

Package ‘AlphaHull3D’

November 25, 2022

Type Package

Title Alpha Hull Computation

Version 2.0.0

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Description Computation of the alpha hull of a set of points in the 3D space, that is to say ``something like the shape formed by these points''. The alpha hull depends on a positive parameter alpha. When alpha goes to zero, the alpha hull degenerates to the set of points, while it is the convex hull of the set of points when alpha goes to infinity. Computations are performed by the 'CGAL' 'C++' library <<https://www.cgal.org/>>.

License GPL-3

Imports Rcpp (>= 1.0.9), rgl, Rvcg

Suggests uniformly

LinkingTo BH, Rcpp, RcppCGAL, RcppEigen

Encoding UTF-8

RoxygenNote 7.2.1

URL <https://github.com/stla/AlphaHull3D>

BugReports <https://github.com/stla/AlphaHull3D/issues>

SystemRequirements C++ 17, gmp, mpfr

Depends R (>= 2.10)

LazyData true

NeedsCompilation yes

Author Stéphane Laurent [aut, cre]

Repository CRAN

Date/Publication 2022-11-25 09:00:02 UTC

R topics documented:

| | |
|-------------|---|
| ahull3d | 2 |
| fullAhull3d | 3 |
| gissid | 3 |
| optimal | 4 |
| setAlpha | 5 |

| | |
|--------------|----------|
| Index | 6 |
|--------------|----------|

| | |
|---------|--|
| ahull3d | <i>3D alpha hull for a given alpha</i> |
|---------|--|

Description

Computes the alpha hull of a set of points for a given alpha.

Usage

```
ahull3d(points, alpha, volume = FALSE)
```

Arguments

| | |
|--------|---|
| points | the points given as a matrix with three columns |
| alpha | positive number |
| volume | Boolean, whether to return the volume of the alpha hull, but this is not always reliable |

Value

A mesh3d object, with an attribute "volume" if volume = TRUE.

Note

If you want to compute the alpha hull for several values of alpha, then instead of using this function you can use the [fullAhull3d](#) and the [setAlpha](#) functions.

Examples

```
library(AlphaHull3D)
library(uniformly)
library(rgl)

set.seed(666L)
pts <- runif_in_torus(5000L, R = 3, r = 1)
ahull <- ahull3d(pts, alpha = 2)

open3d(windowRect = c(50, 50, 562, 562))
points3d(pts)
shade3d(ahull, color = "orange", alpha = 0.4)
```

| | |
|-------------|------------------------------------|
| fullAhull3d | <i>3D alpha hull for all alpha</i> |
|-------------|------------------------------------|

Description

Computes the complete alpha hull of a set of points. The result is an external pointer to be used in the function [setAlpha](#).

Usage

```
fullAhull3d(points)
```

Arguments

points the points given as a matrix with three columns

Value

An external pointer for usage in the function [setAlpha](#).

Examples

```
library(AlphaHull3D)
ahull <- fullAhull3d(gissid)
mesh <- setAlpha(ahull, alpha = "solid")
mesh$normals <- NULL
library(rgl)
open3d(windowRect = c(50, 50, 562, 562))
shade3d(mesh, color = "purple")
```

| | |
|--------|-------------------------------------|
| gissid | <i>Great stellated dodecahedron</i> |
|--------|-------------------------------------|

Description

The vertices of the great stellated dodecahedron (Bowers acronym: gissid).

Usage

```
gissid
```

Format

A numeric matrix with 32 rows.

| | |
|---------|--|
| optimal | <i>Integer for the computation of an optimal alpha</i> |
|---------|--|

Description

This function simply returns the given integer with a special class; it is intended to be used in the alpha argument of the [setAlpha](#) function to compute and set the optimal value of alpha for which the alpha hull has no more than n connected components.

Usage

```
optimal(n)
```

Arguments

n a non-negative integer, the maximal number of connected components of the alpha hull

Value

The value is the input integer n with a special class.

Examples

```
library(AlphaHull3D)
library(uniformly)
# sample some points in a torus and some points in a sphere:
set.seed(666L)
pts_torus <- runif_in_torus(1000L, R = 3, r = 1)
pts_sphere <- runif_in_sphere(1000L, d = 3L, r = 1)
# shift the points in the sphere:
pts_sphere <- sweep(pts_sphere, 2L, c(0, 0, 2), FUN = "+")
# full alpha hull:
ahull <- fullAhull3d(rbind(pts_torus, pts_sphere))
# set optimal alpha for 2 connected components:
mesh <- setAlpha(ahull, alpha = optimal(2))
# plot:
library(rgl)
open3d(windowRect = c(50, 50, 562, 562))
shade3d(mesh, color = "yellow")
```

| | |
|----------|--|
| setAlpha | <i>Set alpha value to a full alpha hull and computes this alpha hull</i> |
|----------|--|

Description

Given a full alpha hull, this function allows to set the value of alpha, either an arbitrary positive number, or an optimal value for a desired number of connected components, or a value for which the alpha hull is solid.

Usage

```
setAlpha(ahull, alpha = "solid")
```

Arguments

| | |
|-------|--|
| ahull | a full alpha hull, i.e. an output of fullAhull3d |
| alpha | there are three possibilities for this parameter: it can be a positive number, or the character string "solid" to get and set the smallest alpha for which the alpha hull is solid, or a positive integer obtained with the optimal function to get and set the optimal alpha for which the alpha hull has no more than a desired number of connected components |

Value

A **rgl** mesh (class mesh3d) with the value of alpha in the "alpha" attribute.

See Also

[fullAhull3d](#), [optimal](#).

Examples

```
library(AlphaHull3D)
ahull <- fullAhull3d(gissid)
mesh <- setAlpha(ahull, alpha = optimal(1))
mesh$normals <- NULL
library(rgl)
open3d(windowRect = c(50, 50, 562, 562))
shade3d(mesh, color = "maroon")
```

Index

* **datasets**

gissid, 3

ahull3d, 2

fullAhull3d, 2, 3, 5

gissid, 3

optimal, 4, 5

setAlpha, 2–4, 5