Package ‘xROI’

June 2, 2021

Title Delineate Region of Interests (ROI's) and Extract Time-Series Data from Digital Repeat Photography Images

Version 0.9.20

Date 2021-06-01

Author Bijan Seyednasrollah, Thomas Milliman, Andrew D. Richardson

Maintainer Bijan Seyednasrollah <bijan.s.nasr@gmail.com>

Description Digital repeat photography and near-surface remote sensing have been used by environmental scientists to study the environmental change for nearly a decade. However, a user-friendly, reliable, and robust platform to extract color-based statistics and time-series from a large stack of images is still lacking. Here, we present an interactive open-source toolkit, called 'xROI', that facilitate the process time-series extraction and improve the quality of the final data. 'xROI' provides a responsive environment for scientists to interactively a) delineate regions of interest (ROI), b) handle field of view (FOV) shifts, and c) extract and export time series data characterizing image color (i.e. red, green and blue channel digital numbers for the defined ROI). Using 'xROI', user can detect FOV shifts without minimal difficulty. The software gives user the opportunity to readjust the mask files or redraw new ones every time an FOV shift occurs. 'xROI' helps to significantly improve data accuracy and continuity.

Depends R (>= 4.0.0)

Imports colourpicker, data.table, graphics, jpeg, lubridate, methods, moments, RCurl, raster, rgdal, rjson, sp, stats, stringr, tiff, utils, shiny, shinyjs,

Suggests knitr, testthat, rmarkdown, shinyBS, shinyAce, shinyTime, shinyFiles, shinydashboard, shinythemes, plotly

License AGPL-3

Encoding UTF-8

BugReports https://github.com/bnasr/xROI/issues

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-06-02 04:30:02 UTC

R topics documented:

`addMask` *Plot or add a mask*

Description

This function plots or adds a mask raster on the default graphics.

Usage

```
addMask(mask, add = TRUE, col = "black")
```

Arguments

`mask`	a binary or logical matrix, describing the mask (0:black for selected pixels, 1:white for not selected pixels)
`add`	a logical variable, whether to add the mask to an existing plot
`col`	a character string, color value of the plotted mask

Examples

```
#read a mask file in TIFF format
m <- tiff::readTIFF(system.file(package = 'xROI', 'dukehw-mask.tif'))
str(m)

#plot the mask in black color
addMask(m, add = FALSE)

#add the same mask in the red color to the existing plot
```

```
addMask(m, add = TRUE, col = 'red')
```

`detectShifts` *Detect FOV shift*

Description

This function calculates day-to-day similarity of images based on the CLI file.

Usage

```
detectShifts(cli_path)
```

Arguments

`cli_path`	a character string, path to the CLI file

Value

a data.frame with two columns containing day-to-day correlations of the brightness and blue bands

Examples

```
cli_path <- system.file(package = 'xROI', 'archboldbahia-cli.jpg')

cor_mat <- detectShifts(cli_path)

plot(cor_mat$brightness.cor)
plot(cor_mat$blue.cor)
```

`drawPolygon` *Interactive drawing of a polygon*

Description

This function provides an interactive tool for drawing of polygons by user clicks on the plotted graphics

Usage

```
drawPolygon(col = "#80303080", lty = 1, ...)
```

Arguments

`col`	a character string, color value of the polygon polygon
`lty`	a numeric value, lty variable as line type
`...`	passing graphical arguments

Value

the coodinates of the clicked points

Examples

```
#user can click to add vertices, pressing the Escape key would end it.
if(interactive()){
   drawPolygon()
}
```

`extractCCC` *Extract chromatic coefficients and their statistics*

Description

This function applies a mask matrix to a jpeg image and extract statstical metrics for each chromatic coordinate on R, G and B.

Usage

```
extractCCC(path, m)
```

Arguments

`path`	a character string, path to the JPEG file
`m`	a binary mask, mask binary matrix (0 for included, 1 for not)

Value

The function returns statistical metrics for each color channel. It returns NULL, if dimensions do not agree.

Examples

```
m <- tiff::readTIFF(system.file(package = 'xROI', 'dukehw-mask.tif'))
jpgFile <- system.file(package = 'xROI', 'dukehw.jpg')
cc <- extractCCC(jpgFile, m)
```

`extractCCCTimeSeries` *Extract chromatic coefficients and their statistics for an array of JPEG files*

Description

This function apples a list of mask matrices to a vector of jpeg images and extract statstical metrics for each chromatic coordinate on R, G and B.

Usage

```
extractCCCTimeSeries(rmskList, mIndex, paths)
```

Arguments

`rmskList`	a list, rasters of mask as list
`mIndex`	a numeric vector, a vector of integer numbers as the index of mask files, same length as paths. This vector shows which mask should be used with which JPEG file.
`paths`	a vector of character strings, paths to the JPEG files

Value

This function returns statistical metrics for each color channel. The function returns NULL, if dimensions do not agree.

`getCL` *Center line column of an image*

Description

This function returns the R,G,B vectors extracted from the center-line of a JPEG file

Usage

```
getCL(file)
```

Arguments

`file`	a character string, path to the JPEF file

Value

a three-column matrix of red, green blue bands of the center line

Examples

```
f <- system.file(package = 'xROI', 'dukehw.jpg')
cli <- getCL(f)
```

getCLArray *Center line arrary of an array of image*

Description

This function returns CLI array for vector of JPEG files

Usage

```
getCLArray(files)
```

Arguments

`files`	a vector of character strings, paths to the JPEF files

Value

A 3D array. The center line image as an array (NxHx3), where N is number of files, and H is the height of an image in pixels.

Examples

```
f <- system.file(package = 'xROI', 'dukehw.jpg')
cli <- getCL(f)
```

gettmpdir *Path to the TEMP directory*

Description

This function returns the path to the TEMP directory

Usage

```
gettmpdir()
```

Value

the path to the system temporary directory

Examples

```
p <- gettmpdir()
```

Launch — *Launch xROI app*

Description

This function launches the app by opening the default web browser.

Usage

```
Launch(inputDir = NULL, Interactive = FALSE)
```

Arguments

`inputDir`	a character string. path to the input directory.
`Interactive`	logical variable to force an interactive session

Value

this should be run in an interactive R session

Examples

```
#Launch xROI app
xROI::Launch()
```

parsePhenocamFilenames — *Parse Phenocam filenames*

Description

This function parse filename to extract sitename, date and timing of the images based on the phenocam naming convention.

Usage

```
parsePhenocamFilenames(filepaths)
```

Arguments

`filepaths`	a character vector of filenames

Value

a datatable containing filenames, with site name, date and timing

---

`parseROI` *Parse ROI list file*

Description

This function reads the ROI list file and returns it as a list variable

Usage

```
parseROI(roifilepath)
```

Arguments

`roifilepath`	a character string. path to the ROI file

Value

a list. ROI list file as a list.

Examples

```
f <- system.file(package = 'xROI', 'example/ROI/example_DB_0001_roi.csv')
roi <- parseROI(f)
```

`plotCLArray` *Plot CLI array*

Description

This function plots a CLI array on the graphics.

Usage

```
plotCLArray(clArray, bands = 1:3)
```

Arguments

`clArray`	a numeric array. A 3D array of CLI (HxWx3)
`bands`	an integer vector. integer vector of length 3, showing bands to be plotted

Value

invisibly returns the dimension of the plotted image

Examples

```
f <- system.file(package = 'xROI', 'dukehw-cli.jpg')
jp <- jpeg::readJPEG(f)
if(interactive())plotCLArray(jp)
```

`plotJPEG` *Plot JPEG image*

Description

This funciton plots a jpeg image as a raster given image path.

Usage

```
plotJPEG(path, add = FALSE, xlim = NULL, ylim = NULL)
```

Arguments

`path`	a character string. path to the JPEG file to be plotted.
`add`	logical. logical variable whether to add the image to the existing graphics.
`xlim`	numeric vector of lenght 2, x axis range
`ylim`	numeric vector of lenght 2, y axis range

Value

This function returns statistical metrics for each color channel. The function returns NULL, if dimensions do not agree.

Examples

```
f <- system.file(package = 'xROI', 'dukehw.jpg')
if (interactive() ) plotJPEG(f)
```

rasterizeROI *Rasterize ROI Polygons*

Description

This function convert point-based polygons to raster format

Usage

```
rasterizeROI(pnts, imgSize)
```

Arguments

pnts	a numeric matrix. a two column matrix of points as relative x and y values (0 to 1)
imgSize	a numeric vector, size of the final raster

Value

a binary matrix. matrix of the mask file.

Examples

```
pnts <- matrix(c(0.1, 0.2,
                 0.1, 0.4,
                 0.5, 0.4,
                 0.5, 0.2),
                 4, 2, byrow= TRUE)
imgSize <- c(300, 400)
m <- rasterizeROI(pnts, imgSize)
xROI::addMask(m, add = FALSE)
```

writeROI *Write ROI list file*

Description

This function writes the ROI list file on a disk space.

Usage

```
writeROI(ROIList, roifilepath)
```

Arguments

`ROIList`	a list, ROI List variable to be written
`roifilepath`	a character string, path to the ROI file

Examples

```
#loading the ROI files from the example directory
f <- system.file(package = 'xROI', 'example/ROI/example_DB_0001_roi.csv')

#parsing the example ROI file and store in roi
roi <- parseROI(f)

#write the loaded ROI in the temporary path
tempPath <- file.path(tempdir(), 'roi.csv')
writeROI(roi, tempPath)
```

Index