Package 'rasterpic'

May 21, 2026

Title Convert Digital Images into 'SpatRaster' Objects

Version 0.5.0

Description Create 'SpatRaster' objects, as defined by the 'terra' package, from digital images using a spatial object as a geographic reference. Supported inputs include objects from the 'sf', 'terra' and 'stars' packages. The main function is an S3 generic, so other packages can provide methods for additional spatial classes.

License MIT + file LICENSE

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

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

Depends R (>= 4.1.0)

Imports png, sf (>= 1.0.0), terra (>= 1.8-21)

Suggests doclisting, ggplot2, knitr, quarto, stars, testthat (>= 3.0.0), tidyterra

VignetteBuilder quarto

Config/Needs/check curl

Config/Needs/coverage curl

Config/Needs/website cpp11, curl, devtools, dieghernan/gitdevr, mapsf, maptiles, remotes, tmap

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

X-schema.org-keywords cran, jpeg, jpg, maps, png, r, r-package, r-stats, raster, rstats, sf, stars, terra, tif, tiff, cran-r

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2026-05-21 11:30:02 UTC

Contents

`rasterpic_img` *Convert an image into a geotagged* `SpatRaster`

Description

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

`rasterpic_img()` is an S3 generic. **rasterpic** provides methods for the following classes:

- `SpatExtent`
- `SpatRaster`
- `SpatVector`
- `bbox`
- `default`
- `numeric`
- `sf`
- `sfc`
- `sfg`
- `stars`

Usage

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

```
## S3 method for class 'sf'
rasterpic_img(
  x,
  img,
  halign = 0.5,
  valign = 0.5,
  expand = 0,
  crop = FALSE,
  ...,
  mask = FALSE,
  inverse = FALSE
)

## S3 method for class 'sfc'
rasterpic_img(
  x,
  img,
  halign = 0.5,
  valign = 0.5,
  expand = 0,
  crop = FALSE,
  ...,
  mask = FALSE,
  inverse = FALSE
)

## S3 method for class 'sfg'
rasterpic_img(
  x,
  img,
  halign = 0.5,
  valign = 0.5,
  expand = 0,
  crop = FALSE,
  ...,
  mask = FALSE,
  inverse = FALSE,
  crs = NULL
)

## S3 method for class 'stars'
rasterpic_img(
  x,
  img,
  halign = 0.5,
  valign = 0.5,
  expand = 0,
  crop = FALSE,
```

```
  ...
)

## S3 method for class 'bbox'
rasterpic_img(
  x,
  img,
  halign = 0.5,
  valign = 0.5,
  expand = 0,
  crop = FALSE,
  ...,
  crs = NULL
)

## S3 method for class 'numeric'
rasterpic_img(
  x,
  img,
  halign = 0.5,
  valign = 0.5,
  expand = 0,
  crop = FALSE,
  ...,
  crs = NULL
)

## S3 method for class 'SpatRaster'
rasterpic_img(
  x,
  img,
  halign = 0.5,
  valign = 0.5,
  expand = 0,
  crop = FALSE,
  ...
)

## S3 method for class 'SpatVector'
rasterpic_img(
  x,
  img,
  halign = 0.5,
  valign = 0.5,
  expand = 0,
  crop = FALSE,
  ...,
  mask = FALSE,
```

```
  inverse = FALSE
)

## S3 method for class 'SpatExtent'
rasterpic_img(
  x,
  img,
  halign = 0.5,
  valign = 0.5,
  expand = 0,
  crop = FALSE,
  ...,
  crs = NULL
)
```

Arguments

x	An **R** object (see **S3 methods**).
img	An image to be geotagged. It can be a local file or a URL (e.g. "https://i.imgur.com/6yHmlwT.jpeg") Accepted extensions are png, jpeg/jpg and tiff/tif.
halign	A number between 0 and 1 giving the horizontal alignment of img relative to x. 0 aligns x with the left edge, 1 aligns with the right edge and 0.5 centers it horizontally.
valign	A number between 0 and 1 giving the vertical alignment of img relative to x. 0 aligns x with the bottom edge, 1 aligns with the top edge and 0.5 centers it vertically.
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. See **Details**.
crop	Logical. Should the raster be cropped to the (expanded) bounding box of x? See **Details**.
...	Further arguments passed to methods.
mask	Logical, available for vector methods. Should the raster be masked to the shape of x? See **Details**.
inverse	Logical. This only has an effect when mask = TRUE. If TRUE, areas of the raster covered by x are masked.
crs	Character string describing a CRS. This parameter only applies when x is a SpatExtent, sfg, bbox or a numeric coordinate vector. See **CRS** section.

Details

vignette("rasterpic", package = "rasterpic") explains, with examples, the effect of parameters halign, valign, expand, crop and mask.

S3 methods:

rasterpic supports the following input classes:

- **sf** classes: `sf`, `sfc`, `sfg` or `bbox`.
- **terra** classes: `SpatRaster`, `SpatVector` and `SpatExtent`.
- **stars** classes: `stars`.
- A numeric coordinate vector of the form `c(xmin, ymin, xmax, ymax)`.

Other packages can provide methods for additional spatial classes.

Methods for extent-like inputs use the object extent. Methods for vector inputs can also mask the image to the object shape.

CRS:

This function preserves the CRS of `x` when applicable. 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 also be retrieved with:

- `sf::st_crs(25830)$wkt`.
- `terra::crs()`.
- `tidyterra::pull_crs()`.

See **Value** and **Notes** in `terra::crs()`.

Value

A `SpatRaster` object (see `terra::rast()`) where each layer corresponds to a color channel of `img`:

- If `img` has at least 3 layers, the result will have an additional property setting layers 1 to 3 as the red, green and blue channels with names `"r"`, `"g"` and `"b"` and `alpha` if applicable.
- If `img` already has a definition of RGB values (this may be the case for `tiff`/`tif` files) the result will keep that channel definition.

The resulting `SpatRaster` will have an RGB specification as explained in `terra::RGB()`.

See Also

`vignette("rasterpic", package = "rasterpic")` for examples.

From **sf**:

- `sf::st_crs()`.
- `sf::st_bbox()`.
- `vignette("sf1", package = "sf")` to understand how **sf** organizes **R** objects.

From **stars**:

- `stars::st_as_stars()`.

From **terra**:

- `terra::vect()`, `terra::rast()` and `terra::ext()`.
- `terra::mask()`.
- `terra::crs()`.

- `terra::RGB()`.

For plotting:

- `terra::plot()` and `terra::plotRGB()`.
- `tidyterra::autoplot.SpatRaster()` and `tidyterra::geom_spatraster_rgb()` with **ggplot2**.
- **tmap**, **mapsf** and **maptiles**.

Examples

```
library(sf)
library(terra)
library(ggplot2)
library(tidyterra)

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 configuration.
ex1 <- rasterpic_img(x, img)

ex1

autoplot(ex1) +
  geom_sf(data = x, fill = NA, color = "white", linewidth = 0.5)

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

autoplot(ex2) +
  geom_sf(data = x, fill = NA, color = "white", linewidth = 0.5)

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

autoplot(ex3) +
  geom_sf(data = x, fill = NA, color = "white", linewidth = 0.5)
labs(title = "Align")

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

autoplot(ex4) +
  geom_sf(data = x, fill = NA, color = "white", linewidth = 0.5) +
  labs(title = "Crop")

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

autoplot(ex5) +
```

```
  geom_sf(data = x, fill = NA, color = "white", linewidth = 0.5) +
  labs(title = "Mask")

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

autoplot(ex6) +
  geom_sf(data = x, fill = NA, color = "white", linewidth = 0.5) +
  labs(title = "Mask Inverse")

# Combine inverse masking and cropping.
ex7 <- rasterpic_img(x, img, crop = TRUE, mask = TRUE, inverse = TRUE)

autoplot(ex7) +
  geom_sf(data = x, fill = NA, color = "white", linewidth = 0.5) +
  labs(title = "Combine")
```

Index