Package 'rasterpic'

October 14, 2022

Title Create a Spatial Raster from Plain Images

Version 0.2.1

Description Create a spatial raster, as the ones provided by 'terra', from regular pictures.

License MIT + file LICENSE

URL https://dieghernan.github.io/rasterpic/, https://github.com/dieghernan/rasterpic

BugReports https://github.com/dieghernan/rasterpic/issues

Depends R (>= 3.6.0)

Imports png (>= 0.1-5), sf (>= 1.0.0), terra (>= 1.4-22)

Suggests knitr, rmarkdown, testthat (>= 3.0.0), tidyterra, vdiffr (>= 1.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.0

NeedsCompilation no

Author Diego Hernangómez [aut, cre, cph] (<https://orcid.org/0000-0001-8457-4658>)

Maintainer Diego Hernangómez <diego.hernangomezherrero@gmail.com>

Repository CRAN

Date/Publication 2022-06-10 11:30:02 UTC

R topics documented:

asp_ratio 2
rasterpic_img 2

Index 7

asp_ratio *Compute aspect ratio of an object*

Description

Helper function. Ratio is computed as width/height (or col/rows).

Usage

```
asp_ratio(x)
```

Arguments

x	A SpatRaster object, an sf/sfc object or a numeric vector of length 4 with coordinates c(xmin, ymin, xmax, ymax), as created by sf::st_bbox()

Value

The aspect ratio

Examples

```
library(terra)

x <- rast(system.file("tiff/elev.tiff", package = "rasterpic"))
plot(x)
asp_ratio(x)
```

rasterpic_img *Convert an image to a geo-tagged raster*

Description

Geotags an image based on the coordinates of a given spatial object.

Usage

```
rasterpic_img(
  x,
  img,
  halign = 0.5,
  valign = 0.5,
  expand = 0,
  crop = FALSE,
  mask = FALSE,
```

```
  inverse = FALSE,
  crs
)
```

Arguments

x	It could be • A sf, sfc, sfg or bounding box (see sf::st_bbox()) object (**sf** package). • A SpatRaster, SpatVector or SpatExtent object (**terra** package). • A numeric vector of length 4 with the extent to be used for geotagging (i.e. c(xmin, ymin, xmax, ymax)).
img	An image to be geotagged. It can be a local file or an online file (e.g. "https://i.imgur.com/6yHmlwT.jpeg") The following image extensions are accepted: • png • jpeg/jpg • tiff/tif
halign	Horizontal alignment of img with respect to the x object. It should be a value between 0 (x is aligned on the left edge of the raster) and 1 (x is on the right edge of the raster).
valign	Vertical alignment of img with respect to the x object. It should be a value between 0 (x is aligned on the bottom edge of the raster) and 1 (x is on the top edge of the raster).
expand	An expansion factor of the bounding box of x. 0 means that no expansion is added, 1 means that the bounding box is expanded to double the original size.
crop	Logical. Should the raster be cropped to the (expanded) bounding box of x?
mask	Logical. Should the raster be masked to x? See terra::mask() for details. This option is only valid if x is a sf/sfc object.
inverse	Logical. It affects only if mask = TRUE. If TRUE, areas on the raster that do not overlap with x are masked.
crs	Character string describing a coordinate reference system. This parameter would only affect if x does not present a Coordinate Reference System (e.g. when x is a SpatExtent, sfg bbox or a vector of coordinates). See **Details**

Details

The function preserves the Coordinate Reference System of the x object. For optimal results do not use geographic coordinates (longitude/latitude).

crs can be in a WKT format, as a "authority:number" code such as "EPSG:4326", or a PROJ-string format such as "+proj=utm +zone=12". It can be also retrieved as sf::st_crs(25830)$wkt. See value and **Notes** on terra::crs().

Value

A SpatRaster object.

See Also

sf::st_crs(), sf::st_bbox(), terra::crs().

Examples

```
library(sf)
library(terra)

x_path <- system.file("gpkg/UK.gpkg", package = "rasterpic")
x <- st_read(x_path, quiet = TRUE)
img <- system.file("img/vertical.png", package = "rasterpic")

# Default config
ex1 <- rasterpic_img(x, img)

class(ex1)


plotRGB(ex1)
plot(x$geom,
  add = TRUE,
  col = NA,
  border = "white",
  lwd = 2
)

# Expand
ex2 <- rasterpic_img(x,
  img,
  expand = 0.5
)

plotRGB(ex2)
plot(x$geom,
  add = TRUE,
  col = NA,
  border = "white",
  lwd = 2
)

# Align
ex3 <- rasterpic_img(x,
  img,
  halign = 0
)

plotRGB(ex3)
plot(x$geom,
  add = TRUE,
  col = NA,
  border = "white",
```

```
  lwd = 2
)

# Crop
ex4 <- rasterpic_img(x,
  img,
  crop = TRUE
)

plotRGB(ex4)
plot(x$geom,
  add = TRUE,
  col = NA,
  border = "white",
  lwd = 2
)

# Mask
ex5 <- rasterpic_img(x,
  img,
  mask = TRUE
)

plotRGB(ex5)
plot(x$geom,
  add = TRUE,
  col = NA,
  border = "white",
  lwd = 2
)


# Mask inverse
ex6 <- rasterpic_img(x,
  img,
  mask = TRUE,
  inverse = TRUE
)

plotRGB(ex6)
plot(x$geom,
  add = TRUE,
  col = NA,
  border = "white",
  lwd = 2
)


# Combine Mask inverse and crop
ex7 <- rasterpic_img(x,
  img,
  crop = TRUE,
  mask = TRUE,
```

```
  inverse = TRUE
)

plotRGB(ex7)
plot(x$geom,
  add = TRUE,
  col = NA,
  border = "white",
  lwd = 2
)
```

Index

asp_ratio, 2

rasterpic_img, 2

sf::st_bbox(), *2–4*
sf::st_crs(), *4*

terra::crs(), *3*, *4*
terra::mask(), *3*