Package 'vostokR'

May 8, 2026

Type Package

Title Solar Potential Calculation for Point Clouds using 'VOSTOK'

Version 0.2.1

Date 2026-03-25

Description Calculate solar potential for LiDAR point clouds using the 'VOSTOK' (Voxel Octree Solar Toolkit) algorithm. This R program provides an interface to the original 'VOSTOK' C++ implementation by Bechtold and Hofle (2020), enabling efficient ray casting and solar position algorithms to compute solar irradiance for each point while accounting for shadowing effects. Integrates seamlessly with the 'lidR' package for LiDAR data processing workflows. The original 'VOSTOK' toolkit is available at <doi:10.11588/data/QNA02B>.

License GPL (>= 3)

URL https://github.com/bi0m3trics/vostokR

BugReports https://github.com/bi0m3trics/vostokR/issues

Encoding UTF-8

Imports Rcpp (>= 1.0.11), lidR (>= 4.0.0), data.table, terra, sf (>= 1.0.0), methods

Suggests testthat (>= 3.0.0)

LinkingTo Rcpp, RcppArmadillo, RcppEigen

Config/testthat/edition 3

RoxygenNote 7.3.3

SystemRequirements C++17, OpenMP (optional)

NeedsCompilation yes

Author Andrew J. Sanchez Meador [aut, cre], Sebastian Bechtold [aut] (Original VOSTOK C++ implementation), Bernhard Hofle [aut] (Original VOSTOK C++ implementation)

Maintainer Andrew J. Sanchez Meador <andrew.sanchezmeador@nau.edu>

Repository CRAN

Date/Publication 2026-03-25 22:30:02 UTC

Contents

`add_normals` *Add Normal Vectors to Point Cloud*

Description

This function adds normal vectors to a LiDAR point cloud using a highly optimized C++/Eigen-based eigenvalue decomposition method with OpenMP parallelization. This implementation is 5-10x faster than pure R approaches and comparable to lasR, while providing additional geometric features useful for forestry applications.

Usage

```
add_normals(
  las,
  k = 10,
  add_features = FALSE,
  num_threads = 0,
  verbose = FALSE
)
```

Arguments

`las`	LAS object from lidR package
`k`	Number of neighbors to use for normal estimation (default: 10)
`add_features`	Logical, whether to add eigenvalue-based geometric features (default: FALSE)
`num_threads`	Number of OpenMP threads to use (default: 0 = auto-detect)
`verbose`	Logical. Print informational messages (default: FALSE)

Value

LAS object with added normal vectors (nx, ny, nz) and optionally geometric features

See Also

`calculate_solar_potential` for solar potential calculation

Examples

```
library(lidR)
library(vostokR)

# Load test data
LASfile <- system.file("extdata", "test.laz", package="vostokR")
las <- readLAS(LASfile)

# Add normals using fast C++ method
las <- add_normals(las, k = 10)
names(las@data)
```

`calculate_solar_potential` *Calculate Solar Potential for LiDAR Point Cloud*

Description

This function takes a LiDAR point cloud from the lidR package and calculates the solar potential for each point, taking into account shadowing effects from surrounding points. Location information (lat/lon/timezone) is auto-detected from the CRS when possible.

Usage

```
calculate_solar_potential(las, ...)

## S3 method for class 'LAS'
calculate_solar_potential(
  las,
  year = 2025,
  day_start = NULL,
  day_end = NULL,
  start_date = NULL,
  end_date = NULL,
  day_step = 30,
  minute_step = 30,
  min_sun_angle = 5,
  voxel_size = 1,
  lat = NULL,
  lon = NULL,
  timezone = NULL,
  n_threads = NULL,
  clear_cache = FALSE,
  verbose = FALSE,
  ...
)
```

```
## S3 method for class 'LAScatalog'
calculate_solar_potential(
  las,
  year = 2025,
  day_start = 1,
  day_end = 365,
  day_step = 30,
  minute_step = 30,
  min_sun_angle = 5,
  voxel_size = 1,
  lat,
  lon,
  timezone,
  ...
)
```

Arguments

las	LAS or LAScatalog object from lidR package containing point cloud data
...	Additional arguments passed to other methods
year	Numeric. Year for solar calculation (default: 2025)
day_start	Numeric. Start day of year (1-365). Ignored if start_date is provided.
day_end	Numeric. End day of year (1-365). Ignored if end_date is provided.
start_date	Date or character. Start date (e.g., "2025-06-01"). Overrides day_start.
end_date	Date or character. End date (e.g., "2025-08-31"). Overrides day_end.
day_step	Numeric. Step size in days for calculation (default: 30)
minute_step	Numeric. Time step in minutes for calculation within each day (default: 30)
min_sun_angle	Numeric. Minimum sun angle in degrees for calculation (default: 5)
voxel_size	Numeric. Size of voxels for octree shadow calculation (default: 1)
lat	Numeric. Latitude of the study area (auto-detected from CRS if NULL)
lon	Numeric. Longitude of the study area (auto-detected from CRS if NULL)
timezone	Numeric. Time zone offset from UTC (auto-detected from CRS if NULL)
n_threads	Numeric. Number of OpenMP threads to use (default: auto-detect)
clear_cache	Logical. Clear performance caches before calculation (default: FALSE)
verbose	Logical. Print informational messages (default: FALSE)

Value

LAS object with added column for solar potential in Wh/m^2/day

Examples

```
library(lidR)
library(vostokR)

# Load test data
LASfile <- system.file("extdata", "test.laz", package="vostokR")
las <- readLAS(LASfile)
las <- add_normals(las, k = 10)

# Quick single-day calculation
las_solar <- calculate_solar_potential(las,
                                       year = 2025,
                                       day_start = 172,
                                       day_end = 172,
                                       day_step = 1,
                                       minute_step = 60,
                                       lat = 35.0,
                                       lon = -111.0,
                                       timezone = -7)
```

---

clear_vostokr_caches *Clear VostokR Performance Caches*

Description

Clears internal SOLPOS and shadow caches to free memory

Usage

```
clear_vostokr_caches(verbose = FALSE)
```

Arguments

verbose	Logical. Print informational messages (default: FALSE)

Value

No return value, called for side effects (clears internal caches).

get_vostokr_performance_info *Get VostokR Performance Information*

Description

Returns a named list containing information about OpenMP availability and thread configuration for VostokR.

Usage

```
get_vostokr_performance_info()
```

Value

A named list with the following elements:

openmp_enabled Logical. Whether OpenMP support is available.

max_threads Integer. Maximum number of threads available.

current_threads Integer. Number of threads currently in use.

get_vostokr_threads *Get Current VostokR Thread Count*

Description

Returns the number of OpenMP threads currently configured for VostokR parallel computations.

Usage

```
get_vostokr_threads()
```

Value

An integer scalar indicating the current number of OpenMP threads used by VostokR.

plot_solar_potential *Plot Solar Potential Point Cloud*

Description

Plots solar potential values directly on the point cloud using lidR's native plotting

Usage

```
plot_solar_potential(las, ...)
```

Arguments

las	LAS object with solar potential values
...	Additional arguments passed to lidR::plot()

Value

No return value, called for side effects (produces a plot).

Examples

```
library(lidR)
library(vostokR)
LASfile <- system.file("extdata", "test.laz", package = "vostokR")
las <- readLAS(LASfile)
las <- add_normals(las, k = 10)
las_solar <- calculate_solar_potential(las,
  year = 2025, day_start = 172, day_end = 172,
  day_step = 1, minute_step = 60,
  lat = 35.0, lon = -111.0, timezone = -7)
plot_solar_potential(las_solar)
```

set_vostokr_threads *Set VostokR OpenMP Thread Count*

Description

Control the number of OpenMP threads used by VostokR calculations

Usage

```
set_vostokr_threads(n_threads = NULL, verbose = FALSE)
```

Arguments

n_threads	Integer. Number of threads to use. If NULL, auto-detect based on available cores and lidR settings.
verbose	Logical. Print informational messages (default: FALSE)

Value

No return value, called for side effects (sets thread count).

---

solar_ground_raster *Convert Solar Potential Ground Points to Raster*

Description

Extracts ground points from solar potential results and converts to terra SpatRaster

Usage

```
solar_ground_raster(
  las,
  res = 1,
  ground_class = 2,
  use_all_points = FALSE,
  verbose = FALSE
)
```

Arguments

las	LAS object with solar potential values
res	Numeric. Resolution of output raster (default: 1)
ground_class	Numeric. Classification code for ground points (default: 2)
use_all_points	Logical. If TRUE and no ground points found, use all points (default: FALSE)
verbose	Logical. Print informational messages (default: FALSE)

Value

A `SpatRaster` object (from the **terra** package) containing mean solar potential values (Wh/m^2/day) for ground points, gridded at the specified resolution. Returns `NULL` if no ground points are found and `use_all_points` is `FALSE`.

Examples

```
library(lidR)
library(vostokR)
LASfile <- system.file("extdata", "test.laz", package = "vostokR")
las <- readLAS(LASfile)
las <- add_normals(las, k = 10)
las_solar <- calculate_solar_potential(las,
  year = 2025, day_start = 172, day_end = 172,
  day_step = 1, minute_step = 60,
  lat = 35.0, lon = -111.0, timezone = -7)
solar_raster <- solar_ground_raster(las_solar, res = 0.5)
```

Index