constructed using bilinear interpolation for directions out of main axes. The kernels are intended to be used with "Terra" focal functions (i.e., convolution).

Usage

k1ck4

Format

just matrices.

Source

Sebastiano Trevisani

References

1. Trevisani, S. & Rocca, M. 2015. MAD: Robust image texture analysis for applications in high resolution geomorphometry. *Computers and Geosciences*, vol. 81, pp. 78-92.
2. Trevisani, S. Teza, G., Guth, P., 2023. A simplified geostatistical approach for characterizing key aspects of short-range roughness. *CATENA*, Volume 223, ISSN 0341-8162, <https://doi.org/10.1016/j.catena.2023.10>
3. Trevisani, S., Guth, P.L., 2025. Surface Roughness in Geomorphometry: From Basic Metrics Toward a Coherent Framework. *Remote Sensing* 17. <https://doi.org/10.3390/rs17233864>

Examples

```
#to see kernels (each one is a list with 4 kernels) of order 1
#These should be used with a detrended "surface"
#lag 1 pixel
k1c
#lag 2 pixels
k2c
#lag 4 pixels
k4c
#lag 6 pixels
k6c
#lag 8 pixels
k8c
#kernels of order 2 (differences of differences)
#these can be applied directly without detrending
#lag 05 pixel
k05ck2
#lag 1 pixel
k1ck2
#lag 2 pixels
k2ck2
#kernels of order 4, for filtering curvature
#these can be applied directly without detrending
k1ck4
```

k2c

basic kernels

Description

Kernels for computing directional differences for specific directions and lag distances. These have been constructed using bilinear interpolation for directions out of main axes. The kernels are intended to be used with "Terra" focal functions (i.e., convolution).

Usage

k2c

Format

just matrices.

Source

Sebastiano Trevisani

References

1. Trevisani, S. & Rocca, M. 2015. MAD: Robust image texture analysis for applications in high resolution geomorphometry. *Computers and Geosciences*, vol. 81, pp. 78-92.
2. Trevisani, S. Teza, G., Guth, P., 2023. A simplified geostatistical approach for characterizing key aspects of short-range roughness. *CATENA*, Volume 223, ISSN 0341-8162, <https://doi.org/10.1016/j.catena.2023.10>

Examples

```
#to see kernels (each one is a list with 4 kernels) of order 1
#These should be used with a detrended "surface"
#lag 1 pixel
k1c
#lag 2 pixels
k2c
#lag 4 pixels
k4c
#lag 6 pixels
k6c
#lag 8 pixels
k8c
#kernels of order 2 (differences of differences)
#these can be applied directly without detrending
#lag 05 pixel
k05ck2
#lag 1 pixel
k1ck2
#lag 2 pixels
```

```
k2ck2
#kernels of order 4,for filtering curvature
#these can be applied directly without detrending
k1ck4
```

k2ck2

basic kernels

Description

Kernels for computing directional differences for specific directions and lag distances. These have been constructed using bilinear interpolation for directions out of main axes. The kernels are intended to be used with "Terra" focal functions (i.e., convolution).

Usage

```
k2ck2
```

Format

just matrices.

Source

Sebastiano Trevisani

References

1. Trevisani, S. & Rocca, M. 2015. MAD: Robust image texture analysis for applications in high resolution geomorphometry. *Computers and Geosciences*, vol. 81, pp. 78-92.
2. Trevisani, S. Teza, G., Guth, P., 2023. A simplified geostatistical approach for characterizing key aspects of short-range roughness. *CATENA*, Volume 223, ISSN 0341-8162, <https://doi.org/10.1016/j.catena.2023.10>

Examples

```
#to see kernels (each one is a list with 4 kernels) of order 1
#These should be used with a detrended "surface"
#lag 1 pixel
k1c
#lag 2 pixels
k2c
#lag 4 pixels
k4c
#lag 6 pixels
k6c
#lag 8 pixels
k8c
#kernels of order 2 (differences of differences)
#these can be applied directly without detrending
```

```

#lag 05 pixel
k05ck2
#lag 1 pixel
k1ck2
#lag 2 pixels
k2ck2
#kernels of order 4,for filtering curvature
#these can be applied directly without detrending
k1ck4

```

k4c

basic kernels

Description

Kernels for computing directional differences for specific directions and lag distances. These have been constructed using bilinear interpolation for directions out of main axes. The kernels are intended to be used with "Terra" focal functions (i.e., convolution).

Usage

k4c

Format

just matrices.

Source

Sebastiano Trevisani

References

1. Trevisani, S. & Rocca, M. 2015. MAD: Robust image texture analysis for applications in high resolution geomorphometry. *Computers and Geosciences*, vol. 81, pp. 78-92.
2. Trevisani, S. Teza, G., Guth, P., 2023. A simplified geostatistical approach for characterizing key aspects of short-range roughness. *CATENA*, Volume 223, ISSN 0341-8162, <https://doi.org/10.1016/j.catena.2023.10>

Examples

```

#to see kernels (each one is a list with 4 kernels) of order 1
#These should be used with a detrended "surface"
#lag 1 pixel
k1c
#lag 2 pixels
k2c
#lag 4 pixels
k4c
#lag 6 pixels

```

```
k6c
#lag 8 pixels
k8c
#kernels of order 2 (differences of differences)
#these can be applied directly without detrending
#lag 05 pixel
k05ck2
#lag 1 pixel
k1ck2
#lag 2 pixels
k2ck2
#kernels of order 4, for filtering curvature
#these can be applied directly without detrending
k1ck4
```

k6c

basic kernels

Description

Kernels for computing directional differences for specific directions and lag distances. These have been constructed using bilinear interpolation for directions out of main axes. The kernels are intended to be used with "Terra" focal functions (i.e., convolution).

Usage

k6c

Format

just matrices.

Source

Sebastiano Trevisani

References

1. Trevisani, S. & Rocca, M. 2015. MAD: Robust image texture analysis for applications in high resolution geomorphometry. *Computers and Geosciences*, vol. 81, pp. 78-92.
2. Trevisani, S. Teza, G., Guth, P., 2023. A simplified geostatistical approach for characterizing key aspects of short-range roughness. *CATENA*, Volume 223, ISSN 0341-8162, <https://doi.org/10.1016/j.catena.2023.10>

Examples

```
#to see kernels (each one is a list with 4 kernels) of order 1
#These should be used with a detrended "surface"
#lag 1 pixel
k1c
#lag 2 pixels
k2c
#lag 4 pixels
k4c
#lag 6 pixels
k6c
#lag 8 pixels
k8c
#kernels of order 2 (differences of differences)
#these can be applied directly without detrending
#lag 05 pixel
k05ck2
#lag 1 pixel
k1ck2
#lag 2 pixels
k2ck2
#kernels of order 4,for filtering curvature
#these can be applied directly without detrending
k1ck4
```

k8c

basic kernels

Description

Kernels for computing directional differences for specific directions and lag distances. These have been constructed using bilinear interpolation for directions out of main axes. The kernels are intended to be used with "Terra" focal functions (i.e., convolution).

Usage

k8c

Format

just matrices.

Source

Sebastiano Trevisani

References

1. Trevisani, S. & Rocca, M. 2015. MAD: Robust image texture analysis for applications in high resolution geomorphometry. *Computers and Geosciences*, vol. 81, pp. 78-92.
2. Trevisani, S. Teza, G., Guth, P., 2023. A simplified geostatistical approach for characterizing key aspects of short-range roughness. *CATENA*, Volume 223, ISSN 0341-8162, <https://doi.org/10.1016/j.catena.2023.10>

Examples

```
#to see kernels (each one is a list with 4 kernels) of order 1
#These should be used with a detrended "surface"
#lag 1 pixel
k1c
#lag 2 pixels
k2c
#lag 4 pixels
k4c
#lag 6 pixels
k6c
#lag 8 pixels
k8c
#kernels of order 2 (differences of differences)
#these can be applied directly without detrending
#lag 05 pixel
k05ck2
#lag 1 pixel
k1ck2
#lag 2 pixels
k2ck2
#kernels of order 4,for filtering curvature
#these can be applied directly without detrending
k1ck4
```

KernelCircular

Build a circular moving window

Description

Build a circular moving window

Usage

```
KernelCircular(radius)
```

Arguments

radius The radius of the moving window

Value

A matrix with selected pixels

Examples

```
#A circular moving window with a radius of 3 pixels  
w=KernelCircular(3)  
w
```

KernelRectangular *Build a rectangular kernel of size X x Y*

Description

Build a rectangular kernel of size X x Y

Usage

```
KernelRectangular(lenx, leny)
```

Arguments

- lenx The size in pixels along x
- leny The size in pixels along y

Value

The matrix (square/rectangular) with the selected pixels

Examples

```
#A rectangular moving window 5x5 pixels  
w=KernelRectangular(5,5)  
w
```

 Madscan

 Calculate MAD basic indices

Description

Calculate MAD basic indices considering a specif lag and difference of order K. It computes 3 indices of roughness/image texture: isotropic/omnidirectional; direction of maximum continuity; anisotropy index. The anisotropy index is based on vector dispersion approach: 0 minimum anisotropy; 1 maximum anisotropy. The direction of anisotropy is in degrees according to geographical convention.

Usage

```
Madscan(inRaster, kernels, w)
```

Arguments

<code>inRaster</code>	The DEM/residual-dem/image from which to compute the indices
<code>kernels</code>	The kernels to be used for computing the directional differences (e.g. order 1,2 and 4 for various lags)
<code>w</code>	The moving window adopted for computing the geostatistical index (i.e., MAD)

Value

A list of 3 rasters: 1)isotropic roughness; 2) direction of anisotropy;3)index of anisotropy.

References

1. Trevisani, S. & Rocca, M. 2015. MAD: Robust image texture analysis for applications in high resolution geomorphometry. *Computers and Geosciences*, vol. 81, pp. 78-92, [doi:10.1016/j.cageo.2015.04.003](https://doi.org/10.1016/j.cageo.2015.04.003).
2. Trevisani, S. Teza, G., Guth, P., 2023. A simplified geostatistical approach for characterizing key aspects of short-range roughness. *CATENA*, Volume 223, ISSN 0341-8162, [doi:10.1016/j.catena.2023.106927](https://doi.org/10.1016/j.catena.2023.106927).
3. Trevisani, S., Guth, P.L., 2025. Surface Roughness in Geomorphometry: From Basic Metrics Toward a Coherent Framework. *Remote Sensing* 17, [doi:10.3390/rs17233864](https://doi.org/10.3390/rs17233864).

Examples

```
# MAD for lag 2 with differences of order 2 using a circular search window of radius 3.
# Using differences of order 1, you should
# apply these on a detrended surface/image.
library(terra)
dem=rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif", sep=""))
w=KernelCircular(3)
rough2c=Madscan(dem,k2ck2, w)
#Plot isotropic roughness
```

```
plot(rough2c$IsoRough)
#Plot anisotropy index/strenght
plot(rough2c$AnisoR)
```

MadscanL

Calculate MAD basic indices (version for large files)

Description

Calculate MAD basic indices considering a specif lag and difference of order K. It computes 3 indices of roughness/image texture: isotropic/omnidirectional; direction of maximum continuity; anisotropy index. The anisotropy index is based on vector dispersion approach: 0 minimum anisotropy; 1 maximum anisotropy. The direction of anisotropy is in degrees according to geographical convention. This version clean and remove unused files so as to reduce the consumption of disk space. When non necessary prefer Madscan(). Work is in progress to create a still more memory (ram and disk) efficient function.

Usage

```
MadscanL(inRaster, kernels, w)
```

Arguments

<code>inRaster</code>	The DEM/residual-dem/image from which to compute the indices
<code>kernels</code>	The kernels to be used for computing the directional differences (e.g. order 1,2 and 4 for various lags)
<code>w</code>	The moving window adopted for computing the geostatistical index (i.e., MAD)

Value

A list of 3 rasters: 1)isotropic roughness; 2) direction of anisotropy;3)index of anisotropy.

References

1. Trevisani, S. & Rocca, M. 2015. MAD: Robust image texture analysis for applications in high resolution geomorphometry. *Computers and Geosciences*, vol. 81, pp. 78-92, [doi:10.1016/j.cageo.2015.04.003](https://doi.org/10.1016/j.cageo.2015.04.003).
2. Trevisani, S. Teza, G., Guth, P., 2023. A simplified geostatistical approach for characterizing key aspects of short-range roughness. *CATENA*, Volume 223, ISSN 0341-8162, [doi:10.1016/j.catena.2023.106927](https://doi.org/10.1016/j.catena.2023.106927).
3. Trevisani, S., Guth, P.L., 2025. Surface Roughness in Geomorphometry: From Basic Metrics Toward a Coherent Framework. *Remote Sensing* 17, [doi:10.3390/rs17233864](https://doi.org/10.3390/rs17233864).

Examples

```
# MAD for lag 2 with differences of order 2 using a circular search window of radius 3.
# Using differences of order 1, you should
# apply these on a detrended surface/image.
library(terra)
dem=rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif", sep=""))
w=KernelCircular(3)
rough2c=MadscanL(dem,k2ck2, w)
#Plot isotropic roughness
plot(rough2c$IsoRough)
#Plot anisotropy index/strenght
plot(rough2c$AnisoR)
```

Meanscan	<i>Calculate less robust geostatistical indices (mean of absolute differences raised to an exponent)</i>
----------	--

Description

With this you can compute variogram and madogram (but remember that for conventional geostatistical indices you need to divide the derived isotropic index by 2!). Moreover you can calibrate the exponent in order to filter or enhance hotspots and discontinuities

Usage

```
Meanscan(inRaster, kernels, w, exponent)
```

Arguments

inRaster	The DEM/residual-dem/image from which to compute the indices
kernels	The kernels to be used for computing the directional differences (e.g. order 1 or 2 for various lags)
w	The moving window adopted for computing the geostatistical index (i.e., MAD)
exponent	The exponent: increasing the exponent increase the sensitivity to outliers. Set 2 for Variogram and 1 for Madogram.

Value

A SpatRaster with 3 layers: 1)isotropic roughness; 2) direction of anisotropy; 3)index of anisotropy.

Examples

```
#' Variogram-like for lag 2 with differences of order 2 using a circular search window of radius 3.
# Using differences of order 1, you should
# apply these on a detrended surface/image.
library(terra)
```

```
dem=rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif", sep=""))
w=KernelCircular(3)
rough2c=Meanscan(dem,k2ck2, w,2)
#(divide by two if you need conventional estimator)
plot(rough2c$IsoRough)
```

RRI

*RRI: Radial Roughness index***Description**

Modified TRI, based on increments of order 2 (reducing/removing slope dependence) and correcting for diagonal distance. RRI modifies TRI (topographic ruggedness index) using increments of order 2, symmetrical to the central pixel, so as to reduce/remove the effect of local slope. This version corrects for the diagonal distance using bilinear interpolation. It uses a 5x5 kernel, consequently 12 directional differences of order k=2 are used in the estimation. One could also use a 3x3 kernel using only the 4 differences centered on the central pixel but the metric would be very noisy (see `RRIcore()`). The input is the DEM/image (no need to detrend).

Usage

```
RRI(x, ...)
```

```
## S3 method for class 'numeric'
```

```
RRI(x, ...)
```

```
## S3 method for class 'SpatRaster'
```

```
RRI(x, ..., .method = c("rcpp", "r"))
```

Arguments

x	A DEM/image as a <code>SpatRaster</code> or a vector of numeric values from a focal window in a DEM from which to compute the index
...	reserved for future use
.method	Either <code>r</code> or <code>rcpp</code> (fast batch processing using C++)

Value

RRI (in the same units of input)

References

#' 1) Riley, S. J., S. D. DeGloria, and R. Elliott. 1999. A terrain ruggedness index that quantifies topographic heterogeneity. *Intermountain Journal of Science* 5:23. 2) Wilson, M.F.J., O'Connell, B., Brown, C., Guinan, J.C. & Grehan, A.J. 2007. Multiscale terrain analysis of multibeam bathymetry data for habitat mapping on the continental slope". *Marine Geodesy*, vol. 30, no. 1-2, pp. 3-35. 3) Trevisani S., Teza G., Guth P.L., 2023. Hacking the topographic ruggedness index. *Geomorphology* doi:10.1016/j.geomorph.2023.108838.

Examples

```
library(terra)
dem= rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif",sep=""))
w <- matrix(1, nrow=5, ncol=5)
roughRRI_v1=focal(dem, w=w, fun=RRI)
roughRRI_v2=RRI(dem)
plot(c(roughRRI_v1, roughRRI_v2))
```

RRICore

RRICore: RRI using only the four inner second order directional differences of the RRI kernel

Description

RRICore is like RRI, but it just uses the four inner second order directional differences, using a 3x3 kernel. There are some analogies with Casorati curvature. The input is the DEM/image (no need to detrend).

Usage

```
RRICore(x, ...)

## S3 method for class 'numeric'
RRICore(x, ...)

## S3 method for class 'SpatRaster'
RRICore(x, ..., .method = c("rcpp", "r"))
```

Arguments

x	A DEM/image as a SpatRaster or a vector of numeric values from a focal window in a DEM from which to compute the index
...	reserved for future use
.method	Either r or rcpp (fast batch processing using C++)

Value

RRICore (in the same units of input)

References

Trevisani S., Teza G., Guth P.L., 2023. Hacking the topographic ruggedness index. *Geomorphology*, doi:[10.1016/j.geomorph.2023.108838](https://doi.org/10.1016/j.geomorph.2023.108838).

Examples

```
library(terra)
dem= rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif",sep=""))
roughRRIcore=RRIcore(dem)
plot(roughRRIcore)
```

RRIK3

*RRIK3: Radial Roughness index with differences of order 3***Description**

Extension of RRI using differences of order 3, with a 5x5 kernel. Accordingly, this version filters out a trend of order 2, so it reduces still more the dependence on slope and partially on curvature (for filtering of curvature better to select RRIk4()). The input is the DEM/image (no need to detrend).

Usage

```
RRIK3(x, ...)
```

```
## S3 method for class 'numeric'
RRIK3(x, ...)
```

```
## S3 method for class 'SpatRaster'
RRIK3(x, ..., .method = c("rcpp", "r"))
```

Arguments

x	A DEM/image as a SpatRaster or a vector of numeric values from a focal window in a DEM from which to compute the index
...	reserved for future use
.method	Either r or rcpp (fast batch processing using C++, still to implement)

Value

isotropic roughness (in the same units of input)

References

Trevisani S., Teza G., Guth P.L., 2023. Hacking the topographic ruggedness index. *Geomorphology*, doi:[10.1016/j.geomorph.2023.108838](https://doi.org/10.1016/j.geomorph.2023.108838).

Examples

```
library(terra)
dem= rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif",sep=""))
roughRRIK3=RRIK3(dem)
plot(roughRRIK3)
```

RRIk4*RRIk4: Radial Roughness index with fourth order differences*

Description

RRI based on increments of order 4, permits the filtering of curvature (filters a polynomial of order 3), always using a 5x5 kernel. The input is the DEM/image (no need to detrend).

Usage

```
RRIk4(x, ...)  
  
## S3 method for class 'numeric'  
RRIk4(x, ...)  
  
## S3 method for class 'SpatRaster'  
RRIk4(x, ..., .method = c("rcpp", "r"))
```

Arguments

x	A DEM/image as a SpatRaster or a vector of numeric values from a focal window in a DEM from which to compute the index
...	reserved for future use
.method	Either r or rcpp (fast batch processing using C++)

Value

RRIk4 (in the same units of input)

References

1. Trevisani S., Teza G., Guth P.L., 2023. Hacking the topographic ruggedness index. *Geomorphology*, doi:10.1016/j.geomorph.2023.108838.
2. Trevisani, S., Guth, P.L., 2025. Surface Roughness in Geomorphometry: From Basic Metrics Toward a Coherent Framework. *Remote Sensing* 17, doi:10.3390/rs17233864.

Examples

```
library(terra)  
dem= rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif", sep=""))  
roughRRIk4=RRIk4(dem)  
plot(roughRRIk4)
```

RRIMax*RRIMax: Maximum Radial Roughness index*

Description

Same as RRI but instead of computing the mean of the absolute differences of order 2, the maximum is computed. The input is the DEM/image (no need to detrend).

Usage

```
RRIMax(x, ...)  
  
## S3 method for class 'numeric'  
RRIMax(x, ...)  
  
## S3 method for class 'SpatRaster'  
RRIMax(x, ..., .method = c("rcpp", "r"))
```

Arguments

x	A DEM/image as a SpatRaster or a vector of numeric values from a focal window in a DEM from which to compute the index
...	reserved for future use
.method	Either r or rcpp (fast batch processing using C++)

Value

isotropic roughness (in the same units of input)

References

Trevisani S., Teza G., Guth P.L., 2023. Hacking the topographic ruggedness index. *Geomorphology*, doi:10.1016/j.geomorph.2023.108838.

Examples

```
library(terra)  
dem= rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif", sep=""))  
roughRRIMax=RRIMax(dem)  
plot(roughRRIMax)
```

RRIMin*RRIMin: Minimum Radial Roughness index*

Description

Same as RRI but instead of computing the mean of the absolute differences of order 2, the minimum is computed. The input is the DEM/image (no need to detrend).

Usage

```
RRIMin(x, ...)  
  
## S3 method for class 'numeric'  
RRIMin(x, ...)  
  
## S3 method for class 'SpatRaster'  
RRIMin(x, ..., .method = c("rcpp", "r"))
```

Arguments

x	A DEM/image as a SpatRaster or a vector of numeric values from a focal window in a DEM from which to compute the index
...	reserved for future use
.method	Either r or rcpp (fast batch processing using C++)

Value

isotropic roughness (in the same units of input)

References

Trevisani S., Teza G., Guth P.L., 2023. Hacking the topographic ruggedness index. *Geomorphology*, doi:[10.1016/j.geomorph.2023.108838](https://doi.org/10.1016/j.geomorph.2023.108838).

Examples

```
library(terra)  
dem= rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif",sep=""))  
roughRRIMin=RRIMin(dem)  
plot(roughRRIMin)
```

`stdST`*stdST: standard deviation in a moving window*

Description

A function to compute standard deviation in a moving window. By default it uses the implemented `rcpp` version that is many times faster than using `focal` with `var()` base R function. It provides the same result as ArcGis Pro. It is intended for computing roughness indices expressed as dispersion of local surface parameters (e.g., slope, profile curvature, residual surface, etc.). Whenever there is a NA in the kernel the result is NA. R base `var()` function uses $n-1$ at the denominator, here we use n .

Usage

```
stdST(x, w = 5, ...)
```

Arguments

<code>x</code>	A DEM/image as a <code>SpatRaster</code>
<code>w</code>	Search window (e.g., <code>kernelCircular(3)</code>), default 5x5 window
<code>...</code>	for further use

Value

the STD of the selected property in the search window (same units of the input)

References

Trevisani, S., Guth, P.L., 2025. Surface Roughness in Geomorphometry: From Basic Metrics Toward a Coherent Framework. *Remote Sensing* 17, [doi:10.3390/rs17233864](https://doi.org/10.3390/rs17233864).

Examples

```
library(terra)
dem= rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif",sep=""))
# std of slope in degrees
slope=terrain(dem, v="slope")
w=KernelCircular(3)
w
stdSlope=stdST(slope,w=w)
plot(stdSlope)
```

 TRiBi

TRiBi: TRI with bilinear interpolation along diagonals

Description

TRI is based on DDs of first order (at least in its original definition), however, it does not take into account the different distances between the pixels on the diagonals with respect to the cardinal ones. This version corrects with bilinear interpolation to get the right distance. As TRI, it is a proxy of slope!

Usage

```
TRiBi(x, ...)

## S3 method for class 'numeric'
TRiBi(x, ...)

## S3 method for class 'SpatRaster'
TRiBi(x, ..., .method = c("rcpp", "r"))
```

Arguments

x	A DEM/image as a SpatRaster or a vector of numeric values from a focal window in a DEM from which to compute the index
...	reserved for future use
.method	Either r or rcpp (fast batch processing using C++)

Value

TRI (in the same units of input)

References

Trevisani S., Teza G., Guth P.L., 2023. Hacking the topographic ruggedness index. *Geomorphology*, doi:[10.1016/j.geomorph.2023.108838](https://doi.org/10.1016/j.geomorph.2023.108838).

Examples

```
library(terra)
dem= rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif", sep=""))
roughTRiBi=TRiBi(dem)
plot(roughTRiBi)#proxy of slope!
```

Trik2	<i>Improved TRI (with differences of order 2), reducing/removing slope dependence (for testing purposes)</i>
-------	--

Description

It is essentially a radial roughness index. This has been created for explaining the derivation of RRI, and it is not supposed to be used for doing real analysis. TRIk2 modifies TRI (topographic ruggedness index) using increments of order 2, symmetrical to central pixel, so as to remove/reduce the effect of local slope. This version does not correct for diagonal distance and therefore is mainly for testing/simulation purposes, so in practice the Radial Roughness Index calculated by the RRI function should be used instead. It uses a 5x5 kernel, consequently 12 directional differences of order k (2) are used in the estimation. One could also use a 3x3 kernel using only the 4 differences centered on the central pixel but the metric would be very noisy. The input is the DEM (no need to detrend).

Usage

Trik2(x)

Arguments

x A DEM/image as a SpatRaster or a vector of numeric values from a focal window in a DEM from which to compute the index

Value

isotropic roughness (in the same units of input)

References

1. Riley, S. J., S. D. DeGloria, and R. Elliott. 1999. A terrain ruggedness index that quantifies topographic heterogeneity. *Intermountain Journal of Science* 5:23.
2. Wilson, M.F.J., O'Connell, B., Brown, C., Guinan, J.C. & Grehan, A.J. 2007. Multiscale terrain analysis of multibeam bathymetry data for habitat mapping on the continental slope". *Marine Geodesy*, vol. 30, no. 1-2, pp. 3-35.
3. Trevisani S., Teza G., Guth P.L., 2023. Hacking the topographic ruggedness index. *Geomorphology* [doi:10.1016/j.geomorph.2023.108838](https://doi.org/10.1016/j.geomorph.2023.108838).

Examples

```
library(terra)
dem=rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif", sep=""))
w <- matrix(1, nrow=5, ncol=5)
roughTrik5x5_v1=focal(dem, w=w, fun=Trik2)
roughTrik5x5_v2=Trik2(dem)
plot(c(roughTrik5x5_v1,roughTrik5x5_v2))
```

Trik2.numeric	<i>Improved TRI (with differences of order 2), reducing/removing slope dependence (for testing purposes)</i>
---------------	--

Description

It is essentially a radial roughness index. This has been created for explaining the derivation of RRI, and it is not supposed to be used for doing real analysis. TRIk2 modifies TRI (topographic ruggedness index) using increments of order 2, symmetrical to central pixel, so as to remove the effect of local slope. This version does not correct for diagonal distance and therefore is mainly for testing/simulation purposes, so in practice the Radial Roughness Index calculated by the RRI function should be used instead. It uses a 5x5 kernel, consequently 12 directional differences of order k (2) are used in the estimation. One could also use a 3x3 kernel using only the 4 differences centered on the central pixel but the metric would be very noisy. The input is the DEM (no need to detrend).

Usage

```
## S3 method for class 'numeric'
Trik2(x)
```

Arguments

x A vector of numeric values from a focal window in a DEM from which to compute the index

Value

isotropic roughness (in the same units of input)

References

1. Riley, S. J., S. D. DeGloria, and R. Elliott. 1999. A terrain ruggedness index that quantifies topographic heterogeneity. *Intermountain Journal of Science* 5:23.
2. Wilson, M.F.J., O'Connell, B., Brown, C., Guinan, J.C. & Grehan, A.J. 2007. Multiscale terrain analysis of multibeam bathymetry data for habitat mapping on the continental slope". *Marine Geodesy*, vol. 30, no. 1-2, pp. 3-35.
3. Trevisani S., Teza G., Guth P.L., 2023. Hacking the topographic ruggedness index. *Geomorphology* doi:10.1016/j.geomorph.2023.108838.

Examples

```
library(terra)
dem=rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif", sep=""))
w <- matrix(1, nrow=5, ncol=5)
roughTrik5x5=focal(dem, w=w, fun=Trik2)
plot(roughTrik5x5)
```

Trik2.SpatRaster	<i>Improved TRI (with differences of order 2), reducing/removing slope dependence (for testing purposes).</i>
------------------	---

Description

It is essentially a radial roughness index. This has been created for explaining the derivation of RRI, and it is not supposed to be used for doing real analysis. TRIk2 modifies TRI (topographic ruggedness index) using increments of order 2, symmetrical to central pixel, so as to reduce/remove the effect of local slope. This version does not correct for diagonal distance and therefore is mainly for testing/simulation purposes, so in practice the Radial Roughness Index calculated by the RRI function should be used instead. It uses a 5x5 kernel, consequently 12 directional differences of order k (2) are used in the estimation. One could also use a 3x3 kernel using only the 4 differences centered on the central pixel but the metric would be very noisy. The input is the DEM (no need to detrend).

Usage

```
## S3 method for class 'SpatRaster'
Trik2(x)
```

Arguments

x A DEM as a SpatRaster or a vector of numeric values from a focal window in a DEM from which to compute the index

Value

isotropic roughness (in the same units of input)

References

1. Riley, S. J., S. D. DeGloria, and R. Elliott. 1999. A terrain ruggedness index that quantifies topographic heterogeneity. *Intermountain Journal of Science* 5:23.
2. Wilson, M.F.J., O'Connell, B., Brown, C., Guinan, J.C. & Grehan, A.J. 2007. Multiscale terrain analysis of multibeam bathymetry data for habitat mapping on the continental slope". *Marine Geodesy*, vol. 30, no. 1-2, pp. 3-35.
3. Trevisani S., Teza G., Guth P.L., 2023. Hacking the topographic ruggedness index. *Geomorphology* [doi:10.1016/j.geomorph.2023.108838](https://doi.org/10.1016/j.geomorph.2023.108838).

Examples

```
library(terra)
dem=rast(paste(system.file("extdata", package = "SurfRough"), "/trento1.tif", sep=""))
roughTrik5x5=Trik2(dem)
plot(roughTrik5x5)
```

Index

* datasets

- k05ck2, 8
- k1c, 9
- k1ck2, 10
- k1ck4, 11
- k2c, 13
- k2ck2, 14
- k4c, 15
- k6c, 16
- k8c, 17

- anisoDir, 2
- anisoDirL, 3
- anisoR, 4
- anisoRL, 4

- CalcMeans, 5
- CalcMedians, 5
- circularDispersionGV, 6
- circularDispersionNV, 6
- circularEigenNV, 7

- iqrST, 7

- k05ck2, 8
- k1c, 9
- k1ck2, 10
- k1ck4, 11
- k2c, 13
- k2ck2, 14
- k4c, 15
- k6c, 16
- k8c, 17
- KernelCircular, 18
- KernelRectangular, 19

- Madscan, 20
- MadscanL, 21
- Meanscan, 22

- RRI, 23

- RRIcore, 24
- RRIK3, 25
- RRIk4, 26
- RRIMax, 27
- RRIMin, 28

- stdST, 29

- TRIbi, 30
- Trik2, 31
- Trik2.numeric, 32
- Trik2.SpatRaster, 33