Package ‘geodiv’

February 13, 2020

Type Package

Title Methods for Calculating Gradient Surface Metrics

Version 0.1.1

Author Annie C. Smith, Phoebe Zarnetske, Kyla Dahlin, Adam Wilson, Andrew Latimer

Maintainer Annie C. Smith <annie.smith@dnr.wa.gov>

Description Methods for calculating gradient surface metrics for continuous analysis of landscape features.

Depends R (>= 3.5)

Imports dplyr (>= 0.7.8), phonTools (>= 0.2-2.1), pracma (>= 2.2.2), raster (>= 2.8-4), rgdal (>= 1.4), spatial (>= 7.3-11), tibble (>= 1.4.2), e1071 (>= 1.7-0), sp (>= 1.3-1), rgeos (>= 0.4-3), parallel (>= 3.5.0), sf (>= 0.7-4), zoo (>= 1.8-5), Rcpp (>= 1.0.1)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Suggests testthat, knitr, ggplot2

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-02-13 14:40:08 UTC

R topics documented:

`.deg2rad` *Degree to Radian Conversion*

Description

Converts degree value(s) to radians.

Usage

```
.deg2rad(x)
```

Arguments

`x`	Numeric. Degree value(s).

Value

Numeric of degree value(s).

`.maxdist` *Estimate Maximum Correlation Length*

Description

Internal function to calculates the maximum distances to specified autocorrelation values (e.g., 0.2) of the areal autocorrelation function (AACF). All 180 degrees from the origin of the AACF image are considered for the calculation.

Usage

```
.maxdist(threshold, aacfimg, distimg)
```

Arguments

`threshold`	A number with a value between 0 and 1. Indicates the autocorrelation value to which the rates of decline are measured.
`aacfimg`	A raster of the areal autocorrelation function. This is the AACF raster split in two in terms of height.
`distimg`	A raster of distances to all pixels from the center of the original image. Distances are in meters if original raster was unprojected, and are in map units (usually meters) if raster was projected (see raster::distance documentation for more details).

Value

A list containing the maximum distances from an autocorrelation value of 1 to the specified autocorrelation value < 1. Distances are meters if original raster was unprojected, and are in map units (usually meters) if raster was projected (see raster::distance documentation for more details).

`.mindist` *Estimate Minimum Correlation Length*

Description

Internal function to calculates the minimum distances to specified autocorrelation values (e.g., 0.2) of the areal autocorrelation function (AACF). All 180 degrees from the origin of the AACF image are considered for the calculation.

Usage

```
.mindist(threshold, aacfimg, distimg)
```

Arguments

`threshold`	A number with a value between 0 and 1. Indicates the autocorrelation value to which the rates of decline are measured.
`aacfimg`	A raster of the areal autocorrelation function. This is the AACF raster split in two in terms of height.
`distimg`	A raster of distances to all pixels from the center of the original image. Distances are in meters if original raster was unprojected, and are in map units (usually meters) if raster was projected (see raster::distance documentation for more details).

Value

A list containing the minimum distances from an autocorrelation value of 1 to the specified autocorrelation value < 1. Distances are meters if original raster was unprojected, and are in map units (usually meters) if raster was projected (see raster::distance documentation for more details).

`.rad2deg` *Radian to Degree Conversion*

Description

Converts radian value(s) to degrees.

Usage

```
.rad2deg(x)
```

Arguments

`x`	Numeric. Radian value(s).

Value

Numeric of degree value(s).

`aacf` *Estimate the Areal Autocorrelation Function*

Description

Calculates the areal autocorrelation function (AACF) as the inverse of the Fourier power spectrum. `aacf(x)` returns the AACF in both matrix and raster format.

Usage

```
aacf(x)
```

Arguments

`x`	An n x n raster or matrix.

Value

A raster or matrix representation of the AACF. Both raster and matrix values are normalized so that the maximum is equal to 1.

Examples

```
library(raster)

# import raster image
data(normforest)

# calculate aacf img and matrix
aacf_out <- aacf(normforest)

# plot resulting aacf image
plot(aacf_out)
```

`area_above` *Area Above the Bearing Area Curve*

Description

Calculates the area above the bearing area curve from points `a` to `b`. If a box is drawn around a function with the upper-left at `a` and the bottom-right at `b`, this function extracts the area above the function within the box.

Usage

```
area_above(f, a, b, n = 100)
```

Arguments

`f`	The function for the Bearing Area curve produced by `stats::ecdf()`.
`a`	Numeric. The left x boundary.
`b`	Numeric. The right x boundary.
`n`	Numeric. The number of subdivisions along the function line.

Details

The area under the curve used to calculate area above the curve is calculated as the numerical integral of the Bearing Area function from `a` to `b` using the trapezoid rule with `n` subdivisions. Assume `a` < `b` and `n` is a positive integer.

Value

A numeric value representing the area above the curve with x bounds `a` and `b`.

Examples

```
library(raster)

# import raster image
data(normforest)

# basic values
z <- getValues(normforest)

# calculate cumulative probability density function of surface 'height' (= ndvi)
mod <- ecdf((1 - z))

# valley fluid retention index = void volume in 'valley' zone
Svi <- area_above(f = mod, b = 1, a = 0.8, n = 500)
```

---

bearing_area — *Calculates the Rotated Bearing Area Curve*

Description

Finds a rotated version of the Bearing Area (Abbott-Firestone) curve from a raster or matrix. The resulting function should be rotated 90 degrees clockwise to get the actual Bearing Area curve.

Usage

```
bearing_area(x)
```

Arguments

x	A raster or matrix.

Value

A function describing the rotated Bearing Area curve.

Examples

```
# import raster image
data(normforest)

# find the rotated Bearing Area curve.
ba_func <- bearing_area(normforest)

# rotate the values and re-plot
xval <- environment(ba_func)$y
yval <- (1 - environment(ba_func)$x)
plot(yval ~ xval)
```

bestfitplane *Finds the Best Fit Polynomial Plane*

Description

Finds the best fit polynomial plane for a surface. This function tests least squares polynomial fits with orders of 1 - 3 and determines which order minimizes the error when the fit is subtracted from the original image.

Usage

```
bestfitplane(x)
```

Arguments

`x`	A raster or matrix.

Value

A raster or matrix of the same size as the input with values predicted from the best polynomial fit.

Examples

```
library(raster)

# import raster image
data(orforest)

# find the least squares polynomial plane
poly <- bestfitplane(orforest)

# plot the fit
plot(poly)
```

fftshift *Fourier Transform Shift*

Description

This function serves to shift the zero-frequency component of the Fourier transform to the center of the matrix.

Usage

```
fftshift(x, dim = -1)
```

Arguments

x	An n x n Fourier transform matrix.
dim	Which dimension to shift the matrix. -1 swaps up/down and left/right. 1 swaps up/down. 2 swaps left/right.

Value

An n x n matrix with the zero-frequency component of the Fourier transform in the center.

References

#' This function was created from code posted by rayryeng at: https://stackoverflow.com/questions/38230794/how-to-write-fftshift-and-ifftshift-in-r.

Examples

```
library(raster)

# import raster image
data(normforest)

# convert to matrix form
M <- ncol(normforest)
N <- nrow(normforest)
zmat <- matrix(raster::getValues(normforest), ncol = M, nrow = N, byrow = TRUE)

# calculate fourier transform and shift
ftmat <- fft(zmat)
ftshift <- fftshift(ftmat)

# plot real component
r <- setValues(normforest, Re(ftshift))
plot(r)
```

---

findpeaks — *Find Local Peaks*

Description

Locates local peaks on a raster or matrix. A peak is defined as any pixel where all 8 surrounding pixels have lower values, and the center pixel has a positive value.

Usage

```
findpeaks(x)
```

Arguments

x	A raster or matrix.

Value

A dataframe of local peak locations (x,y) and values (val). The raster or matrix location index (ind), row (row), and column (col) are also listed.

Examples

```
# import raster image
data(normforest)

# locate peaks
peaks <- findpeaks(normforest)

# calculate the summit density (# peaks/area)
N <- ncol(normforest)
M <- nrow(normforest)
Sds <- nrow(peaks) / ((N - 1) * (M - 1))
```

---

findvalleys *Find Local Valleys*

Description

Locates local valleys on a raster or matrix. A valley is defined as any pixel where all 8 surrounding pixels have higher values, and the center pixel has a negative value.

Usage

```
findvalleys(x)
```

Arguments

x	A raster or matrix.

Value

A dataframe of local valley locations (x,y) and values (val). The raster or matrix location index (ind), row (row), and column (col) are also listed.

Examples

```
# import raster image
data(normforest)

# locate peaks and valleys
peaks <- findpeaks(normforest)
valleys <- findvalleys(normforest)

# find top 5 peaks, valleys
top_peaks <- peaks[order(peaks$val, decreasing = TRUE)[1:5],]
```

```
bottom_valleys <- valleys[order(valleys$val)[1:5],]

# calculate the ten-point height
S10z <- (sum(top_peaks$val) + sum(abs(bottom_valleys$val))) / 5
```

find_flat — *Finds the Flattest Part of the Bearing Area Curve*

Description

Locates the flattest x percentage of the Bearing Area curve. Meant to locate the flattest 40 percent of the Bearing Area curve as used in several roughness parameter calculations.

Usage

```
find_flat(x, perc = 0.4)
```

Arguments

x	A raster or matrix.
perc	Numeric between 0 and 1. The percentage of the curve over which to fit the line.

Value

A list containing the equation for the best fit line, the predicted values from that line, the high and low y-intercept values for the intersection points of the line with the Bearing Area curve, and the high and low x-intercept values for the intersection points of the line with the Bearing Area curve.

Examples

```
# import raster image
data(normforest)

# locate the flattest 40% of the bearing area curve
line_data <- find_flat(normforest, perc = 0.4)

# extract the equation of the line
bf_line <- line_data[[1]]
```

fitplane *Calculate a Least Squares Polynomial Plane*

Description

Fits a polynomial plane of order `n` to a raster or matrix.

Usage

```
fitplane(x, order)
```

Arguments

`x`	A raster or matrix.
`order`	Numeric. Indicates the polynomial order to be fit.

Value

A matrix with values predicted from the polynomial fit.

Examples

```
library(raster)

# import raster image
data(orforest)

# find the 2nd order least squares polynomial plane
polyfit <- fitplane(orforest, order = 2)

# create raster of polyfit
x <- setValues(orforest, polyfit)

# plot the fit
plot(x)
```

flatsa *Flattened Surface Area*

Description

Calculates the surface area of a flat raster or matrix with the same x, y bounds as the study surface.

Usage

```
flatsa(x)
```

Arguments

`x`	A raster or matrix.

Details

This function scales both x and y to between 0 and 1. This is done because most satellite data have units where the x, y units do not equal the z units and because the flat surface area is usually compared to the actual surface area. Surface area is calculated over the sample area (N-1, M-1).

Value

A numeric value representing the scaled surface area of a flattened surface with the same x, y bounds.

Examples

```
# import raster image
data(normforest)

# calculate flattened surface area
flatsa(normforest)
```

---

height_ba — *Value of the Bearing Area Curve at a Specified Value*

Description

Determines the value of the bearing area curve for a specific value along the x-axis (`xval`).

Usage

```
height_ba(x, xval)
```

Arguments

`x`	A raster or matrix.
`xval`	Numeric value along the x-axis.

Value

A numeric value of the bearing area function corresponding to `xval`.

Examples

```
# import raster image
data(normforest)

# determine the bearing area function value
# corresponding to an x value of 0.4
val <- height_ba(normforest, 0.4)
```

normforest — *NDVI errors for a portion of southwestern Oregon, USA.*

Description

A raster image of Normalized Difference Vegetation Index (NDVI) errors derived from Landsat data for a small portion of SW Oregon state. This raster was calculated by subtracting the best-fit polynomial plane from the `orforest` values. The best-fit polynomial plane was calculated using `bestfitplane`.

Usage

```
normforest
```

Format

A raster image with 371 x 371 pixels

range -0.5854638 – 0.1585918

bounds -123, -122.9, 43.0002, 43.1

resolution 30m x 30m (0.002694946 degrees)

projection WGS84

scalar 1 ...

Details

NDVI values are derived from Landsat scene path 45, row 30 summarized as the mean NDVI value between June and August 2000 at roughly 30m resolution. Clouds were removed from the Landsat scene before calculating the mean. The image was created using Google Earth Engine in August 2018.

orelevation — *SRTM elevation for a portion of southwestern Oregon, USA.*

Description

A raster image of Shuttle Radar Topography Mission (SRTM) elevation for a portion of southwestern Oregon.

Usage

```
orelevation
```

Format

A raster image with 371 x 371 pixels

range 433 – 1390

bounds -123.0001, -122.9002, 43.00015, 43.10013

resolution 30m x 30m (0.002694946 degrees)

projection WGS84

scalar 1 ...

Details

Elevation values are from the SRTM data for 2000 and are at roughly 30m resolution. The image was created using Google Earth Engine in October 2019.

`orforest` *NDVI for a portion of southwestern Oregon, USA.*

Description

A raster image of Normalized Difference Vegetation Index (NDVI) derived from Landsat data for a small portion of SW Oregon state.

Usage

```
orforest
```

Format

A raster image with 371 x 371 pixels

range 0 – 1

bounds -123, -122.9, 43.0002, 43.1

resolution 30m x 30m (0.002694946 degrees)

projection WGS84

scalar 1 ...

Details

NDVI values are derived from Landsat scene path 45, row 30 summarized as the mean NDVI value between June and August 2000 at roughly 30m resolution. Clouds were removed from the Landsat scene before calculating the mean. The image was created using Google Earth Engine in August 2018.

pad_edges *Extend edges of a raster or matrix.*

Description

Extends edge values of a raster or matrix by a specified number of pixels.

Usage

```
pad_edges(x, size = 11, val = NULL)
```

Arguments

`x`	A raster or matrix.
`size`	Numeric. Number of pixels to add to each side.
`val`	Numeric. If NULL (default), this extends the edge values out. If not null, this value will be used for the extended cells.

Value

A raster with edges padded `size` number of pixels on each edge.

Examples

```
library(raster)

# import raster image
data(normforest)

# crop raster to much smaller area
x <- pad_edges(normforest, 11)
```

plot_ba_curve *Plots the Bearing Area Curve*

Description

Calculates and plots the Bearing Area curve for a raster or matrix using the `bearing_area()` function (with correctly rotated results).

Usage

```
plot_ba_curve(x, divisions = FALSE)
```

Arguments

x	A raster or matrix.
divisions	Logical, defaults to FALSE. If TRUE, divisions of the curve will be plotted. See details section for more information.

Details

If divisions = TRUE, the lines representing the best fit line to the flattest 40 percent of the curve will be shown, as well as both the x and y interception points of that line.

Value

Plots the Bearing Area curve.

Examples

```
# import raster image
data(normforest)

# plot the bearing area curve
plot_ba_curve(normforest, divisions = TRUE)
```

---

remove_plane *Removes the Best Fit Polynomial Plane from a Surface*

Description

Finds the best fit polynomial plane for a surface and subtracts it from the actual values. The remaining surface has positive values where the actual values are higher than the plane and negative values where the actual value are lower than the plane.

Usage

```
remove_plane(x)
```

Arguments

x	A raster or matrix.

Value

A raster or matrix of the same size as the input with values equal to the difference between the original and bestfit plane.

Examples

```
library(raster)

# import raster image
data(orforest)

# remove the least squares polynomial plane
new_rast <- remove_plane(orforest)

# plot
plot(new_rast)
```

---

s10z *Ten-Point Height*

Description

Calculates the average height above the mean surface for the five highest local maxima plus the average height below the mean surface for the five lowest local minima.

Usage

```
s10z(x)
```

Arguments

x	A raster or matrix.

Value

A numeric value representing the ten-point height.

Examples

```
# import raster image
data(normforest)

# calculate ten-point height.
S10z <- s10z(normforest)
```

sa *Calculates the Average Roughness of a Surface*

Description

Finds the average roughness of a surface (Sa) as the absolute deviation of surface heights from the mean surface height. Height is measured as the value of a raster and may not necessarily represent actual height.

Usage

```
sa(x)
```

Arguments

x	A raster or matrix.

Value

A value of average roughness in the units of the original raster or matrix.

Examples

```
# import raster image
data(normforest)

# find the surface roughness
roughness <- sa(normforest)
```

sbi *Surface Bearing Index*

Description

Determines the surface bearing index (Sbi), calculated as the ratio of root mean square roughness (Sq) to height at 5% of bearing area (z05).

Usage

```
sbi(x)
```

Arguments

x	A raster or matrix.

Value

A numeric value representing the surface bearing index.

Examples

```
# import raster image
data(normforest)

# determine the surface bearing index
Sbi <- sbi(normforest)
```

---

sci — *Core Fluid Retention Index*

Description

Determines the core fluid retention index (Sci). This value is the void volume (area under the bearing area curve) in the 'core' zone. See Figure 2a from Kedron et al. (2018) for more details.

Usage

```
sci(x)
```

Arguments

x	A raster or matrix.

Value

A numeric value representing the core fluid retention index.

Examples

```
# import raster image
data(normforest)

# determine the core fluid retention index
Sci <- sci(normforest)
```

scl *Calculate Correlation Length*

Description

Calculates the smallest and largest distances to specified autocorrelation values (e.g., 0.2) of the areal autocorrelation function (AACF). All 180 degrees from the origin of the AACF image are considered for the calculation.

Usage

```
scl(x, threshold = c(0.2, 1/exp(1)), plot = FALSE)
```

Arguments

x	A raster or matrix.
threshold	A numeric vector containing values between 0 and 1. Indicates the autocorrelation values to which the rates of decline are measured.
plot	Logical. Defaults to FALSE. If TRUE, the AACF and lines showing the considered directions of autocorrelation from the origin will be plotted.

Value

A list containing the minimum and maximum distances from an autocorrelation value of 1 to the specified autocorrelation values < 1. Distances are in the units of the x, y coordinates of the raster image. If more than one threshold value is specified, the order of this list will be [minval(t1),minval(t2),maxval(t1),maxval(t2)].

Examples

```
library(raster)

# import raster image
data(normforest)

# crop raster to much smaller area
x <- crop(normforest, extent(-123, -122.99, 43, 43.01))

# calculate aacf img or matrix
aacf_out <- aacf(x)

# estimate the fastest/slowest declines to 0.20 and 0.37 (1/e) autocorrelation
sclvals <- scl(aacf_out)

# calculate Scl20, the minimum distance to an autocorrelation value of 0.2 in the AACF
Scl20 <- sclvals[1]
```

sdc *Height Intervals of the Bearing Area Curve*

Description

Determines the height interval (height distance) for points along the bearing area curve as defined by their x values.

Usage

```
sdc(x, low, high)
```

Arguments

x	A raster or matrix.
low	Numeric value along the x-axis corresponding to the lowest value of interest along the x-axis.
high	Numeric value along the y-axis corresponding to the highest value of interest along the x-axis.

Value

A numeric value of the difference in height of the y values along the bearing area curve corresponding to the specified x values.

Examples

```
# import raster image
data(normforest)

# determine the 10-40% height interval of the
# bearing area curve
val <- sdc(normforest, 0.1, 0.4)
```

sdq *Root Mean Square Slope of Surface*

Description

Calculates the root mean square slope of a raster or matrix surface using the two-point method.

Usage

```
sdq(x)
```

Arguments

`x` A raster or matrix.

Value

A numeric value representing the two-point root mean square slope, Sdq. The units of the returned value are change in z per one unit (pixel).

References

This function is based on the equations found at https://www.ntmdt-si.ru/data/media/files/manuals/image_analisys_p9_nov12

Examples

```
# import raster image
data(normforest)

# calculate root mean square slope
Sdq <- sdq(normforest)
```

---

sdq6 *Root Area Mean Square Slope of Surface*

Description

Calculates the area root mean square slope of a raster or matrix surface using the seven-point method.

Usage

```
sdq6(x)
```

Arguments

`x` A raster or matrix.

Value

A numeric value representing the seven-point root mean square slope, Sdq6. The units of the returned value are change in z per one unit (pixel).

References

This function is based on the equations found at https://www.ntmdt-si.ru/data/media/files/manuals/image_analisys_p9_nov12

Examples

```
# import raster image
data(normforest)

# calculate area root mean square slope
Sdq6 <- sdq6(normforest)
```

`sdr` *Surface Area Ratio*

Description

Calculates the surface area ratio of a raster or matrix. This is the ratio of a flat surface to the actual surface.

Usage

```
sdr(x)
```

Arguments

`x`	A raster or matrix.

Details

This function scales both x and y, as well as the surface value (z), to between 0 and 1 to best match their units. This is done because most satellite data have units where the x, y units do not equal the z units. Surface area is calculated over the sample area (N-1, M-1).

Value

A numeric value representing the surface area ratio.

Examples

```
# import raster image
data(normforest)

# calculate the surface area ratio
Sdr <- sdr(normforest)
```

sds *Summit Density*

Description

Calculates the summit density of a raster or matrix. Summit density is the number of local peaks per unit area.

Usage

```
sds(x)
```

Arguments

`x`	A raster or matrix.

Value

A numeric value representing the summit density.

Examples

```
# import raster image
data(normforest)

# calculate summit density.
Sds <- sds(normforest)
```

sfd *Calculate the fractal dimension of a raster.*

Description

Calculates the 3D fractal dimension of a raster using the triangular prism surface area method.

Usage

```
sfd(x, silent = FALSE)
```

Arguments

`x`	A raster or matrix.
`silent`	Logical. If `FALSE` (default), the function will print warning messages.

Value

A numeric value representing the fractal dimension of the image.

References

Clarke, K.C., 1986. Computation of the fractal dimension of topographic surfaces using the triangular prism surface area method. Computers & Geosciences, 12(5), pp.713-722.

Examples

```
# import raster image
data(normforest)

# calculate the fractal dimension
Sfd <- sfd(normforest)
```

---

sfd_ *Calculate the fractal dimension of a raster (C function).*

Description

Calculates the 3D fractal dimension of a raster using the triangular prism surface area method.

Usage

```
sfd_(mat)
```

Arguments

mat	A matrix.

Value

A numeric value representing the fractal dimension of the image.

References

Clarke, K.C., 1986. Computation of the fractal dimension of topographic surfaces using the triangular prism surface area method. Computers & Geosciences, 12(5), pp.713-722.

Examples

```
# import raster image
data(normforest)

# convert to matrix
mat <- raster::as.matrix(normforest)

# calculate the fractal dimension
Sfd <- sfd_(mat)
```

simpsons — *Simpson's Rule Empirical Area Under a Curve*

Description

Calculates the area below a curve from points a to b. This function is provided for general use.

Usage

```
simpsons(f, a, b, n = 100)
```

Arguments

f	A function.
a	Numeric. The left x boundary.
b	Numeric. The right x boundary.
n	Numeric. The number of subdivisions along the function line.

Details

Note that if y-values are negative, this returns the area above the function line.

Value

A numeric value representing the area under the curve with x bounds a and b.

Examples

```
library(raster)

# import raster image
data(normforest)

# basic values
z <- getValues(normforest)

# calculate cumulative probability density function of surface 'height' (= ndvi)
mod <- ecdf((1 - z))

# calculate integral
int_area <- simpsons(f = mod, b = 1, a = 0.8, n = 500)
```

sk — *Core Roughness Depth*

Description

Determines the core roughness depth (Sk), the height difference between y values of the intersection points of the least mean square line fit to the flattest 40% of the bearing area curve. See Figure 2a from Kedron et al. (2018) for more details.

Usage

```
sk(x)
```

Arguments

x	A raster.

Value

A numeric value representing the core roughness depth of the image.

Examples

```
# import raster image
data(normforest)

# determine the core roughness depth
Sk <- sk(normforest)
```

sku — *Calculates the Kurtosis of Raster Values*

Description

Finds the kurtosis for a distribution of raster or matrix values (Sku). Kurtosis represents the peakedness of the raster surface height distribution. Height is measured as the value of a raster/matrix and may not necessarily represent actual height.

Usage

```
sku(x, excess = TRUE)
```

Arguments

x	A raster or matrix.
excess	Logical, defaults to `TRUE`. If `TRUE`, excess kurtosis is calculated. If `FALSE`, kurtosis is calculated as the difference from the normal distribution.

Value

A numeric value representing kurtosis.

Examples

```
# import raster image
data(normforest)

# find the excess kurtosis of the raster distribution
Sku <- sku(normforest, excess = TRUE)
```

---

slopecalc — *Determines the Slopes Along the Bearing Area Curve*

Description

Calculates the slopes along the bearing area curve of a raster or matrix. Slopes are determined at points x, from point x - h to x + h.

Usage

```
slopecalc(x, h, f)
```

Arguments

x	A vector of x values.
h	Spacing before and after each point. 2h is the distance over which slopes are calculated.
f	Bearing area function as calculated with bearing_area.

Value

A dataframe with the slope for each segment with centerpoint x.

Examples

```
# import raster image
data(normforest)

# find the slopes along the bearing area curve
ba <- bearing_area(normforest)
x <- seq(0, 1, length.out = 100000)
slopes <- slopecalc(x = x, h = 0.01, f = ba)
```

slopemeans — *Determines the Average Slope Along Larger Segments of the Bearing Area Curve*

Description

Calculates the average slope over every segment of a specified percentage length of the total bearing area curve.

Usage

```
slopemeans(slopes, l = 0.4)
```

Arguments

slopes	A dataframe containing all slopes along the bearing area curve, calculated using the slopecalc function.
l	Percentage of the curve over which to calculate mean slope.

Value

A dataframe with the average slope over segments beginning at specified x locations along the bearing area curve. 'slope' represents the mean slope over the segment, 'xstart' is the beginning x location of the segment, and 'xend' is the concluding x location of the segment.

Examples

```
# import raster image
data(normforest)

# find the average slope of segments of the bearing area
# curve.
ba <- bearing_area(normforest)
x <- seq(0, 1, length.out = 10000)
slopes <- slopecalc(x = x, h = 0.01, f = ba)
slopes_forty <- slopemeans(slopes = slopes, l = 0.4)
```

smean — *Calculates the Mean Peak Height of a Surface Image*

Description

Finds the mean height of positive values in the landscape (mean peak height; Smean) for a raster or matrix representing a surface.

Usage

```
smean(x)
```

Arguments

x	A raster or matrix.

Value

A numeric value of mean peak height.

Examples

```
# import raster image
data(normforest)

# find the maximum peak height
Smean <- smean(normforest)
```

---

sph *Calculates the Maximum Peak Height of a Surface Image*

Description

Finds the absolute value of the highest value in the landscape (maximum peak height; Sph) for a raster or matrix representing a surface.

Usage

```
sph(x)
```

Arguments

x	A raster or matrix.

Value

A numeric value of maximum peak height.

Examples

```
# import raster image
data(normforest)

# find the maximum peak height
Sph <- sph(normforest)
```

spk *Reduced Peak Height*

Description

Determines the reduced peak height (Spk), the height difference between the maximum y value of the bearing area curve and the y value of the highest intersection point of the least mean square line fit to the flattest 40% of the bearing area curve. See Figure 2a from Kedron et al. (2018) for more details.

Usage

```
spk(x)
```

Arguments

x A raster or matrix.

Value

A numeric value representing the reduced peak height.

Examples

```
# import raster image
data(normforest)

# determine the reduced peak height
Spk <- spk(normforest)
```

sq *Calculates the Root Mean Square Roughness of a Surface*

Description

Finds the root mean square roughness of a surface (Sq) as the standard deviation of surface heights from the mean surface height. Height is measured as the value of a raster and may not necessarily represent actual height.

Usage

```
sq(x)
```

Arguments

x A raster or matrix.

Value

A value of root mean square roughness in the units of the original raster or matrix.

Examples

```
# import raster image
data(normforest)

# find the surface roughness
roughness <- sq(normforest)
```

---

srw *Radial Wavelength Metrics*

Description

Calculates the dominant radial wavelength, radial wavelength index, and mean half wavelength of the radial Fourier spectrum. See Kedron et al. (2018) for more detailed description.

Usage

```
srw(x, plot = FALSE)
```

Arguments

x	A raster or matrix.
plot	Logical. If TRUE, returns a plot of the amplitude spectrum with lines showing the radii at which Srw, Srwi, and Shw are calculated.

Value

A vector containing numeric values for the dominant radial wavelength, radial wavelength index, and mean half wavelength.

Examples

```
library(raster)

# import raster image
data(normforest)

# calculate metrics
srwvals <- srw(normforest)

# extract each value
Srw <- srwvals[1]
Srwi <- srwvals[2]
Shw <- srwvals[3]
```

ssc — *Mean Summit Curvature*

Description

Calculates the mean summit curvature of a raster or matrix. Mean summit curvature is the average principle curvature of local maximas on the surface.

Usage

```
ssc(x)
```

Arguments

x	A raster or matrix.

Value

A numeric value representing the average curvature of surface peaks.

Examples

```
# import raster image
data(normforest)

# calculate mean summit curvature
Ssc <- ssc(normforest)
```

ssk — *Calculates the Skewness of Raster Values*

Description

Finds the Fisher-Pearson coefficient of skewness for raster or matrix values (Ssk). Skewness represents the asymmetry of the surface height distribution. Height is measured as the value of a raster/matrix and may not necessarily represent actual height.

Usage

```
ssk(x, adj = TRUE)
```

Arguments

x	A raster or matrix.
adj	Logical, defaults to TRUE. If TRUE, the adjusted Fisher-Pearson coefficient of skewness is calculated. Otherwise, the standard coefficient is calculated.

Value

A numeric value representing skewness. # import raster image data(normforest)

find the adjusted coefficient of skewness Ssk <- ssk(normforest, adj = TRUE)

---

std   *Texture Direction Metrics*

Description

Calculates the angle of dominating texture and the texture direction index of the Fourier spectrum image calculated from a raster image (see Kedron et al. 2018).

Usage

```
std(x, plot = FALSE)
```

Arguments

`x`	A raster or matrix.
`plot`	Logical. If `TRUE`, returns a plot of the amplitude spectrum with lines showing directions in which amplitude is summed for the Std and Stdi calculations. Plotting is not possible when input is a matrix.

Value

A vector containing numeric values for the angle of dominating texture and the texture direction index.

Examples

```
library(raster)

# import raster image
data(normforest)

# calculate Std and Stdi
stdvals <- std(normforest)

# extract each value
Std <- stdvals[1]
Stdi <- stdvals[2]
```

stxr *Estimate Texture Aspect Ratio*

Description

Calculates the texture aspect ratio (Str) at defined autocorrelation values. The texture aspect ratio is the ratio of the fastest to the slowest decay lengths of the autocorrelation function to the defined autocorrelation values.

Usage

```
stxr(x, threshold = c(0.2, 1/exp(1)))
```

Arguments

`x`	A raster or matrix.
`threshold`	A vector of autocorrelation values with values between 0 and 1. Indicates the autocorrelation value(s) to which the rates of decline are measured.

Value

A vector with length equal to that of `threshold` containing the texture aspect ratio(s) for the input autocorrelation value(s).

Examples

```
library(raster)

# import raster image
data(normforest)

# crop raster to much smaller area
x <- crop(normforest, extent(-123, -122.99, 43, 43.01))

# estimate the texture aspect ratio for autocorrelation
# thresholds of 0.20 and 0.37 (1/e)
strvals <- stxr(x, threshold = c(0.20, 1 / exp(1)))

# calculate Str20, the texture aspect ratio for
# autocorrelation value of 0.2 in the AACF
Str20 <- strvals[1]
```

surface_area *Surface Area*

Description

Calculates the scaled surface area of a raster or matrix.

Usage

```
surface_area(x)
```

Arguments

x	A raster or matrix.

Details

This function scales both x and y, as well as the surface value (z), to between 0 and 1 to best match their units. This is done because most satellite data have units where the x, y units do not equal the z units. The surface area represents the surface area of the sample area (N-1, M-1).

Note that the surface object may have NA values around the edges, but should not have any missing values within the main area.

Value

A numeric value representing the scaled surface area.

Examples

```
# import raster image
data(normforest)

# calculate surface area
surface_area(normforest)
```

sv *Calculates the Maximum Valley Depth of a Surface Image*

Description

Finds the absolute value of the lowest value in the landscape (maximum valley depth; Sv) for a raster or matrix representing a surface.

Usage

```
sv(x)
```

Arguments

x A raster or matrix.

Value

A numeric value of maximum valley depth.

Examples

```
# import raster image
data(normforest)

# find the maximum valley depth
Sv <- sv(normforest)
```

---

svi *Valley Fluid Retention Index*

Description

Determines the valley fluid retention index (Svi). This value is the void volume (area under the bearing area curve) in the 'valley' zone. See Figure 2a from Kedron et al. (2018) for more details.

Usage

```
svi(x)
```

Arguments

x A raster or matrix.

Value

A numeric value representing the valley fluid retention index.

Examples

```
# import raster image
data(normforest)

# determine the valley fluid retention index
Svi <- svi(normforest)
```

svk *Reduced Valley Depth*

Description

Determines the reduced valley depth (Svk), the height difference between y value of the lowest intersection point of the least mean square line fit to the flattest 40% of the bearing area curve and the minimum y value of the bearing area curve. See Figure 2a from Kedron et al. (2018) for more details.

Usage

```
svk(x)
```

Arguments

x	A raster or matrix.

Value

A numeric value representing the reduced valley depth.

Examples

```
# import raster image
data(normforest)

# determine the reduced valley depth
Svk <- svk(normforest)
```

texture_image *Calculate Texture Metrics per Pixel*

Description

Calculates the various texture metrics over windows centered on individual pixels. This creates a continuous surface of the texture metric.

Usage

```
texture_image(
  x,
  window_type = "square",
  size = 5,
  in_meters = FALSE,
  epsg_proj = 5070,
```

```
  metric,
  args = NULL,
  parallel = TRUE,
  ncores = NULL,
  nclumps = 100
)
```

Arguments

`x`	A raster or matrix. If a raster is given, it will be projected to an equal area projection (given by `epsg_proj` argument).
`window_type`	Character. Type of window, either circular or square.
`size`	Numeric. Size of window, in number of pixels extra on each side, or radius (in meters).
`in_meters`	Logical. Is the size given in meters?
`epsg_proj`	Numeric. Appropriate equal area EPSG code used to crop raster to each circular window.
`metric`	Character. Metric to calculate for each window. Metrics from the geodiv package are listed below.
`args`	List. Arguments from function to be applied over each window (e.g., list(threshold = 0.2)).
`parallel`	Logical. Option to run the calculations in parallel on available cores.
`ncores`	Numeric. If parallel is TRUE, number of cores on which to run the calculations. Defaults to all available, minus 1.
`nclumps`	Numeric. Number of clumps to split the raster or matrix into.

Details

Metrics available from geodiv package:

1. `'sa'`: average surface roughness
2. `'sq'`: root mean square roughness
3. `'s10z'`: ten-point height
4. `'sdq'`: root mean square slope of surface, 2-point method
5. `'sdq6'`: root mean square slope of surface, 7-point method
6. `'sdr'`: surface area ratio
7. `'sbi'`: surface bearing index
8. `'sci'`: core fluid retention index
9. `'ssk_adj'`: adjusted skewness
10. `'ssk'`: skewness
11. `'sku_exc'`: excess kurtosis
12. `'sku'`: kurtosis
13. `'sds'`: summit density

14. 'sfd': 3d fractal dimension
15. 'srw': dominant radial wavelength
16. 'srwi': radial wavelength index
17. 'shw': mean half wavelength
18. 'std': angle of dominating texture
19. 'stdi': texture direction index
20. 'svi': valley fluid retention index
21. 'stxr': texture aspect ratio
22. 'ssc': mean summit curvature
23. 'sv': maximum valley depth
24. 'sph': maximum peak height
25. 'sk': core roughness depth
26. 'smean': mean peak height
27. 'svk': reduced valley depth
28. 'spk': reduced peak height
29. 'scl': correlation length
30. 'sdc': bearing area curve height interval

Value

A raster or list of rasters (if function results in multiple outputs) with pixel values representative of the metric value for the window surrounding that pixel. Note that the raster will always be projected to an equal area projection because calculations are done on matrices with a radius of number of pixels.

Note

The total window size for square windows will be (size * 2) + 1.

Examples

```
library(raster)

# import raster image
data(normforest)

# crop raster to much smaller area
x <- crop(normforest, extent(-123, -122.99, 43, 43.01))

# get a surface of root mean square roughness
sa_img <- texture_image(x = x, window = 'square',
size = 11, metric = 'sa',
parallel = FALSE)

# plot the result
plot(sa_img)
```

window_metric — *Calculate Texture Metric for Single Pixel*

Description

Calculates the various texture metrics over a window centered on an individual pixel.

Usage

```
window_metric(
  x,
  i,
  window_type = "square",
  size = 11,
  rownum,
  colnum,
  metric,
  args = NULL
)
```

Arguments

`x`	A raster or matrix.
`i`	Index of cell at which to calculate the metric.
`window_type`	Character. Type of window, either circular or square.
`size`	Numeric. Radius of window in number of pixels.
`rownum`	Vector of row numbers at which to calculate the metric.
`colnum`	Vector of column numbers at which to calculate the metric.
`metric`	Character. Metric to calculate for each window. Metrics from the geodiv package are listed below.
`args`	List. Arguments from function to be applied over each window (e.g., list(threshold = 0.2)).

Details

Metrics from geodiv package:

1. `'sa'`: average surface roughness
2. `'sq'`: root mean square roughness
3. `'s10z'`: ten-point height
4. `'sdq'`: root mean square slope of surface, 2-point method
5. `'sdq6'`: root mean square slope of surface, 7-point method
6. `'sdr'`: surface area ratio

7. 'sbi': surface bearing index
8. 'sci': core fluid retention index
9. 'ssk_adj': adjusted skewness
10. 'ssk': skewness
11. 'sku_exc': excess kurtosis
12. 'sku': kurtosis
13. 'sds': summit density
14. 'sfd': 3d fractal dimension
15. 'srw': dominant radial wavelength
16. 'srwi': radial wavelength index
17. 'shw': mean half wavelength
18. 'std': angle of dominating texture
19. 'stdi': texture direction index
20. 'svi': valley fluid retention index
21. 'stxr': texture aspect ratio
22. 'ssc': mean summit curvature
23. 'sv': maximum valley depth
24. 'sph': maximum peak height
25. 'sk': core roughness depth
26. 'smean': mean peak height
27. 'svk': reduced valley depth
28. 'spk': reduced peak height
29. 'scl': correlation length
30. 'sdc': bearing area curve height interval

Value

A raster with pixel values representative of the metric value for the window surrounding that pixel.

Note

Note that if calculating the metric at the edge of a raster or matrix, the input raster/matrix must be padded. This can be done using the pad_edges function.

Examples

```
library(raster)

# import raster image
data(normforest)

# crop raster to much smaller area if on a smaller computer
x <- crop(normforest, extent(-123, -122.99, 43, 43.01))
```

```
# get coordinates, rownums, cellnums
pixlist <- seq(1, length(x), 1)
ext_x <- pad_edges(x, size = 4)
rownum <- rowFromCell(x, pixlist) + 4
colnum <- colFromCell(x, pixlist) + 4

# get a surface of root mean square roughness
sq_val <- window_metric(x = x, i = 40, window = 'square',
size = 4, rownum = rownum, colnum = colnum, metric = 'sq')
```

---

zshift *Offset Raster or Matrix Values*

Description

Calculates a matrix of values with a negative or positive, x or y, offset.

Usage

```
zshift(r, xdist = 0, ydist = 0, xrm, yrm, scale = FALSE)
```

Arguments

`r`	A raster or matrix.
`xdist`	Numeric indicating the number and direction (+, -) of columns for the offset.
`ydist`	Numeric indicating the number and direction (+, -) of rows for the offset.
`xrm`	Numeric value or vector indicating the number of columns to be removed from the final matrix. If not set, this value defaults to `xdist`. Positive values remove columns from the right, while negative values remove columns from the left. The absolute value of `xrm` must be >= `abs(xdist)`.
`yrm`	Numeric value or vector indicating the number of rows to be removed from the final matrix. If not set, this value defaults to `ydist`. Positive values remove rows from the bottom, while negative values remove rows from the top. The absolute value must be >= `abs(ydist)`.
`scale`	Logical. Indicates whether or not to scale the values of the raster.

Value

A numeric vector of values created from a matrix of the values with the specified offset. The vector is created from a matrix with `xrm` fewer columns and `yrm` fewer rows than the original raster value matrix.

Examples

```
# import raster image
data(normforest)

# remove right and bottom borders 2 deep
noborder <- zshift(normforest, xdist = 2, ydist = 2)
```

Index