

Package ‘AMAPVox’

July 9, 2021

Type Package

Title LiDAR Data Voxelisation

Version 0.1.0

Description Read, manipulate and write voxel spaces. Voxel spaces are read from text-based output files of the 'AMAPVox' software. 'AMAPVox' is a LiDAR point cloud voxelisation software that aims at estimating leaf area through several theoretical/numerical approaches. See more in the article Vincent et al. (2017) <[doi:10.23708/1AJNMP](https://doi.org/10.23708/1AJNMP)> and the technical note Vincent et al. (2021) <[doi:10.23708/1AJNMP](https://doi.org/10.23708/1AJNMP)>.

License CeCILL (>= 2)

Encoding UTF-8

URL <https://github.com/umr-amap/AMAPVox>

BugReports <https://github.com/umr-amap/AMAPVox/issues>

RoxygenNote 7.1.1

Imports methods, utils, data.table, stringr

Collate 'AMAPVox.R' 'AMAPVoxClasses.R' 'AMAPVoxGenerics.R' 'AMAPVoxGetters.R' 'ReadVoxelSpace.R' 'WriteVoxelSpace.R'

NeedsCompilation no

Author Grégoire Vincent [aut],
Julien Heurtebize [aut],
Philippe Verley [aut, cre]

Maintainer Philippe Verley <philippe.verley@ird.fr>

Repository CRAN

Date/Publication 2021-07-09 08:10:07 UTC

R topics documented:

AMAPVox	2
getMaxCorner	3
getMinCorner	4

getParameter	4
getPosition	5
getResolution	6
is.VoxelSpace	7
length.VoxelSpace	7
readVoxelSpace	8
show,VoxelSpace-method	8
VoxelSpace-class	9
VoxHeader-class	9
writeVoxelSpace	10

Index	11
--------------	-----------

AMAPVox

AMAPVox package

Description

The package provides a set of R functions for reading, manipulating and writing voxel spaces. Voxel spaces are read from text-based output files of the <https://amap-dev.cirad.fr/projects/amapvox> AMAPVox software.

Details

As of version 0.1, available functions are limited and rudimentary, basically read/write voxel space. With time the package will include functions to launch full GUI tool from R and a set of useful pre/post-processing functions.

References

- Research paper first describing AMAPVox:
Vincent, G., Antin, C., Laurans, M., Heurtebize, J., Durrieu, S., Lavalley, C., & Dauzat, J. (2017). Mapping plant area index of tropical evergreen forest by airborne laser scanning. A cross-validation study using LAI2200 optical sensor. *Remote Sensing of Environment*, 198, 254-266. doi: [10.1016/j.rse.2017.05.034](https://doi.org/10.1016/j.rse.2017.05.034)
- Up-to-date description of PAD/LAD estimators implemented in AMAPVox:
VINCENT, Gregoire; PIMONT, François; VERLEY, Philippe, 2021, "A note on PAD/LAD estimators implemented in AMAPVox 1.7", doi: [10.23708/1AJNMP](https://doi.org/10.23708/1AJNMP), DataSuds, V1

Contact

<contact@amapvox.org>

Author(s)

Philippe VERLEY <philippe.verley@ird.fr>

See Also

Useful links:

- <https://github.com/umr-amap/AMAPVox>
- Report bugs at <https://github.com/umr-amap/AMAPVox/issues>

getMaxCorner	<i>Gets the x, y, z coordinates of the voxel space top right corner.</i>
--------------	--

Description

Gets the x, y, z coordinates of the voxel space top right corner.

Usage

```
getMaxCorner(voxelSpace)  
  
## S4 method for signature 'VoxelSpace'  
getMaxCorner(voxelSpace)
```

Arguments

voxelSpace the [VoxelSpace-class](#) object.

Value

the x, y, z coordinates of the voxel space top right corner, as a numerical vector.

Examples

```
# load a voxel file  
vox <- readVoxelSpace(system.file("extdata", "als_sample.vox", package = "AMAPVox"))  
# retrieve 'max_corner' parameter  
getMaxCorner(vox)
```

getMinCorner	<i>Gets the x, y, z coordinates of the voxel space bottom left corner.</i>
--------------	--

Description

Gets the x, y, z coordinates of the voxel space bottom left corner.

Usage

```
getMinCorner(voxelSpace)

## S4 method for signature 'VoxelSpace'
getMinCorner(voxelSpace)
```

Arguments

voxelSpace the [VoxelSpace-class](#) object.

Value

the x, y, z coordinates of the voxel space bottom left corner, as a numerical vector.

Examples

```
# load a voxel file
vox <- readVoxelSpace(system.file("extdata", "als_sample.vox", package = "AMAPVox"))
# retrieve 'min_corner' parameter
getMinCorner(vox)
```

getParameter	<i>Gets a parameter from the VoxelSpace header.</i>
--------------	---

Description

Gets a parameter from the VoxelSpace header.

Usage

```
getParameter(object, what)

## S4 method for signature 'VoxHeader,character'
getParameter(object, what)

## S4 method for signature 'VoxelSpace,character'
getParameter(object, what)
```

Arguments

object either the [VoxelSpace-class](#) object or the associated [VoxHeader-class](#)
 what the name of the parameter

Value

the parameter as a character

See Also

[VoxHeader-class](#), [VoxelSpace-class](#);

Examples

```
# load a voxel file
vox <- readVoxelSpace(system.file("extdata", "als_sample.vox", package = "AMAPVox"))
# retrieve 'min_corner' parameter
getParameter(vox, "min_corner")
```

getPosition

Gets the x, y, z coordinates of a given voxel.

Description

Gets the x, y, z coordinates of the voxel center. If the voxel parameter is missing, it returns the positions of all the voxels in the voxel space.

Usage

```
getPosition(voxelSpace, voxel)

## S4 method for signature 'VoxelSpace,vector'
getPosition(voxelSpace, voxel)

## S4 method for signature 'VoxelSpace,list'
getPosition(voxelSpace, voxel)

## S4 method for signature 'VoxelSpace,missing'
getPosition(voxelSpace, voxel)
```

Arguments

voxelSpace the [VoxelSpace-class](#) object.
 voxel either the voxel index as a c(i,j,k) vector or a voxel from the VoxelSpace data.table.

Value

the x, y, z coordinates of the voxel center.

Examples

```
# load a voxel file
vox <- readVoxelSpace(
  system.file("extdata", "als_sample.vox", package = "AMAPVox"))

# get position of voxel(i=0, j=0, k=0)
getPosition(vox, c(0, 0, 0))

# get position of 1st voxel in the data.table
getPosition(vox, vox@voxels[1,])

# get positions of every voxel
getPosition(vox)
```

getResolution	<i>Gets the elemental size of a voxel (dx, dy, dz) in meter.</i>
---------------	--

Description

Gets the elemental size of a voxel (dx, dy, dz) in meter.

Usage

```
getResolution(voxelSpace)

## S4 method for signature 'VoxelSpace'
getResolution(voxelSpace)
```

Arguments

voxelSpace the [VoxelSpace-class](#) object.

Value

the size of the voxel in meter, as a numerical vector.

Examples

```
# load a voxel file
vox <- readVoxelSpace(system.file("extdata", "als_sample.vox", package = "AMAPVox"))
# retrieve 'resolution' parameter
getResolution(vox)
```

is.VoxelSpace *Tests for objects of class VoxelSpace*

Description

Tests for objects of class VoxelSpace

Usage

```
is.VoxelSpace(x)
```

Arguments

x to be tested

See Also

[VoxelSpace-class](#);

length.VoxelSpace *Length of a VoxelSpace*

Description

Get the number of voxels in the voxel space.

Retrieve the dimension of a [VoxelSpace-class](#)

Usage

```
## S3 method for class 'VoxelSpace'  
length(x)
```

```
## S3 method for class 'VoxelSpace'  
dim(x)
```

Arguments

x a [VoxelSpace-class](#) object.

Value

the number of voxels in the voxel space.

the number of voxels nx, ny, nz of a [VoxelSpace-class](#) along x, y, z axis.

Warning

AMAPVox allows to discard empty voxels in the voxel file. In such case `length.VoxelSpace` will return the expected number of voxels as if none were missing. As a consequence the number of voxels stored in the [VoxelSpace-class](#) object may be inferior to the returned value, namely `nrow(x@voxels) <= length(x)`

<code>readVoxelSpace</code>	<i>Read a voxel file</i>
-----------------------------	--------------------------

Description

read a voxel file and cast it into a [VoxelSpace-class](#) object.

Usage

```
readVoxelSpace(f)
```

Arguments

`f` The path of the voxel file.

See Also

[writeVoxelSpace](#)

Examples

```
# load a voxel file
vox <- readVoxelSpace(system.file("extdata", "als_sample.vox", package = "AMAPVox"))
```

<code>show, VoxelSpace-method</code>	<i>Show VoxelSpace-class object.</i>
--------------------------------------	--

Description

Display the [VoxelSpace-class](#) object main characteristics.

Usage

```
## S4 method for signature 'VoxelSpace'
show(object)
```

Arguments

`object` a [VoxelSpace-class](#) object.

Value

show returns an invisible NULL.

Examples

```
# load a voxel file
vox <- readVoxelSpace(
  system.file("extdata", "als_sample.vox", package = "AMAPVox"))
# show VoxelSpace object
show(vox)
```

VoxelSpace-class *VoxelSpace*

Description

Class that holds the state variables of every voxel of the voxel space in a [data.table](#), plus metadata from the voxel space header.

Value

An object of class VoxelSpace.

Fields

`file` the path of the voxel file (.vox).
`header` the [VoxHeader-class](#) object associated to this voxel file.
`voxels` the voxels hold in a data.table.

See Also

[readVoxelSpace](#)

VoxHeader-class *VoxHeader*

Description

Class containing metadata that are read in the header of the VoxelSpace file.

Value

An object of class VoxHeader

Fields

mincorner the (x, y, z) coordinates of the bottom left corner of the voxel space
maxcorner the (x, y, z) coordinates of the top right corner of the voxel space
split (nx, ny, nz) the number of voxels along x, y & z axis
resolution (x, y, z) the elemental voxel size
columnNames the names of the output variables in the voxel space. Column names of the data.table in the [VoxelSpace-class](#) object.

writeVoxelSpace	<i>Write a voxel file</i>
-----------------	---------------------------

Description

write a voxel file out of a [VoxelSpace-class](#) object.

Usage

```
writeVoxelSpace(voxelSpace, outputFile)
```

Arguments

voxelSpace the object of class [VoxelSpace](#) to write
outputFile The path where to write the voxel file.

See Also

[readVoxelSpace](#)

Examples

```

## Not run:
# load a voxel file
vox <- readVoxelSpace(system.file("extdata", "als_sample.vox", package = "AMAPVox"))
# set max PAD to 5
vox@voxels[, PadBVTotal:=max(PadBVTotal, 5, na.rm = TRUE)]
# write updated voxel file
writeVoxelSpace(vox, tempfile("pattern="amapvox_", fileext=".vox"))

## End(Not run)

```

Index

AMAPVox, [2](#)
AMAPVox-package (AMAPVox), [2](#)

data.table, [9](#)
dim.VoxelSpace (length.VoxelSpace), [7](#)

getMaxCorner, [3](#)
getMaxCorner, VoxelSpace-method
 (getMaxCorner), [3](#)
getMinCorner, [4](#)
getMinCorner, VoxelSpace-method
 (getMinCorner), [4](#)
getParameter, [4](#)
getParameter, VoxelSpace, character-method
 (getParameter), [4](#)
getParameter, VoxHeader, character-method
 (getParameter), [4](#)
getPosition, [5](#)
getPosition, VoxelSpace, list-method
 (getPosition), [5](#)
getPosition, VoxelSpace, missing-method
 (getPosition), [5](#)
getPosition, VoxelSpace, vector-method
 (getPosition), [5](#)
getResolution, [6](#)
getResolution, VoxelSpace-method
 (getResolution), [6](#)

is.VoxelSpace, [7](#)

length.VoxelSpace, [7](#)

readVoxelSpace, [8](#), [9](#), [10](#)

show, VoxelSpace-method, [8](#)

VoxelSpace-class, [8](#), [9](#)
VoxHeader-class, [9](#)

writeVoxelSpace, [8](#), [10](#)