

Package ‘dracor’

January 19, 2023

Type Package

Title Decode Draco Format 3D Mesh Data

Version 0.2.5

Date 2023-01-11

Description Decodes meshes and point cloud data encoded by the Draco mesh compression library from Google. Note that this is only designed for basic decoding and not intended as a full scale wrapping of the Draco library.

License Apache License (>= 2.0)

Imports Rcpp (>= 1.0.5)

LinkingTo Rcpp

Enhances rgl

Suggests testthat, spelling, covr

URL <https://github.com/natverse/dracor>,
<https://github.com/google/draco>

BugReports <https://github.com/natverse/dracor/issues>

RoxygenNote 7.2.3

Encoding UTF-8

Language en-GB

NeedsCompilation yes

Author Gregory Jefferis [aut, cre] (<<https://orcid.org/0000-0002-0587-9355>>),
Google Inc [aut, cph] (for the Draco library)

Maintainer Gregory Jefferis <jefferis@gmail.com>

Repository CRAN

Date/Publication 2023-01-19 12:10:02 UTC

R topics documented:

draco_decode	2
Index	4

draco_decode	<i>Decode Draco encoded raw bytes containing mesh or point cloud data</i>
--------------	---

Description

Decode Draco encoded raw bytes containing mesh or point cloud data

Usage

```
draco_decode(data, mesh3d = TRUE, ...)
```

Arguments

data	raw bytes containing Draco data e.g. as read by readBin OR a character vector containing a URL or a path to a file on disk.
mesh3d	Whether to return <code>rgl::mesh3d</code> object (when TRUE, the default) or something as close as possible to what is provided by the Draco library (when FALSE).
...	Additional arguments passed to download.file when data is a URL (e.g. <code>quiet=TRUE</code> or <code>method</code>)

Details

Note that the Draco library returns 0-based indices for the faces whereas R in general and `rgl::mesh3d` in particular expect 1-based indices. When `mesh3d=FALSE`, the result will have 0-based indices as returned by the Draco library.

If data is an http/https URL it will be downloaded to a temporary location on disk (using [download.file](#)). If data is a character vector that does not look like a URL then it is assumed to refer to a file on disk (which will be read with [readBin](#)).

Value

a `rgl::mesh3d` object or a list containing elements `points` and (for meshes). `faces`.

Examples

```
# fetch test data
# originally downloaded from:
carurl='https://github.com/google/draco/blob/master/testdata/car.drc?raw=true'
## Not run:
car.m=draco_decode(carurl)

## End(Not run)
# use cached version in package for example
car.m=draco_decode(system.file('draco/car.drc', package = 'dracor'))
str(car.m)

## show the result
```

```
if(requireNamespace("rgl", quietly=TRUE)) {  
  rgl::shade3d(car.m, col='red')  
  
  ## demonstrate conversion of raw form to rgl::mesh3d object  
  car.raw=draco_decode(carurl, mesh3d=FALSE)  
  str(car.raw)  
  car.m2 = rgl::tmesh3d(  
    vertices = car.raw$points,  
    indices = car.raw$faces + 1,  
    homogeneous = FALSE)  
}
```

Index

`download.file`, [2](#)

`draco_decode`, [2](#)

`raw`, [2](#)

`readBin`, [2](#)