Package ‘bwimage’

July 22, 2025

Type Package

Title Describe Image Patterns in Natural Structures

Version 1.3

Date 2020-04-22

Author Carlos Biagolini-Jr.

Maintainer Carlos Biagolini-Jr.<`c.biagolini@gmail.com`>

Depends stats, utils

Imports jpeg, png

Description A computational tool to describe patterns in black and white images from natural structures. 'bwimage' implemented functions for exceptionally broad subject. For instance, 'bwimage' provide examples that range from calculation of canopy openness, description of patterns in vertical vegetation structure, to patterns in bird nest structure.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2020-04-22 20:24:10 UTC

Contents

`bwimage-package` *Describe Image Patterns in Natural Structures*

Description

A computational tool to describe patterns in black and white images from natural structures. ’bwimage’ implemented functions for exceptionally broad subject. For instance, ’bwimage’ provide examples that range from calculation of canopy openness, description of patterns in vertical vegetation structure, to patterns in bird nest structure.

Details

The DESCRIPTION file:

Package:	bwimage
Type:	Package
Title:	Describe Image Patterns in Natural Structures
Version:	1.3
Date:	2020-04-22
Author:	Carlos Biagolini-Jr.
Maintainer:	Carlos Biagolini-Jr.<c.biagolini@gmail.com>
Depends:	stats, utils
Imports:	jpeg, png
Description:	A computational tool to describe patterns in black and white images from natural structures. ’bwimage’ implen
License:	GPL (>= 2)

Index of help topics:

`aggregation_index`	`Aggregation index calculator`
`altitudinal_profile`	`Highest black pixel by sections`
`bwimage-package`	`Describe Image Patterns in Natural Structures`
`compress`	`Compress square to circle`
`denseness_column`	`Denseness in column sections`
`denseness_row`	`Denseness in row sections`
`denseness_sample`	`Denseness in samples`

`denseness_total`	Denseness for whole image
`heigh_maximum`	Height of the highest black pixel in the image
`heigh_propotion`	Cumulative denseness for each line
`heigh_propotion_test`	Cumulative denseness test
`hole_columm`	Holes description in columns sections
`hole_row`	Holes description in row sections
`hole_section`	Hole finder
`hole_section_data`	Summary of holes information
`image_information`	Summary of image information
`light_gap`	Light gap
`plot_samples`	Plot samples from denseness_sample
`stretch`	stretch circle to square
`threshold_color`	Image to matrix - Single
`threshold_image_list`	Image to matrix - List
`topline`	Top line

A computational tool to describe patterns in black and white images from natural structures.

Author(s)

Carlos Biagolini-Jr.

Maintainer: Carlos Biagolini-Jr.<c.biagolini@gmail.com>

References

Biagolini-Jr C, Macedo RH (2019) bwimage: A package to describe image patterns in natural structures. F1000Research 8 Lambers M (2016) Mappings between sphere, disc, and square. Journal of Computer Graphics Techniques Vol 5:1-21 Nobis M, Hunziker U (2005) Automatic thresholding for hemispherical canopy-photographs based on edge detection. Agricultural and forest meteorology 128:243-250 Shirley P, Chiu K (1997) A low distortion map between disk and square. Journal of graphics tools 2:45-52 Zehm A, Nobis M, Schwabe A (2003) Multiparameter analysis of vertical vegetation structure based on digital image processing. Flora-Morphology, Distribution, Functional Ecology of Plants 198:142-160

Examples

```
bush<-system.file("extdata/bush.JPG", package ="bwimage")
bush_imagematrix<-threshold_color(bush, "jpeg", "proportional",compress_rate = 0.1)
aggregation_index(bush_imagematrix)
```

`aggregation_index` *Aggregation index calculator*

Description

The function aggregation_index calculate the aggregation index. It works for matrix with and without transparent pixel. The aggregation index is a standardized estimation of the average proportion of same-color pixels around each image pixel. First, the proportion of same-color neighboring pixels (SCNP) is calculated (marginal lines and columns are excluded). Next, the SCNP for all pixels are averaged; then, given the proportion of black and white pixels, number of pixels in height and width, and location of transparent pixels (when present), the maximum and minimum possible aggregation indexes are calculated. Finally, the observed aggregation is standardized to a scale where the minimum possible value is set at zero and the maximum value is set at one.

Usage

```
aggregation_index(imagematrix)
```

Arguments

`imagematrix`	The matrix to be analysed.

Value

`adjusted_aggregation`	Standardized aggregation.
`non_adjusted_aggregation`	Observed aggregation.

Author(s)

Carlos Biagolini-Jr.

See Also

threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
# Using aggregation_index to estimate vegetation agregation
bush<-system.file("extdata/bush.JPG", package ="bwimage")
bush_imagematrix<-threshold_color(bush, "jpeg", "proportional", compress_rate = 0.1)
aggregation_index(bush_imagematrix)

# Using aggregation_index to estimate aggregation of nest wall holes
nestwall<-system.file("extdata/bird_nestwall.png", package ="bwimage")
nestwall_imagematrix<-threshold_color(nestwall, "png", "width_fixed", target_width=300)
aggregation_index(nestwall_imagematrix)
```

`altitudinal_profile` *Highest black pixel by sections*

Description

Break the original matrix in a number of section (n_sections), then find the higher black pixel in each image section.

Usage

```
altitudinal_profile(imagematrix, n_sections, height_size)
```

Arguments

`imagematrix`	The matrix to be analysed.
`n_sections`	Break the image in this number of columns.
`height_size`	Real size of image height (in mm, cm, m, etc..).

Value

`Mean`	Height mean of the highest black pixel in sections.
`SD`	Standard deviations of the highest black pixel in sections.
`Size`	Height of the highest black pixel in sections.

Author(s)

Carlos Biagolini-Jr.

References

Zehm et al 2003 Multiparameter analysis of vertical vegetation structure based on digital image processing. Flora-Morphology, Distribution, Functional Ecology of Plants, 198: 142-160.

See Also

threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush,  "jpeg",  "proportional", compress_rate = 0.1)

# Profile  of  highest black pixels on sections of the bush image matrix
altitudinal_profile(bush_imagematrix,n_sections = 10, height_size=100)
# Conclusions:
# i)  the mean height of the highest black pixel is 45.28 cm.
# ii) standard deviation of highest black height is 21.54.
```

`compress` *Compress square to circle*

Description

Compress data from square image to circular in binary matrix

Usage

```
compress(imagematrix, method = "radial", background = NA)
```

Arguments

`imagematrix`	The matrix to be compressed.
`method`	Compress algorithm. Four algorithms (radial, shirley, squircle, and elliptical) are available to stretch the image. The algorithms were adapted from Lambers 2016.
`background`	Code for background cell value. When compressing a squared matrix, corners of the transformed matrix will no have corresponding pixel from original matrix. Thus, the background value will be the value of transformed matrix corners.

Value

A matrix of 0, 1 and NA representing white, black and transparent pixels, respectively.

Author(s)

Carlos Biagolini-Jr.

References

Lambers 2016 Mappings between Sphere, Disc, and Square. Journal of Computer Graphics Techniques, 5(2): 1-21.

Examples

```
img_location <- system.file("extdata/chesstable.png",package ="bwimage")
image_matrix<- threshold_color(img_location,"png", "frame_fixed",target_width = 50,target_height=50)
compress(image_matrix,method="radial")
```

`denseness_column` *Denseness in column sections*

Description

Calculate the denseness (proportion of black pixel in relation to the total number of pixels) for a given number of sections (n_sections). n_sections should be set as a number, in this situation denseness_column will break the original matrix in slices, and apply denseness_total function for each section. For instance, in a matrix of 1000x1000 if n_sections = 10, it will break to 10 sections of 1000x100 and analyze it. In other words, the sections will be the following sections of the original matrix [1:1000, 1:100] ,[1:1000,101:200] ,[1:1000,201:300] ,[1:1000,301:400] ,[1:1000,401:500] ,[1:1000,501:600] ,[1:1000,601:700] ,[1:1000,701:800] ,[1:1000,801:900] ,[1:1000,901:1000]. The default for parameter n_sections is "all", it will calculate denseness for each column of pixel. In other words, it will break the image in a number of section equal to the image pixel width.

Usage

```
denseness_column(imagematrix, n_sections = "all")
```

Arguments

`imagematrix`	The matrix to be analysed.
`n_sections`	Break the image in this number of columns.

Value

`Denseness`	Denseness of each column section.
`Mean`	Mean of column sections denseness.
`SD`	standard deviations of column sections denseness.

Author(s)

Carlos Biagolini-Jr.

References

Zehm et al 2003 Multiparameter analysis of vertical vegetation structure based on digital image processing. Flora-Morphology, Distribution, Functional Ecology of Plants, 198: 142-160.

See Also

denseness_total threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush,  "jpeg",  "proportional", compress_rate = 0.1)

# Calculate vegetation denseness in 20 column sections
denseness_column(bush_imagematrix,20)
```

`denseness_row` *Denseness in row sections*

Description

Calculate the denseness (proportion of black pixel in relation to the total number of pixels) for a given number of sections (n_sections). n_sections should be set as a number, in this situation denseness_row will break the original matrix in slices, and apply denseness_total function for each section. For instance, in a matrix of 1000x1000 if n_sections = 10, it will break to 10 sections of 100x1000 and analyze it. In other words, the sections will be the following sections of the original matrix [1:100, 1:1000] , [101:200, 1:1000] , [201:300, 1:1000] , [301:400, 1:1000] , [401:500, 1:1000] , [501:600, 1:1000] , [601:700, 1:1000] , [701:800, 1:1000] , [801:900, 1:1000] , [901:1000, 1:1000] .The default for parameter n_sections is "all", it will calculate denseness for each row of pixel. In other words, it will break the image in a number of section equal to the image pixel height.

Usage

```
denseness_row(imagematrix, n_sections = "all")
```

Arguments

`imagematrix`	The matrix to be analysed.
`n_sections`	Break the image in this number of rows.

Value

`Denseness`	Denseness of each row section.
`Mean`	Mean of row sections denseness.
`SD`	standard deviations of row sections denseness.

Author(s)

Carlos Biagolini-Jr.

References

Zehm et al 2003 Multiparameter analysis of vertical vegetation structure based on digital image processing. Flora-Morphology, Distribution, Functional Ecology of Plants, 198: 142-160.

See Also

denseness_total threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush,  "jpeg", "proportional",compress_rate = 0.1)

# Calculate vegetation denseness in 20 row sections
denseness_row(bush_imagematrix, n_sections = 20)
```

`denseness_sample` *Denseness in samples*

Description

Calculate the denseness (proportion of black pixel in relation to the total number of pixels) for a given number of samples.

Usage

```
denseness_sample(imagematrix, width_size, height_size, sample_width,
  sample_height, method = "random", sample_shape = "rectangle",
  n_samples = 10, n_sample_horizontal = 10, n_sample_vertical = 1,
  proportion_horizontal = 1, proportion_vertical = 1,
  aligin_horizontal = "center", aligin_vertical = "bottom")
```

Arguments

`imagematrix`	The matrix to be analysed.
`width_size`	Real size of image width (in mm, cm, m, etc..).
`height_size`	Real size of image height (in mm, cm, m, etc..).
`sample_width`	Width of sample area.
`sample_height`	Height of sample area.
`method`	Method for sample ("random" or "uniform").
`sample_shape`	The shape of sample unity ("rectangle" or "ellipse"). See plot_samples function.
`n_samples`	Defines the number of samples, when sample_shape="random".
`n_sample_horizontal`	Defines the number of samples column, when sample_shape=" uniform".
`n_sample_vertical`	Defines the number of samples lines, when sample_shape=" uniform".
`proportion_horizontal`	Range from 0 to 1. Represent the proportion of horizontal plane to be sample. If proportion_horizontal=1 (default) all columns beacome potentially sample.

`proportion_vertical`	Range from 0 to 1. Represent the proportion of vertical plane to be sample. If proportion_vertical=1 (default) all lines become potentially sample.
`aligin_horizontal`	Define horizontal align. Three options are available: "center", "left" or "right".
`aligin_vertical`	Define vertical align. Three options are available: "middle","bottom" or "top".

Value

`Sample_denseness`	Proportion of black pixels in samples. It do not take into account transparent pixels (when present)..
`Height`	Height of each sample (in mm, cm, m, etc. ..). Central point used as reference.
`Distance(left)`	Distance ti the left side of each sample (in mm, cm, m, etc. ..). Central point used as reference.
`Matrix(line)`	Imagem matrix line coordinates.
`Matrix(column)`	Imagem matrix column coordinates.

Author(s)

Carlos Biagolini-Jr.

See Also

plot_samples

Examples

```
# Get a matrix from your image. Here  examples provided by bwimage package.

bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush<-threshold_color(bush,  "jpeg", "proportional",compress_rate = 0.1)
denseness_sample(bush, width_size=100, height_size=100, sample_width=5, sample_height=5)
```

`denseness_total` *Denseness for whole image*

Description

Proportion of black pixels in relation to all pixels. It do not take into account transparent pixels (when present).

Usage

```
denseness_total(imagematrix)
```

Arguments

`imagematrix`	The matrix to be analysed.

Value

Proportion of black pixels in relation to all pixels. It do not take into account transparent pixels (when present).

Author(s)

Carlos Biagolini-Jr.

References

Zehm et al 2003 Multiparameter analysis of vertical vegetation structure based on digital image processing. Flora-Morphology, Distribution, Functional Ecology of Plants, 198: 142-160.

See Also

threshold_color

Examples

```
# Get a matrix from your image. Here  examples provided by bwimage package.

# I) Calculate vegetation denseness
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush,  "jpeg", "proportional",compress_rate = 0.1)
denseness_total(bush_imagematrix)

# II) Calculate canopy openness
# Convert image into binary matrix
canopy<-system.file("extdata/canopy.JPG",package ="bwimage")
canopy_matrix<-threshold_color(canopy,"jpeg", compress_method="proportional",compress_rate=0.1)
1-denseness_total(canopy_matrix) # canopy openness
```

`heigh_maximum` *Height of the highest black pixel in the image*

Description

Find the higher black pixel in the whole image.

Usage

```
heigh_maximum(imagematrix, height_size)
```

Arguments

`imagematrix`	The matrix to be analysed.
`height_size`	Real size of image width (in mm, cm, m, etc..).

Value

Height of the highest black pixel. It is scaleted for the real size (in mm, cm, m, etc..) based in the information from argument height_size.

Author(s)

Carlos Biagolini-Jr.

References

Zehm et al 2003 Multiparameter analysis of vertical vegetation structure based on digital image processing. Flora-Morphology, Distribution, Functional Ecology of Plants, 198: 142-160.

See Also

threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush,  "jpeg",  "proportional", compress_rate = 0.1)

# Calculate height of the highest black pixel in the bush image matrix
heigh_maximum(bush_imagematrix,height_size=100)
# Conclusions: The highest vegetation unit ,i.e. highest black pixel, is 84.4 cm above ground.
```

`heigh_propotion` *Cumulative denseness for each line*

Description

Proportion of black pixel below each matrix line.

Usage

```
heigh_propotion(imagematrix)
```

Arguments

`imagematrix`	The matrix to be analysed.

Author(s)

Carlos Biagolini-Jr.

References

Zehm et al 2003 Multiparameter analysis of vertical vegetation structure based on digital image processing. Flora-Morphology, Distribution, Functional Ecology of Plants, 198: 142-160.

See Also

threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush,  "jpeg",  "proportional", compress_rate = 0.1)

# Proportion of black pixel below each matrix line.
heigh_propotion(bush_imagematrix)
```

`heigh_propotion_test` *Cumulative denseness test*

Description

Find the height which a given proportion of black pixel is found.

Usage

```
heigh_propotion_test(imagematrix, proportion, height_size)
```

Arguments

`imagematrix`	The matrix to be analysed.
`proportion`	Proportion of denseness to test.
`height_size`	Real size of image height (in mm, cm, m, etc..).

Author(s)

Carlos Biagolini-Jr.

References

Zehm et al 2003 Multiparameter analysis of vertical vegetation structure based on digital image processing. Flora-Morphology, Distribution, Functional Ecology of Plants, 198: 142-160.

See Also

threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush,  "jpeg",  "proportional", compress_rate = 0.1)

# See the proportion of black pixels (1) below each bush image matrix row
heigh_propotion_test(bush_imagematrix,0.75,100)
# Conclusion: in this imagem, 75 percent of the vegetation is hold below 31.2 cm.
```

`hole_columm` *Holes description in columns sections*

Description

Summary information of holes in a given number of columns (n_sections). n_sections must be set as a number, in this situation hole_columm will sample columns, and apply hole_section_data function for each section. Next, all results will be display on hole_columm output. Example of how column sample works: in a matrix of 250x250 if n_sections = 5 , it will sample columns 1,51,101,151, and 201 and analyze it. In other words, the sections will be following sections of the original matrix [1:250,1] , [1:250,51], [1:250,101], [1:250,151], [1:250,201]. The default for parameter n_sections is "all", it will calculate hole_section_data for each column of pixel. In other words, it will break the image in a number of section equal to the image pixel width.

Usage

```
hole_columm(imagematrix, color = 0, n_sections = "all")
```

Arguments

`imagematrix`	The matrix to be analysed.
`color`	Color of the hole (0 or 1).
`n_sections`	Sample this number of columns.

Value

`N`	Number of sections.
`Mean`	Mean sections size.
`SD`	standard deviations of sections size.
`Min`	Minimum sections size sections size.
`Max`	Maximum sections size.
`LH`	Stratum with largest hole count.

Author(s)

Carlos Biagolini-Jr.

References

Zehm et al 2003 Multiparameter analysis of vertical vegetation structure based on digital image processing. Flora-Morphology, Distribution, Functional Ecology of Plants, 198: 142-160.

See Also

hole_section_data threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush,  "jpeg",  "proportional", compress_rate = 0.1)

# Information of white (i.e. 0s in matrix) holes in 5 columns uniformly sample among matrix.
hole_columm(bush_imagematrix, n_sections=5 )

# Information of black (i.e. 1s in matrix) holes in 20 columns uniformly sample among matrix.
hole_columm(bush_imagematrix, n_sections=20 )
```

`hole_row` *Holes description in row sections*

Description

Summary information of holes in a given number of rows (n_sections). n_sections must be set as a number, in this situation hole_row will sample rows, and apply hole_section_data function for each section. Next, all results will be display on hole_columm output. Example of how row sample works: in a matrix of 250x250 if n_sections = 5 , it will sample rows 1,51,101,151, and 201 and analyze it. In other words, the sections will be following sections of the original matrix [1,1:250] , [51,1:250] , [101,1:250] , [151,1:250] , [201,1:250]. The default for parameter n_sections is "all", it will calculate hole_section_data for each row of pixel. In other words, it will break the image in a number of section equal to the image pixel height.

Usage

```
hole_row(imagematrix, color = 0, n_sections = "all")
```

Arguments

`imagematrix`	The matrix to be analysed.
`color`	Color of the hole (0 or 1).
`n_sections`	Sample this number of rows.

Value

`N`	Number of sections.
`Mean`	Mean sections size.
`SD`	standard deviations of sections size.
`Min`	Minimum sections size sections size.
`Max`	Maximum sections size.
`LH`	Stratum with largest hole count.

Author(s)

Carlos Biagolini-Jr.

See Also

hole_section_data threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush,  "jpeg",  "proportional", compress_rate = 0.1)

# Information of white (i.e. 0s in matrix) holes in 10 rows uniformly sample among matrix.
hole_row(bush_imagematrix, n_sections=10)

# Information of black (i.e. 1s in matrix) holes in 15 rows uniformly sample among matrix.
hole_row(bush_imagematrix, n_sections=15)
```

`hole_section` *Hole finder*

Description

Description of when a sequence of same color pixel start and end.

Usage

```
hole_section(section)
```

Arguments

`section`	Section to be analysed.

Value

Description of start and end of each same color sequence

Author(s)

Carlos Biagolini-Jr.

References

Zehm et al 2003 Multiparameter analysis of vertical vegetation structure based on digital image processing. Flora-Morphology, Distribution, Functional Ecology of Plants, 198: 142-160.

See Also

hole_section_data threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush,  "jpeg",  "proportional", compress_rate = 0.1)

# Find pixel hole sections in the column 200 of bush image
hole_section(bush_imagematrix[,200])

# Find pixel hole sections in the row 250 of bush image
hole_section(bush_imagematrix[250,])
```

`hole_section_data` *Summary of holes information*

Description

Summary information of holes of a given color in a given section. Result unit is the number of cell.

Usage

```
hole_section_data(section, color = 0)
```

Arguments

`section`	Section to be analysed.
`color`	Color of the hole (0 or 1).

Value

`N`	Number of hole sections
`Mean`	Mean size of hole sections
`SD`	Standard deviation of hole sections size
`Min`	Minimum size of hole sections
`Max`	Maximum size of hole sections

Author(s)

Carlos Biagolini-Jr.

References

Zehm et al 2003 Multiparameter analysis of vertical vegetation structure based on digital image processing. Flora-Morphology, Distribution, Functional Ecology of Plants, 198: 142-160.

See Also

hole_section threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush,  "jpeg",  "proportional", compress_rate = 0.1)

# Detail information of white (0) holes sections in the column 200 of bush image
hole_section_data(bush_imagematrix[,200], color = 0)

# Detail information of black (1) holes sections in the row 250 of bush image
hole_section_data(bush_imagematrix[250,], color = 1)
```

`image_information` *Summary of image information*

Description

Provide the information of: number of black, white and transparent pixels, total number of pixels, height and width size.

Usage

```
image_information(imagematrix)
```

Arguments

`imagematrix`	The matrix to be analysed.

Value

`Black`	Number of black pixels
`White`	Number of white pixels
`Transparent`	Number of transparent pixels
`Total`	Total number of pixels
`Height`	Size in height
`Width`	Size in width

Author(s)

Carlos Biagolini-Jr.

See Also

threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush,  "jpeg", "proportional",compress_rate = 0.1)
image_information(bush_imagematrix)
```

`light_gap` *Light gap*

Description

Left and right distances from first black pixel to image edge.

Usage

```
light_gap(imagematrix, width_size = NA, scale = TRUE)
```

Arguments

`imagematrix`	The matrix to be analysed
`width_size`	Real size of image width (in mm, cm, m, etc..).
`scale`	If FALSE do not ajust the output for real size.

Value

Distances without black pixel in each side of the picture

Author(s)

Carlos Biagolini-Jr.

References

Zehm et al 2003 Multiparameter analysis of vertical vegetation structure based on digital image processing. Flora-Morphology, Distribution, Functional Ecology of Plants, 198: 142-160.

See Also

threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush,  "jpeg",  "proportional", compress_rate = 0.1)

# Calculate vegetation Light gap in the bush image matrix
light_gap(bush_imagematrix,width_size=100)
# Conclusion: there is no light gap on both sides of bush image.
```

`plot_samples` *Plot samples from denseness_sample*

Description

Plot samples from denseness_sample.

Usage

```
plot_samples(imagematrix, central_lines, central_collumns, width_size,
  height_size, sample_width, sample_height, sample_shape)
```

Arguments

`imagematrix`	The matrix to be analysed.
`central_lines`	Lines data (i.e. "Matrix(line)") provided by denseness_sample
`central_collumns`	Collumns data (i.e. "Matrix(column)") provided by denseness_sample
`width_size`	Real size of image width (in mm, cm, m, etc..).
`height_size`	Real size of image height (in mm, cm, m, etc..).
`sample_width`	Width of sample area.
`sample_height`	Height of sample area.
`sample_shape`	Inform the shape of sample unity used ("rectangle" or "ellipse"). See denseness_sample function.

Value

Plot of the analysed matrix (black and white) and sample locations (red).

Author(s)

Carlos Biagolini-Jr.

See Also

denseness_sample

Examples

```
# Get a matrix from your image. Here  examples provided by bwimage package.

bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush<-threshold_color(bush,  "jpeg", "proportional",compress_rate = 0.1)
a<-denseness_sample(bush, width_size=100, height_size=100, sample_width=5, sample_height=5)
plot_samples(bush, a[,4],a[,5], 100,100, 5, 5,"rectangle")
```

`stretch` *stretch circle to square*

Description

Stretch data from circular image to square in binary matrix

Usage

```
stretch(imagematrix, method = "radial")
```

Arguments

`imagematrix`	The matrix to be stretched.
`method`	Stretch algorithm. Four algorithms (radial, shirley, squircle, and elliptical) are available to stretch the image. The algorithms were adapted from Lambers 2016.

Value

A matrix of 0, 1 and NA representing white, black and transparent pixels, respectively.

Author(s)

Carlos Biagolini-Jr.

References

Lambers 2016 Mappings between Sphere, Disc, and Square. Journal of Computer Graphics Techniques, 5(2): 1-21.

Examples

```
img_location <- system.file("extdata/chesstable.png",package ="bwimage")
image_matrix<- threshold_color(img_location,"png", "frame_fixed",target_width = 50,target_height=50)
stretch(image_matrix,method="radial")
```

`threshold_color` *Image to matrix - Single*

Description

Convert a single image into a matrix

Usage

```
threshold_color(filename, filetype = "jpeg", compress_method = "none",
  compress_rate = 1, target_width = 100, target_height = 100,
  threshold_value = 0.5, transparency_regulation = 0.5,
  channel = "rgb")
```

Arguments

`filename`	Name of the file to be load - ex: "Figure01.JPG".
`filetype`	Type of the file to be load. Compatible file types: ".JPGE", ".JPG" or ".PNG".
`compress_method`	For high resolution files, i.e. numbers of pixels in width and height, it is suggested to reduce the resolution to create a smaller matrix, it strongly reduce GPU usage and time necessary to run analyses. On the other hand, by reducing resolution, it will also reduce the accuracy of data description. The available methods for image reduction are: i) frame_fixed, which resamples images to a desired target width and height; ii) proportional, which resamples the image by a given ratio provided in the argument "proportion"; iii) width_fixed, which resamples images to a target width, and also reduces the image height by the same factor. For instance, if the original file had 1000 pixels in width, and the new width_was set to 100, height will be reduced by a factor of 0.1 (100/1000); and iv) height_fixed, analogous to width_fixed, but assumes height as reference.
`compress_rate`	Compress rate to by apply if compress_method=proportional. Note: it should be ser as number range from 0 to 1 .
`target_width`	Target width to be used if compress_method=frame_fixed or compress_method= width_fixed.
`target_height`	Target height to be used if compress_method=frame_fixed or compress_method= height_fixed.
`threshold_value`	For each pixel, the intensity of color channels (red, green and blue) are averaged and compared to a threshold_value (threshold). If the average intensity is less than the threshold_value (default is 0.5) the pixel will be set as black, otherwise it will be white. See channel argument.
`transparency_regulation`	For PNG images, the alpha channel is used to set transparent pixels, i.e. alpha channel values above transparency_regulation (a threshold) will set the pixel as transparent, default is 0.5. NOTE: In the data matrix the value 1 represents black pixels, 0 represents white pixels and NA represents transparent pixels.

`channel`	RGB channel to be considered in threshold. If channel=RGB (default), the intensity of red, green and blue is averaged and compared to threshold_value. If the average intensity is less than the threshold_value (default is 50 If only one channel is defined ("R" for red, "G" for green, and "B" for blue), the average intensity selected channel compared direct to the threshold_value value.

Value

A matrix of 0, 1 and NA representing white, black and transparent pixels, respectively.

Author(s)

Carlos Biagolini-Jr.

Examples

```
bush<-system.file("extdata/bush.JPG",package ="bwimage")
threshold_color(bush,"jpeg", "frame_fixed",target_width = 15,target_height=15)

# For your images, if the file is in the working directory type:
# threshold_color("FILE_NAME.EXTENSION", filetype ="FILE_EXTENSION")
# or, if the file is in the other directory:
# threshold_color("C:/PATH TO FILE FOLDER/YOUR_FILE_NAME.EXTENSION", "FILE_EXTENSION")
```

`threshold_image_list` *Image to matrix - List*

Description

Convert two or more images into a list of matrices

Usage

```
threshold_image_list(list_names, filetype = "jpeg",
  compress_method = "none", compress_rate = 1, target_width = 100,
  target_height = 100, threshold_value = 0.5,
  transparency_regulation = 0.5, channel = "rgb")
```

Arguments

`list_names`	An object contains the names of the files.
`filetype`	Type of the file to be load. Compatible file types: ".JPGE", ".JPG" or ".PNG".
`compress_method`	For high resolution files, i.e. numbers of pixels in width and height, it is suggested to reduce the resolution to create a smaller matrix, it strongly reduce GPU usage and time necessary to run analyses. On the other hand, by reducing resolution, it will also reduce the accuracy of data description. The available methods for image reduction are: i) frame_fixed, which resamples images to a

	desired target width and height; ii) proportional, which resamples the image by a given ratio provided in the argument "proportion"; iii) width_fixed, which resamples images to a target width, and also reduces the image height by the same factor. For instance, if the original file had 1000 pixels in width, and the new width_was set to 100, height will be reduced by a factor of 0.1 (100/1000); and iv) height_fixed, analogous to width_fixed, but assumes height as reference.
`compress_rate`	Compress rate to by apply if compress_method=proportional. Note: it should be ser as number range from 0 to 1 .
`target_width`	Target width to be used if compress_method=frame_fixed or compress_method= width_fixed.
`target_height`	Target height to be used if compress_method=frame_fixed or compress_method= height_fixed.
`threshold_value`	For each pixel, the intensity of color channels (red, green and blue) are averaged and compared to a threshold_value (threshold). If the average intensity is less than the threshold_value (default is 0.5) the pixel will be set as black, otherwise it will be white. See channel argument.
`transparency_regulation`	For PNG images, the alpha channel is used to set transparent pixels, i.e. alpha channel values above transparency_regulation (a threshold) will set the pixel as transparent, default is 0.5. NOTE: In the data matrix the value 1 represents black pixels, 0 represents white pixels and NA represents transparent pixels.
`channel`	RGB channel to be considered in threshold. If channel=RGB (default), the intensity of red, green and blue is averaged and compared to threshold_value. If the average intensity is less than the threshold_value (default is 50 If only one channel is defined ("R" for red, "G" for green, and "B" for blue), the average intensity selected channel compared direct to the threshold_value value.

Value

A matrix of 0, 1 and NA representing white, black and transparent pixels, respectively.

Author(s)

Carlos Biagolini-Jr.

See Also

threshold_color

Examples

```
# Image examples provided by bwimage package
bush<-system.file("extdata/bush.JPG",package ="bwimage")
canopy<-system.file("extdata/canopy.JPG",package ="bwimage")

# Convert images to a list of matrices
working_matrices<-threshold_image_list(c(bush,canopy), "jpeg", "proportional", compress_rate = 0.1)
```

`topline` *Top line*

Description

Line running along the crest of highest black pixel.

Usage

```
topline(imagematrix, height_size = NA, width_size = NA)
```

Arguments

`imagematrix`	The matrix to be analysed.
`height_size`	Real size of image height (in mm, cm, m, etc..).
`width_size`	Real size of image width (in mm, cm, m, etc..).

Value

Top line size that cover black pixels

Author(s)

Carlos Biagolini-Jr.

References

Zehm et al 2003 Multiparameter analysis of vertical vegetation structure based on digital image processing. Flora-Morphology, Distribution, Functional Ecology of Plants, 198: 142-160.

See Also

threshold_color

Examples

```
# First, get a matrix from your image. Here an example of a bush image is used.
bush<-system.file("extdata/bush.JPG",package ="bwimage")
bush_imagematrix<-threshold_color(bush, "jpeg", "proportional", compress_rate = 0.1)

# See the proportion of black pixels (1) below each bush image matrix row
topline(bush_imagematrix,100,100)
# Conclusion: topline size is 785.6 cm.
```

Index