Package 'Sky'

February 4, 2016

Type Package

Title Canopy Openness Analyzer Package

Version 1.0

Date 2016-02-01

Author Benedicte Bachelot

Maintainer Benedicte Bachelot <bmb43@duke.edu>

Description Provides an alternative to manually process hemispherical pictures. The algorithm processes each picture one by one to determine the proportion of sky pixel. The algorithm uses the Ridler and Calvard method (Ridler and Calvard 1978).

Depends R (>= 2.10),EBImage

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2016-02-04 11:27:45

R topics documented:

`Sky-package` *Canopy Openness Analyzer ~~ Sky ~~*

Description

Provides an alternative to manually process hemispherical pictures in order to evaluate the amount of light reaching the understory. The algorithm processes each picture one by one to determine the proportion of sky pixels (canopy openness). The algorithm uses the Ridler and Calvard method (Ridler and Calvard 1978) and was inspired by the matlab work of Jose Suau Rodriguez.

Details

Package:	Sky
Type:	Package
Version:	1.0
Date:	2016-02-01
License:	GPL (>=2)

This package should be used to process hemispherical canopy pictures in order to evaluate the amount of light reaching the understory. Pictures can be analyzed one by one, using the Ridler function, or automatically using the Ridler_auto function. At the moment, only one algorithm is offered to assess the proportion of sky pixels, using the function Ridler (Ridler and Calvard 1978). In the future, other alternative algorithmes will be incorporated. This package is ideal to quickly process many canopy pictures and get a good estimate of the amount of direct light reaching the understory. As a result, it is a useful tool for plant community ecologists, plant physiologists, and teachers that are investigating the effects of light on various processes.

Ridler function enables the user to process one picture and visualize the result (this can be useful to tune up the algorithm).

Ridler_auto function enables to automatically process all the pictures from picture folders. It also allows the user to have sub-folders such that pictures are analyzed in an organized fashion. For example, if you want to analyze pictures by years, organize the pictures within year folders (EX: 2000, 2001, 2002). Ridler_auto will then analyze the pictures year after year.

Author(s)

Benedicte Bachelot

Maintainer: Benedicte Bachelot <bmb43@duke.edu>

References

Ridler, T.W. & Calvard, S. (1978). Picture thresholding an iterative selection method. IEEE transactions on Systems, Man and Cybernetics, 8(8), 630-632.

Examples

```
data(Original_ex)
Ridler(Original_ex,p=FALSE)
```

`Original_ex` *An hemispherical picture*

Description

This dataset is an example of a typical hemispherical picture.

Usage

```
data(Original_ex)
```

Format

The format is: Formal class 'Image' [package "EBImage"] with 2 slots ..@ .Data : num [1:600, 1:450, 1:3] 0 0 0 0 0 0 0 0 0 0 ... ..@ colormode: int 2

Examples

```
data(Original_ex)
display(Original_ex)
```

Ridler *Analyzing canopy pictures*

Description

Ridler uses the Ridler and Calvard algorithm (1978) to iteratively find the threshold for each picture in order to calculate the proportion of sky (canopy openness) in one picture. The function can also display the original picture, the gray-scale picture, and the final picture (once the threshold is applied). This function was inspired by the matlab work of Jose Suau Rodriguez.

Usage

```
Ridler(Original, pixel, p = TRUE)
```

Arguments

`Original`	is the original picture to analyze
`pixel`	Indicates the number of pixel of the hemispherical picture. If not given, Ridler will use the total number of pixel detected.
`p`	if p==TRUE, three pictures are displayed: the original, the gray version, and the final version.

Value

the value corresponds to the proportion of sky (canopy openness).

Author(s)

Benedicte Bachelot

References

Ridler, T.W. & Calvard, S. (1978). Picture thresholding an iterative selection method. IEEE transactions on Systems, Man and Cybernetics, 8(8), 630-632.

Examples

```
data(Original_ex)
Ridler(Original_ex,p = TRUE)
```

`Ridler_auto` *Automatically analyzing canopy openness*

Description

Ridler_auto uses the Ridler and Calvard algorithm (1978) to iterativelly find the threshold for each picture in order to calculate the proportion of sky (canopy openness) in multiple pictures at time.

Usage

```
Ridler_auto(path1, path2 = TRUE, write = TRUE, pixel)
```

Arguments

`path1`	corresponds to the path to the picture folders.
`path2`	if TRUE it indicates that there are sub-folder within path1, for example different year folders. Then the alogrithm will look within these folders to analyze the pictures.
`write`	if TRUE, the function will write the result file as a .csv in path1.
`pixel`	Indicates the number of pixel of the hemispherical picture. If not given, Ridler_auto will use the total number of pixel detected.

Value

if path2==FALSE, the value is a dataframe with two columns. The first one, "Name", corresponds to the name of each picture. The second one, "Sky", is the proportion of sky pixels corresponding to each picture. If path2==TRUE, the value is a list containing one dataframe (similar to the one previously described) for each picture sub-folder. The names of the list correspond to the names of the sub-folders.

Author(s)

Benedicte Bachelot

References

Ridler, T.W. & Calvard, S. (1978). Picture thresholding an iterative selection method. IEEE transactions on Systems, Man and Cybernetics, 8(8), 630-632.

Examples

```
## The function is currently defined as
function (path1, path2 = TRUE, write = TRUE, pixel)
{
    R <- list()
    setwd(path1)
    ll <- list.files(path = path1)
    for (j in 1:length(ll)) {
        if (path2 == TRUE) {
            file <- ll[j]
            l <- list.files(path = file)
            eval(parse(text = paste("setwd('", path1, "/", file,
                "')", sep = "")))
        }
        if (path2 == FALSE) {
            l <- ll
        }
        Result <- matrix(data = NA, nrow = length(l), ncol = 2)
        for (k in 1:length(l)) {
            r <- Ridler(readImage(l[k]), pixel, p = FALSE)
            Result[k, 1] <- l[k]
            Result[k, 2] <- r
        }
        setwd(path1)
        Result <- data.frame(Result)
        Result[, 1] <- as.character(Result[, 1])
        Result[, 2] <- as.numeric(as.character(Result[, 2]))
        colnames(Result) <- c("Name", "Sky")
        R[[j]] <- Result
        if (write == TRUE & path2 == TRUE) {
            eval(parse(text = paste("write.csv(Result,'Result_",
                file, ".csv')", sep = "")))
            names(R)[j] <- file
        }
        if (write == FALSE & path2 == TRUE) {
            names(R)[j] <- file
        }
        if (write == TRUE & path2 == FALSE) {
            write.csv(R[[1]], "Result.csv")
        }
    }
    if (path2 == TRUE) {
        return(R)
    }
    if (path2 == FALSE) {
        return(R[[1]])
    }
  }
```

Index