Package 'crstools'

March 19, 2026

Title Tools to Work with Projections

Version 1.0.0

Description Choose the appropriate map projection for a given application, visualise the resulting distortion, and georeference data from unknown projections. The full functionalities of the package are described in Pozzi et al. (2026) <doi:10.31223/X5K17P> (pre-print).

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

Imports sf, ggplot2, terra, cli, jpeg, rlang

Suggests testthat (>= 3.0.0), knitr, rmarkdown, pastclim, rnaturalearth, rnaturalearthdata, tidyterra, svglite, spelling

Config/testthat/edition 3

VignetteBuilder knitr

Language en-GB

NeedsCompilation no

Author Andrea Manica [aut, cre] (ORCID: <https://orcid.org/0000-0003-1895-450X>), Andrea Pozzi [aut], Cecilia Padilla-Iglesias [aut], Evie Carter [aut], Michela Leonardi [aut], Aramish Fatima [aut], Ondrej Pelanek [aut], Nile Stephenson [aut], Margherita Colucci [aut], Dominik C Jud [aut], Cassandra Gunasekaram [aut]

Maintainer Andrea Manica <am315@cam.ac.uk>

Repository CRAN

Date/Publication 2026-03-19 13:30:02 UTC

Contents

choose_gcp 2
extract_coords 3
find_gcp_coords 4
geom_tissot 5
georeference_img 6
suggest_crs 7

Index **10**

`choose_gcp` *Function to choose the Ground Control Point (GCP) on an image*

Description

GCPs are used to georeference images by providing known coordinates for specific points in the image.

Usage

```
choose_gcp(image_obj, gcp = NULL, col = "red")
```

Arguments

`image_obj`	An array representing the image (colour images are generally imported as an array of nx x ny x 3 colour channels), or a file path to the image (currenly this can only be of type .jpg).
`gcp`	(optional) A dataframe of GCPs, containing columns `id`, `x`, `y`,`longitude`, `latitude`. This is used if we want to add additional GCP to an existing list (usually created by running this function multiple times).
`col`	The colour of the points to be plotted on the image. Default is "red".

Value

A dataframe with the GCPs, including the image coordinates and their corresponding geographic coordinates.

Examples

```
# Get the path to an example image included in the package and choose GCPs
img_path <- system.file("extdata/europe_map.jpeg", package = "crstools")
# this will open a new window where you can choose some points
gcp_europe <- choose_gcp(img_path)
# after the first set of points is chosen, we can add more points
gcp_europe <- choose_gcp(img_path, gcp = gcp_europe)
```

extract_coords *Extract coordinates*

Description

Extract coordinates from a georeferenced image.

Usage

```
extract_coords(georef_image, coords_df = NULL, col = "red")
```

Arguments

georef_image	A spatraster object representing the georeferenced image.
coords_df	An optional dataframe containing the coordinates of points previously extracted from the image. Default is NULL.
col	The colour of the points to be plotted on the image. Default is "red".

Value

A dataframe with ID and coordinates of the points extracted from the image.

Examples

```
# Load required packages
library(terra)
# Georeference the image using the created GCPs
# get the gcp coordinates
gcp_europe_coords <-
readRDS(system.file("vignettes/img/europe_gcp_georef_v2.RDS",
package = "crstools"))
# get the path to the image
img_path <- system.file("extdata/europe_map.jpeg", package = "crstools")
georef_path <-
georeference_img(image_obj = img_path,
gcp = gcp_europe_coords,
output_path = file.path(tempdir(), "europe_map_georef"))
# georeference the image using the GCPs
map_warp <- rast(georef_path)
# get the coordinates of the points
coords_df <- extract_coords(map_warp)
# if needed, extract additional points by supplying the coords_df argument
# and re run the function
coords_df <- extract_coords(map_warp, coords_df)
# dataframe with ID and coordinates of the points extracted from the image.
print(coords_df)
```

find_gcp_coords *Find the coordinates (longitude and latitude) of the ground control points (GCPs) in a given image.*

Description

Find the coordinates (longitude and latitude) of the ground control points (GCPs) in a given image.

Usage

```
find_gcp_coords(gcp, sf_obj)
```

Arguments

gcp	A data frame containing the GCPs with columns id, x, y, longitude, and latitude.
sf_obj	the reference map, as an sf object (already cut to the extend and if needed projected)

Value

A data frame with the GCPs, including their image coordinates and corresponding geographic coordinates.

Examples

```
# load required packages
library(sf)
library(rnaturalearth)
# get the path to an example image included in the package and choose GCPs
img_path <- system.file("extdata/europe_map.jpeg",
package = "crstools")
# choose some points
gcp_europe <- choose_gcp(img_path)
# now get some more
gcp_europe <- choose_gcp(img_path, gcp = gcp_europe)
# create a map of europe to use to get the coordinates
world <- ne_countries(scale = "medium", returnclass = "sf")
# transform it to a suitable projection
world <- st_transform(world, crs = 4326)
# crop it to the extent of the image
europe <- st_crop(world, c(xmin = -25, ymin=25, xmax = 45, ymax = 70))
# get the coordinates for these points
new_gcp_europe <- find_gcp_coords(gcp_europe, sf_obj = europe)
# data frame with the GCPs
print(new_gcp_europe)
```

geom_tissot *Add Tissot's indicatrix to a map*

Description

This function adds Tissot's indicatrix to a map. Tissot's indicatrix is a mathematical contrivance used in cartography to characterize local distortions due to map projection.

Usage

```
geom_tissot(
  mapping = ggplot2::aes(),
  data = NULL,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  centres = c(5, 5),
  radius = NULL,
  fill = "red",
  alpha = 0.7,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by ggplot2::aes(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are two options: A sf object or a SpatRaster object
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. ggplot2::geom_point().
centres	Either a list with elements "lng" and "lat" giving the longitudes and latitudes of the grid of centres for the Tissot's indicatrix, or a vector of length 2 with the number of rows and columns to generate such a grid automatically. The latter is passed to pretty() to get pretty breaks, and the exact number of Tissot's circles might differ from the input numbers (see pretty() for details. It defaults to c(5,5).

radius	The radius of the circles (see sf::st_buffer() for an explanation of how units are created; we recommend that you explicitly state your using with the units::as_units(), e.g units::as_units(100, "km")
fill	The fill colour of the circles
alpha	The transparency of the circles
...	Other arguments passed on to ggplot2::geom_sf()

Value

A ggplot2 object

Examples

```
# load required packages
library(rnaturalearth)
library(sf)
library(ggplot2)
s_america_sf <- ne_countries(continent = "South America", returnclass = "sf")
s_am_equal_area <- suggest_crs(s_america_sf, distortion = "equal_area")
ggplot(data = s_america_sf) +
 geom_sf() +
 geom_tissot() +
 coord_sf(crs = s_am_equal_area$proj4) +
 theme_minimal()
```

---

georeference_img *Georeference an Image using Ground Control Points (GCPs)*

Description

This function warps a given image using a set of Ground Control Points (GCPs) to create a georeferenced version of the image. The function uses GDAL tools to first translate the image into a georeferenced TIFF format and then applies a warp operation to reproject the image into a spatial reference system (EPSG:4326). The warped image is saved as a new file with the suffix _warp.tif.

Usage

```
georeference_img(image_obj, gcp, output_path = NULL)
```

Arguments

image_obj	A character string specifying the file path to the input image (JPEG format). The function reads this image and applies the GCPs for georeferencing.
gcp	A data frame containing the Ground Control Points (GCPs). This dataframe can be produced with the *draw_gcp_points* function. This data frame should have the following columns:

- `id`: An identifier for each GCP (numeric).
- `x`: The x-coordinate of the GCP (in pixel space).
- `y`: The y-coordinate of the GCP (in pixel space).
- `lon`: The longitude of the GCP (georeferenced).
- `lat`: The latitude of the GCP (georeferenced).

`output_path` A character string representing the file path to the input image. (`_warp.tif`) will be appended to it.

Value

A character string representing the path to the newly created warped TIFF image file (`_warp.tif`). This file contains the georeferenced image.

Examples

```
# get the path to an example image included in the package
img_path <- system.file("extdata/europe_map.jpeg", package = "crstools")
# load a set of GCPs (or we could create them using the choose_gcp()
# and find_gcp() functions)
gcp_df <- readRDS(system.file(
  "extdata/europe_gcp_georef.RDS",
  package = "crstools"
))
#' # Assuming you have a set of GCPs in gcp_df and an image file "image.jpg"
warped_img <- georeference_img(
  image_obj = img_path, gcp = gcp_df,
  output_path = tempfile(
    patter = "georef_img_",
    tmpdir = tempdir(),
    fileext = ".tif"
  )
)
```

`suggest_crs` *Suggestions for map projections depending on the extent*

Description

This function provides suggestions for map projections depending on the extent of the map. For SDMs, an equal area projection is generally favoured, but it is also possible to get suggestions for conformal, equidistant or compromise projections. The algorithm is a reimplementation in R of Projection Wizard (Bojan et al., 2016), version 2.1. (https://projectionwizard.org/)

Usage

```
suggest_crs(
  x,
  distortion = c("equal_area", "conformal", "equidistant", "compromise"),
  round_cm = FALSE,
  return_best = TRUE,
  datum = c("WGS84", "ETRS89", "NAD83"),
  unit = c("m", "ft"),
  lat_check = TRUE,
  world_equidist = NULL,
  quiet = FALSE
)
```

Arguments

x	A vector of four numeric values representing the extent of the map (xmin, xmax, ymin, ymax), or a SpatExtent object, or a SpatRaster object.
distortion	The type of distortion to be minimized. Options are "equal_area", "conformal", "equidistant" and "compromise". Default is "equal_area".
round_cm	Logical. If TRUE, the central meridian is rounded to the nearest degree. Default is FALSE.
return_best	Logical. If TRUE, only the best projection is returned, otherwise, if there are multiple options, a list will be returned
datum	The datum to use. Options are "WGS84", "ETRS89" and "NAD83". Default is "WGS84".
unit	The unit to use. Options are "m" and "ft". Default is "m".
lat_check	Logical. If TRUE, the function will check if lat values are within range (-90,90). Default is TRUE.
world_equidist	if distortion="equidistant" for a whole world projection, then this parameter should be a list with one of the following sets of elements: • "Polar azimuthal equidistant": prj = "polar", pole, lng_central``, where poleis either -90 or = "polar", pole = 90, lng_central = -180)‘ • "Oblique azimuthal equidistant": prj = "oblique", lat_centre, lng_centre, where lat_centre and lng_centre are the latitude and longitude of the centre from or through which distances are correct. E.g. list(prj = "oblique", lat_centre = 39, lng_centre = 145) • "Two-point azimuthal": prj = "two_points", lat1, lng1, lat2, lng2, where lat1, lng1, lat2, lng2 are the latitude and longitude of two points on the map from which distances are correct. E.g. list(prj = "two_points", lat1 = 34, lng1 = -117, lat2 = 46, lng2 = 16)
quiet	Logical. If TRUE, suppresses messages. Default is FALSE.

Value

Either a list of two strings (proj4 and WKT) for a single projection (if either only one projection is available or return_best is TRUE), or a list of lists, each containing two strings (proj4 and WKT) for a single projection (if multiple projections are available and return_best is FALSE).

References

Bojan, S, Bernhard, J, & Helen, J (2016) Projection Wizard - An Online Map Projection Selection Tool. The Cartographic Journal 53(2):177-185 DOI: 10.1080/00087041.2015.1131938

Examples

```
# Whole map
suggest_crs(c(-180, 180, -90, 90))
# Northen Hemisphere
suggest_crs(c(-180, 180, 21, 70))
# Hemisphere showing the tropics
suggest_crs(c(-180, 180, -7, 21))
# Regional map for EW extent
suggest_crs(c(-60, 20, 40, 70))
```

Index

choose_gcp, 2

extract_coords, 3

find_gcp_coords, 4

geom_tissot, 5
georeference_img, 6
ggplot2::aes(), *5*
ggplot2::geom_point(), *5*
ggplot2::geom_sf(), *6*

pretty(), *5*

sf::st_buffer(), *6*
suggest_crs, 7