

Package ‘rasterpic’

August 24, 2026

Title Convert Images to Spatially Referenced 'SpatRaster' Objects

Version 1.0.0

Description Convert digital images to spatially referenced 'SpatRaster' objects, as defined by the 'terra' package, using coordinates from supported spatial input classes. Supported inputs include numeric coordinate vectors and objects from the 'sf', 'terra' and 'stars' packages. The main function is an S3 generic, allowing other packages to extend support to 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 cli, png, sf (>= 1.0.0), terra (>= 1.8-21), tools, utils

Suggests ggplot2, knitr, quarto, stars, testthat (>= 3.3.0),
tidyterra, withr

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 8.1.0

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 2026-08-24 16:30:03 UTC

Contents

rasterpic_img 2
Index 9

rasterpic_img	<i>Geotag an image as a SpatRaster</i>
---------------	--

Description

Geotag an image and return a SpatRaster using coordinates from a supported spatial input class.
rasterpic_img() is an S3 generic. See **S3 methods** for supported input classes.

Usage

```

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

## Default S3 method:
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 for supported classes.
img	An image to geotag. It can be a local file or a URL, for example "https://i.imgur.com/6yHmlwT.jpeg". Accepted file extensions are png, jpg, jpeg, tif and tiff.
halign	A number between 0 and 1 giving the horizontal alignment of img relative to x. 0 aligns img with the left edge of x, 1 aligns it 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 img with the bottom edge of x, 1 aligns it 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 for vector methods. Should the raster be masked to the shape of x? See Details .
inverse	Logical. Only used when mask = TRUE. If TRUE, areas covered by x are masked instead of areas outside x.
crs	Character string describing a CRS. This parameter only applies when x is a SpatExtent, sfg, bbox or a numeric coordinate vector. See the CRS section.

Details

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

S3 methods:

All methods share the parameters, return value and examples documented on this page. **rasterpic** groups them by input behavior:

- Vector methods for the **sf** classes sf, sfc and sfg and the **terra** class SpatVector can mask the image to the input shape.

- Extent-like methods for bbox, numeric coordinate vectors, the **stars** class stars and the **terra** classes SpatRaster and SpatExtent use the input bounding box.
- The default method reports unsupported classes. Other packages can provide methods for additional spatial classes.

CRS:

This function preserves the CRS of x when applicable. For optimal results, **do not use** geographic coordinates (longitude/latitude).

crs can be in WKT format, as an "authority:number" code such as "EPSG:4326" or as a PROJ-string such as "+proj=utm +zone=12". It can also be retrieved with:

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

See the **Value** and **Notes** sections 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 names layers 1 to 3 "r", "g" and "b" for the red, green and blue channels and names layer 4 "alpha" when applicable.
- If img already has an RGB specification (this may be the case for tif/tiff files), the result keeps that specification.

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**.

Other image geotagging tools: `asp_ratio()`

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")

# Use the default configuration.
ex1 <- rasterpic_img(x, img)

ex1

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

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

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

# Align the image to the left edge.
ex3 <- rasterpic_img(x, img, halign = 0)

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

# Crop to the bounding box.
ex4 <- rasterpic_img(x, img, crop = TRUE)

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

# Mask to the vector shape.
ex5 <- rasterpic_img(x, img, mask = TRUE)

autoplot(ex5) +
  geom_sf(data = x, fill = NA, color = "white", linewidth = 0.5) +
  labs(title = "Mask")
```

```
# Mask outside the vector shape.
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 cropping and inverse masking.
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

* **geotagging**

 rasterpic_img, [2](#)

asp_ratio(), [7](#)

masked, [5](#)

rasterpic_img, [2](#)

sf::st_bbox(), [6](#)

sf::st_crs(), [6](#)

sf::st_crs(25830)\$wkt, [6](#)

stars::st_as_stars(), [6](#)

terra::crs(), [6](#)

terra::ext(), [6](#)

terra::mask(), [6](#)

terra::plot(), [7](#)

terra::plotRGB(), [7](#)

terra::rast(), [6](#)

terra::RGB(), [6](#)

terra::vect(), [6](#)

tidyterra::autoplot.SpatRaster(), [7](#)

tidyterra::geom_spatraster_rgb(), [7](#)

tidyterra::pull_crs(), [6](#)