Package 'readobj'

July 23, 2025

Type Package

Title Fast Reader for 'Wavefront' OBJ 3D Scene Files

Version 0.4.1

Description Wraps 'tiny_obj_loader' C++ library for reading the 'Wavefront' OBJ 3D file format including both mesh objects and materials files. The resultant R objects are either structured to match the 'tiny_obj_loader' internal data representation or in a form directly compatible with the 'rgl' package.

License BSD_2_clause + file LICENSE

Imports Rcpp (>= 0.11.6), grDevices

LinkingTo Rcpp

Suggests testthat, rgl, spelling

URL https://github.com/jefferis/readobj

BugReports https://github.com/jefferis/readobj/issues

RoxygenNote 7.1.1

Encoding UTF-8

Language en-GB

NeedsCompilation yes

Author Gregory Jefferis [aut, cph, cre] (ORCID:
<https://orcid.org/0000-0002-0587-9355>),
Syoyo Fujita [aut, cph] (tiny_obj_loader.* are copyright Syoyo Fujita),
Trevor L Davis [aut]

Maintainer Gregory Jefferis <jefferis@gmail.com>

Repository CRAN

Date/Publication 2021-07-03 07:10:02 UTC

Contents

`readobj-package` *Wrapper for tiny_obj_loader single file C++ library*

Description

This package provides fast reading of Wavefront OBJ files with support for some material properties using the tinyobjloader C++ library. It is noticeably faster than the pure R `readOBJ` implemented in the `rgl` package.

Details

Note that the rgl package does provide a `writeOBJ` function, whereas this library only focusses on fast reading of OBJ files.

As of readobj v0.4 released in June 2021, tinyobjloader was updated tag v1.0.7; this was after considering and rejecting the 2.0 series where between 2.0 rc3 and rc4 a default diffuse value was added. This update means that the internal structure of more complex meshes has changed.

See Also

`read.obj`, `readOBJ`

`read.obj` *Read a Wavefront OBJ 3D scene file into an R list*

Description

Read a Wavefront OBJ 3D scene file into an R list

Usage

```
read.obj(f, materialspath = NULL, convert.rgl = FALSE, triangulate = TRUE)
```

Arguments

`f`	Path to an OBJ file
`materialspath`	Path to a folder containing materials files. This is optional and only required if materials files are in a different folder from the OBJ file defined by `f`.
`convert.rgl`	Whether to convert the returned list to a `rgl::shapelist3d` object containing `rgl::mesh3d` objects.
`triangulate`	(default `TRUE`) Whether to convert all mesh faces to triangles. Note that only meshes with triangular or quad faces are supported, so setting `triangulate=FALSE` will throw an error for more complex files.

Details

tinyobjloader made some substantial changes to its data structures after the first code snapshot was taken for the this package in 2015. In order to benefit from bug fixes, we updated the code in 2020 but we note that tinyobjloader now de-duplicates vertices more aggressively e.g. in the situation where there are normals or texture coordinates. We were forced when converting to rgl::shapelist3d objects to revert these de-duplications on the R side in order for display in rgl; note that this only happens when there are texture coordinates and/or normals in the obj file.

Note that some fields in the tinyobjloader return structure will be omitted when they are not relevant for a given obj file. In this case, as with any R list, the list element will have the value NULL when tested. See examples.

Value

When convert.rgl=FALSE, the default, a named list with items shapes and materials, each containing sublists with one entry per object (shapes) or material (materials). Objects in the shapes list have the following structure

- positions 3 x N matrix of 3D vertices
- indices 3/4 x M matrix of indices into vertex array (trimesh/quadmesh) 0-indexed
- normals 3 x N matrix of normal directions for each vertex (missing when there are no normals)
- normindices 3/4 x M matrix of indices into normals array (trimesh/quadmesh) 0-indexed (missing when there are no normals)
- texcoords 2 x N matrix of texture coordinates (missing when there are no texture coordinates)
- texindices 3/4 x M matrix of indices into texcoords array (trimesh/quadmesh) 0-indexed (missing when there are no texture coordinates)
- nvfaces Raw vector specifying the number of vertices per face (missing unless triangulate=FALSE and there are a mixture of different numbers of vertices per face.)
- material_ids 0-indexed, -1 when not set (missing when no materials)

When convert.rgl=TRUE a list of class shapelist3d containing a mesh3d for each object or group element in the original OBJ file. See tinyobj2shapelist3d for details of rgl conversion.

Sample files

Note that at the request of the CRAN maintainers the sample files have the file extension .wavefront instead of the standard .obj because this triggers a false positive R CMD check NOTE.

See Also

tinyobj2shapelist3d, rgl::readOBJ for simpler, pure R implementation.

Examples

```
cube=read.obj(system.file("obj/cube.wavefront", package = "readobj"))
str(cube)
# elements will be NULL when not present in the obj file e.g. normals
```

```
is.null(cube$shapes[[1]]$texcoords)

# demonstrate direct conversion of result to rgl format
if(require('rgl')) {
  cuber=read.obj(system.file("obj/cube.wavefront", package = "readobj"),
    convert.rgl=TRUE)
  shade3d(cuber)
}
```

`tinyobj2shapelist3d` *Convert the raw tinyobjloader shapes/materials list into an rgl shapelist3d*

Description

Convert the raw tinyobjloader shapes/materials list into an rgl shapelist3d

Usage

```
tinyobj2shapelist3d(x)
```

Arguments

x	A raw tinyobjloader shapes/materials list

Details

Not all materials settings can be processed at the moment. In particular only the following are used:

- `diffuse` -> mapped onto rgl material `color` field
- `ambient`
- `specular`
- `emission`

Value

a list of class `shapelist3d` containing a `mesh3d` for each object or group element in the original OBJ file.

See Also

`read.obj`, `mesh3d`, `shapelist3d`, `rgl.material`

Examples

```
cube=read.obj(system.file("obj/cube.wavefront", package = "readobj"))
if(require("rgl")){
  cubesl=tinyobj2shapelist3d(cube)
  shade3d(cubesl)
}
```

Index