Package: CityShadeMapper (via r-universe)

August 22, 2024

Title Generate High Resolution Shade Maps and Shaded Routes from Remote Sensing Data

Version 0.0.1

Description CityShadeMapper can generate high resolution shade maps (e.g. for every square meter and every hour of the year) for any city or town from open remote sensing (LiDAR) data. CityShadeMapper can also return optimal routes within any two points in the city that maximise the amount of shade for pedestrians.

License AGPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

LazyData true

RoxygenNote 7.3.1

Imports class, gdistance, leaflet, leaflet.extras, lidR, lubridate, lutz, magrittr, nominatimlite, raster, RColorBrewer, solartime, rayshader, terra

Suggests covr, testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://pakillo.github.io/CityShadeMapper/

BugReports https://github.com/Pakillo/CityShadeMapper/issues

Repository https://pakillo.r-universe.dev

RemoteUrl https://github.com/Pakillo/CityShadeMapper

RemoteRef HEAD

RemoteSha 2b84fdcc1ef08a8546eb95eb68c2516ed023e4dd

Contents

`calc_heights_from_lidar`

Generate height raster from LiDAR data

Description

Generate height raster from LiDAR data

Usage

```
calc_heights_from_lidar(
  las = NULL,
  res = 1,
  ground = FALSE,
  thresholds = c(0, 2, 5, 10, 20),
  filename = NULL,
  ...
)
```

Arguments

`las`	LiDAR data (a `lidR::LAS-class()` or `lidR::LAScatalog-class()` object).
`res`	Spatial resolution of the raster
`ground`	Logical. Calculate ground height in addition to canopy/roof heights?
`thresholds`	Set of height thresholds (see `lidR::dsm_pitfree()`).
`filename`	Character. Output filename. Note that if a file already exists with that name, it will be overwritten.
`...`	further arguments to `lidR::pitfree()`

Value

SpatRaster with height data

Examples

```
## Not run:
heights <- calc_heights_from_lidar(PlazaNueva())

## End(Not run)
```

`calc_shaded_route` *Calculate shaded route*

Description

Calculate shaded route

Usage

```
calc_shaded_route(origin = NULL, destination = NULL, shaderas = NULL)
```

Arguments

`origin`	A character vector describing an address, an `sf` or a `SpatVector` object.
`destination`	A character vector describing an address, an `sf` or a `SpatVector` object.
`shaderas`	Shade raster

Value

A SpatVector object with the optimal route

Examples

```
## Not run:
shaderas <- terra::rast("/vsicurl/https://zenodo.org/record/7213637/files/m7_h13_ground.tif")
shade.route <- calc_shaded_route("Plaza Nueva, Sevilla", "Mateos Gago, Sevilla", shaderas)

library(leaflet)
leaflet(sf::st_as_sf(shade.route)) |>
  leaflet::addWMSTiles(baseUrl = "https://www.ign.es/wms-inspire/ign-base",
  layers = "IGNBaseTodo-nofondo") |>
  leaflet::addTiles(urlTemplate =
    "https://mapasdesombra.github.io/Sevilla-jul-ground/13/{z}/{x}/{y}.png",
  options = leaflet::tileOptions(minZoom = 15, maxZoom = 18, tms = TRUE, opacity = 0.4)) |>
  addPolylines(weight = 8, opacity = 0.8)


## End(Not run)
```

example_building *Single building example*

Description

An example 9 x 9 meter SpatRaster with a 4 m tall building in the middle.

Usage

```
example_building()
```

Value

A SpatRaster.

example_square *Square example*

Description

An example 100 x 100 meter SpatRaster with a 10 m tall rectangular wall and a 4-m tall tree in the middle.

Usage

```
example_square()
```

Value

A SpatRaster.

example_tree *Single tree example*

Description

An example 9 x 9 meter SpatRaster with a 4 m tall tree in the middle.

Usage

```
example_tree()
```

Value

A SpatRaster.

`get_sun_position` *Get sun position*

Description

Get sun position (declination, elevation, and azimuth) for a given day and time.

Usage

```
get_sun_position(
  lon = NULL,
  lat = NULL,
  date = NULL,
  hour = NULL,
  omit.nights = TRUE
)
```

Arguments

`lon`	Longitude (numeric value between -180 and 180)
`lat`	Latitude (numeric value between -90 and 90)
`date`	A Date object or character giving the date in YYYY-MM-DD format (e.g. "2021-02-19"). Can be a vector too, e.g. c("2021-02-19", "2021-08-04"). Note different years have no effect on sun position calculations.
`hour`	Hour of the day. Integer number (or numeric vector) between 0 and 23 (both included).
`omit.nights`	Logical. If TRUE, sun positions will only be returned when it is daytime (i.e. nighttimes will be omitted)

Value

A data frame with solar elevation and azimuth per hour as returned by `solartime::computeSunPositionDoyHour()` but converted to degrees rather than radians.

Examples

```
sunpos <- get_sun_position(lon = -5.99, lat = 37.39, date = "2021-02-19", hour = 15)
sunpos <- get_sun_position(lon = -5.99, lat = 37.39,
date = c("2021-02-19", "2022-08-05"), hour = 10:14)
```

`leaflet_shademap` *Make leaflet shade map*

Description

Make leaflet shade map

Usage

```
leaflet_shademap(
  url.canopy = NULL,
  url.ground = NULL,
  url.cog = NULL,
  band = 1,
  satellite = FALSE,
  opacity = 0.5
)
```

Arguments

`url.canopy`	Character. Url pointing to shade tiles at the canopy level.
`url.ground`	Character. Url pointing to shade tiles at the ground level.
`url.cog`	Character. Url pointing to a cloud-optimised geotiff (starting with 'https://'). The COG raster must have EPSG:4326 projection. Currently not implemented.
`band`	Numeric. Band to show in multilayer rasters (only for COG).
`satellite`	Logical. Add satellite images as another layer to the map?
`opacity`	Numeric between 0 and 1. Opacity of the shade layer.

Value

A leaflet map

Examples

```
## Not run:
leaflet_shademap(url.canopy = "https://mapasdesombra.github.io/Sevilla-jul-canopy/11/{z}/{x}/{y}.png")

## End(Not run)
```

make_shademap *Calculate shade map*

Description

Calculate shade raster for given dates and hours

Usage

```
make_shademap(
  height.ras = NULL,
  date = NULL,
  hour = NULL,
  type = c("canopy", "ground"),
  cover.ras = NULL,
  zscale = 1,
  omit.nights = TRUE,
  filename = NULL,
  ...
)
```

Arguments

height.ras	A terra::SpatRaster() with heights data. Note height.ras must have a well defined crs (see terra::crs()).
date	A Date object or character giving the date in YYYY-MM-DD format (e.g. "2021-02-19"). Can be a vector too, e.g. c("2021-02-19", "2021-08-04"). Note different years have no effect on sun position calculations.
hour	Hour of the day. Integer number (or numeric vector) between 0 and 23 (both included).
type	Character. Either 'canopy' to get illumination at the canopy/roof level, or 'ground' to get illumination at the ground level (i.e. for pedestrians).
cover.ras	A SpatRaster containing cover classes (ground, vegetation, buildings...). See rasterize_lidar_cover_class().
zscale	Default 1. The ratio between the x and y spacing (which are assumed to be equal) and the z axis. For example, if the elevation is in units of meters and the grid values are separated by 10 meters, zscale would be 10.
omit.nights	Logical. If TRUE, sun positions will only be returned when it is daytime (i.e. nighttimes will be omitted)
filename	Character. Output filename. Note that if a file already exists with that name, it will be overwritten.
...	further arguments to rayshader::ray_shade()

Value

A (possibly multilayer) SpatRaster object with the intensity of illumination at each pixel for every date and time

Examples

```
## Not run:
lidar <- PlazaNueva()
heights <- calc_heights_from_lidar(lidar)
shaderas <- make_shademap(heights, date = "2022-10-15", hour = 13)
plot_shademap(shaderas, smooth = TRUE)
shaderas <- make_shademap(heights, date = "2022-10-15", hour = 8:20)
plot_shademap(shaderas, animate = TRUE, smooth = TRUE)
shaderas <- make_shademap(heights, date = "2022-07-15", hour = 8:21)
plot_shademap(shaderas, animate = TRUE, smooth = TRUE)
shaderas <- make_shademap(heights, date = c("2022-07-15", "2022-10-15"), hour = 13)
plot_shademap(shaderas, legend = FALSE)

## Ground-level shade maps require additional raster with cover classes
lidar <- read_lidar(system.file("extdata", "PlazaNueva.laz", package = "CityShadeMapper"))
cover.ras <- rasterize_lidar_cover_class(lidar)
shaderas <- make_shademap(heights, date = "2022-10-15", hour = 13,
  type = "ground", cover.ras = cover.ras)


## End(Not run)
```

---

make_shademap_ground *Calculate shade map at the ground level*

Description

Calculate shade map at the ground level

Usage

```
make_shademap_ground(shademap.canopy = NULL, cover.ras = NULL, filename = NULL)
```

Arguments

shademap.canopy	A SpatRaster with the illumination at the canopy level, made with make_shademap().
cover.ras	A SpatRaster containing cover classes (ground, vegetation, buildings...). See rasterize_lidar_cover_class().
filename	Character. Output filename. Note that if a file already exists with that name, it will be overwritten.

Value

A (possibly multilayer) SpatRaster object with the intensity of illumination at the ground level

Examples

```
## Not run:
lidar <- read_lidar(system.file("extdata", "PlazaNueva.laz", package = "CityShadeMapper"))
cover.ras <- rasterize_lidar_cover_class(lidar)
shaderas.canopy <- make_shademap(heights, date = "2022-10-15", hour = 13)
shaderas.ground <- make_shademap_ground(shaderas.canopy, cover.ras)

## Alternatively, call make_shademap directly:
heights <- calc_heights_from_lidar(lidar)
shaderas.ground <- make_shademap(heights, date = "2022-10-15", hour = 13,
  type = "ground", cover.ras = cover.ras)

## End(Not run)
```

PlazaNueva *Plaza Nueva LiDAR data*

Description

LiDAR data of Plaza Nueva in Sevilla, Spain. Data provided by LiDAR-PNOA 2018 CC-BY 4.0 scne.es.

Usage

```
PlazaNueva()
```

Value

A LAScatalog.

---

plot_shademap *Plot shade map*

Description

Plot shade map

Usage

```
plot_shademap(shade.ras = NULL, legend = TRUE, animate = FALSE, ...)
```

Arguments

shade.ras	A terra::SpatRaster() object with shade intensity, as produced by make_shademap().
legend	Logical. Show legend?
animate	Logical. Show animation of all the shade.ras layers?
...	Further arguments to terra::plot(), or to terra::animate() if animate is TRUE.

Value

A static or animated plot.

---

rasterize_lidar_cover_class

Get raster of cover classification from lidar points

Description

Get raster of cover classification from lidar points

Usage

```
rasterize_lidar_cover_class(
  las = NULL,
  res = 1,
  fill.holes = TRUE,
  filename = NULL
)
```

Arguments

las	A lidR::LAScatalog-class() object, or a character vector with paths to LAS/LAZ objects.
res	Resolution of the resulting raster.
fill.holes	Logical. Try to fill holes in lidar point classification.
filename	Character. Output filename. Note that if a file already exists with that name, it will be overwritten.

Value

A SpatRaster with the classification of cover types: 2 = ground (including low vegetation < 1m) 4 = high vegetation (> 1m) 6 = buildings 9 = water and NA values. Note that points classified as bridges (class 17) will be reclassified as ground.

Examples

```
## Not run:
pza <- system.file("extdata", "PlazaNueva.laz", package = "CityShadeMapper")
pza.cover <- rasterize_lidar_cover_class(pza)

## End(Not run)
```

`read_lidar` *Read LiDAR data*

Description

Read LiDAR data

Usage

```
read_lidar(folder = NULL, select = "xyz", filter = "-drop_class 7", ...)
```

Arguments

`folder`	Character. Path to folder containing las/laz files. Can also be a vector of file paths.
`select`	Character. Point attributes to read from the las/laz files. Use `select = "*"` to read all attributes. Default is "xyz" to save memory. See `lidR::readLAS()` for more details.
`filter`	Character. Optional. Use if you want to filter out some data points. For example, we could filter out noisy data points (class = 7) using `filter = "-drop_class 7"`. Use `filter = ""` to read all data points.
`...`	Further arguments for `lidR::readLAScatalog()`.

Value

LAScatalog object with LiDAR data

Examples

```
## Not run:
las <- system.file("extdata", "PlazaNueva.laz", package = "CityShadeMapper")
lidar <- read_lidar(las)

## End(Not run)
```

Index