Package ‘alphashape3d’

December 9, 2020

Version 1.3.1
Date 2020-12-09
Title Implementation of the 3D Alpha-Shape for the Reconstruction of
3D Sets from a Point Cloud
Author Thomas Lafarge, Beatriz Pateiro-Lopez
Maintainer Beatriz Pateiro-Lopez <beatriz.pateiro@usc.es>
Depends geometry, rgl
Suggests alphahull
Description Implementation in R of the alpha-shape of a finite set of points in the three-dimensional space. The alpha-shape generalizes the convex hull and allows to recover the shape of non-convex and even non-connected sets in 3D, given a random sample of points taken into it. Besides the computation of the alpha-shape, this package provides users with functions to compute the volume of the alpha-shape, identify the connected components and facilitate the three-dimensional graphical visualization of the estimated set.
License GPL-2
NeedsCompilation yes
Repository CRAN
Date/Publication 2020-12-09 14:00:03 UTC

R topics documented:

ashape3d .......................................................... 2
components_ashape3d ........................................ 4
inashape3d ...................................................... 5
plot.ashape3d .................................................. 6
rtorus ............................................................ 7
surfaceNormals ................................................ 8
volume_ashape3d .............................................. 9

Index 11
ashape3d  

3D \( \alpha \)-shape computation

Description

This function calculates the 3D \( \alpha \)-shape of a given sample of points in the three-dimensional space for \( \alpha > 0 \).

Usage

ashape3d(x, alpha, pert = FALSE, eps = 1e-09)

Arguments

- x: A 3-column matrix with the coordinates of the input points. Alternatively, an object of class "ashape3d" can be provided, see Details.
- alpha: A single value or vector of values for \( \alpha \).
- pert: Logical. If the input points are not in general position and pert is set to TRUE the observations are perturbed by adding random noise, see Details.
- eps: Scaling factor used for data perturbation when the input points are not in general position, see Details.

Details

If \( x \) is an object of class "ashape3d", then ashape3d does not recompute the 3D Delaunay triangulation (it reduces the computational cost).

If the input points \( x \) are not in general position and pert is set to TRUE, the function adds random noise to the data. The noise is generated from a normal distribution with mean zero and standard deviation \( \text{eps} \times \text{sd}(x) \).

Value

An object of class "ashape3d" with the following components (see Edelsbrunner and Mucke (1994) for notation):

- tetra: For each tetrahedron of the 3D Delaunay triangulation, the matrix tetra stores the indices of the sample points defining the tetrahedron (columns 1 to 4), a value that defines the intervals for which the tetrahedron belongs to the \( \alpha \)-complex (column 5) and for each \( \alpha \) a value (1 or 0) indicating whether the tetrahedron belongs to the \( \alpha \)-shape (columns 6 to last).

- triang: For each triangle of the 3D Delaunay triangulation, the matrix triang stores the indices of the sample points defining the triangle (columns 1 to 3), a value (1 or 0) indicating whether the triangle is on the convex hull (column 4), a value (1 or 0) indicating whether the triangle is attached or unattached (column 5), values that define the intervals for which the triangle belongs to the \( \alpha \)-complex (columns 6 to 8) and for each \( \alpha \) a value (0, 1, 2 or 3) indicating, respectively, that
the triangle is not in the $\alpha$-shape or it is interior, regular or singular (columns 9 to last). As defined in Edelsbrunner and Mucke (1994), a simplex in the $\alpha$-complex is interior if it does not belong to the boundary of the $\alpha$-shape. A simplex in the $\alpha$-complex is regular if it is part of the boundary of the $\alpha$-shape and bounds some higher-dimensional simplex in the $\alpha$-complex. A simplex in the $\alpha$-complex is singular if it is part of the boundary of the $\alpha$-shape but does not bounds any higher-dimensional simplex in the $\alpha$-complex.

**edge**

For each edge of the 3D Delaunay triangulation, the matrix `edge` stores the indices of the sample points defining the edge (columns 1 and 2), a value (1 or 0) indicating whether the edge is on the convex hull (column 3), a value (1 or 0) indicating whether the edge is attached or unattached (column 4), values that define the intervals for which the edge belongs to the $\alpha$-complex (columns 5 to 7) and for each $\alpha$ a value (0, 1, 2 or 3) indicating, respectively, that the edge is not in the $\alpha$-shape or it is interior, regular or singular (columns 8 to last).

**vertex**

For each sample point, the matrix `vertex` stores the index of the point (column 1), a value (1 or 0) indicating whether the point is on the convex hull (column 2), values that define the intervals for which the point belongs to the $\alpha$-complex (columns 3 and 4) and for each $\alpha$ a value (1, 2 or 3) indicating, respectively, if the point is interior, regular or singular (columns 5 to last).

**x**

A 3-column matrix with the coordinates of the original sample points.

**alpha**

A single value or vector of values of $\alpha$.

**xpert**

A 3-column matrix with the coordinates of the perturbated sample points (only when the input points are not in general position and `pert` is set to TRUE).

**References**


**Examples**

```r
T1 <- rtorus(1000, 0.5, 2)
T2 <- rtorus(1000, 0.5, 2, ct = c(2, 0, 0), rotx = pi/2)
x <- rbind(T1, T2)
# Value of alpha
alpha <- 0.25
# 3D alpha-shape
ashape3d.obj <- ashape3d(x, alpha = alpha)
plot(ashape3d.obj)

# For new values of alpha, we can use ashape3d.obj as input (faster)
alpha <- c(0.15, 1)
ashape3d.obj <- ashape3d(ashape3d.obj, alpha = alpha)
plot(ashape3d.obj, indexAlpha = 2:3)
```
components_ashape3d  

Description

This function calculates and clusters the different connected components of the $\alpha$-shape of a given sample of points in the three-dimensional space.

Usage

components_ashape3d(as3d, indexAlpha = 1)

Arguments

as3d  An object of class "ashape3d" that represents the $\alpha$-shape of a given sample of points in the three-dimensional space, see ashape3d.

indexAlpha  A single value or vector with the indexes of as3d$alpha that should be used for the computation, see Details.

Details

The function components_ashape3d computes the connected components of the $\alpha$-shape for each value of $\alpha$ in as3d$alpha[indexAlpha] when indexAlpha is numeric.

If indexAlpha="all" or indexAlpha="ALL" then the function computes the connected components of the $\alpha$-shape for all values of $\alpha$ in as3d$alpha.

Value

If indexAlpha is a single value then the function returns a vector $v$ of length equal to the sample size. For each sample point $i$, $v[i]$ represents the label of the connected component to which the point belongs (for isolated points, $v[i]=-1$). The labels of the connected components are ordered by size where the largest one (in number of vertices) gets the smallest label which is one.

Otherwise components_ashape3d returns a list of vectors describing the connected components of the $\alpha$-shape for each selected value of $\alpha$.

See Also

ashape3d, plot.ashape3d

Examples

T1 <- rtorus(1000, 0.5, 2)
T2 <- rtorus(1000, 0.5, 2, ct = c(2, 0, 0), rotx = pi/2)
x <- rbind(T1, T2)
alpha <- c(0.25, 2)
ashape3d.obj <- ashape3d(x, alpha = alpha)
### inashape3d

Test of the inside of an \( \alpha \)-shape

#### Description

This function checks whether points are inside an \( \alpha \)-shape.

#### Usage

```r
inashape3d(as3d, indexAlpha = 1, points)
```

#### Arguments

- **as3d**: An object of class "ashape3d" that represents the \( \alpha \)-shape of a given sample of points in the three-dimensional space, see `ashape3d`.
- **indexAlpha**: A single value or vector with the indexes of `as3d$alpha` that should be used for the computation, see Details.
- **points**: A 3-column matrix with the coordinates of the input points.

#### Details

The function `inashape3d` checks whether each point in `points` is inside the \( \alpha \)-shape for each value of \( \alpha \) in `as3d$alpha[indexAlpha]`.

If `indexAlpha="all"` or `indexAlpha="ALL"` then the function checks whether each point in `points` is inside the \( \alpha \)-shape for all values of \( \alpha \) in `as3d$alpha`.

#### Value

If `indexAlpha` is a single value then the function returns a vector of boolean of length the number of input points. The element at position \( i \) is TRUE if the point in `points[i,]` is inside the \( \alpha \)-shape. Otherwise `inashape3d` returns a list of vectors of boolean values (each object in the list as described above).
### Description

This function plots the $\alpha$-shape in 3D using the package `rgl`.

### Usage

```r
## S3 method for class 'ashape3d'
plot(x, clear = TRUE, col = c(2, 2, 2),
     byComponents = FALSE, indexAlpha = 1, transparency = 1,
     walpha = FALSE, triangles = TRUE, edges = TRUE, vertices = TRUE, ...)
```

### Arguments

- **x**
  An object of class "ashape3d" that represents the $\alpha$-shape of a given sample of points in the three-dimensional space, see `ashape3d`.

- **clear**
  Logical, specifying whether the current rgl device should be cleared.

- **col**
  A vector of length three specifying the colors of the triangles, edges and vertices composing the $\alpha$-shape, respectively.

- **byComponents**
  Logical, if TRUE the connected components of the $\alpha$-shape are represented in different colors, see `components_ashape3d`.

- **indexAlpha**
  A single value or vector with the indexes of `x$alpha` that should be used for the computation, see Details.

- **transparency**
  The coefficient of transparency, from 0 (transparent) to 1 (opaque), used to plot the $\alpha$-shape.

- **walpha**
  Logical, if TRUE the value of $\alpha$ is displayed in the rgl device.
rtorus

Generate points in the torus

Description

This function generates \( n \) random points within the torus whose minor radius is \( r \), major radius is \( R \) and center is \( ct \).

Usage

```r
torus(n, r, R, ct = c(0, 0, 0), rotx = NULL)
```
surfaceNormals

Arguments

- **n**: Number of observations.
- **r**: Minor radius (radius of the tube).
- **R**: Major radius (distance from the center of the tube to the center of the torus).
- **ct**: A vector with the coordinates of the center of the torus.
- **rotX**: If not NULL, a rotation through an angle `rotX` (in radians) about the x-axis is performed.

Examples

```r
T1 <- rtorus(2000, 0.5, 2.5)
rgl.bbox()
rgl.points(T1, col = 4)

T2 <- rtorus(2000, 0.5, 2.5, ct = c(2, 0, 0.5), rotX = pi/2)
rgl.points(T2, col = 2)
```

Description

This function calculates the normal vectors of all the triangles which belong to the boundary of the α-shape.

Usage

```r
surfaceNormals(x, indexAlpha = 1, display = FALSE, col = 3, scale = 1, ...)
```

Arguments

- **x**: An object of class "asshape3d" that represents the α-shape of a given sample of points in the three-dimensional space, see `asshape3d`.
- **indexAlpha**: A single value or vector with the indexes of `as3d$alpha` that should be used for the computation, see Details.
- **display**: Logical, if TRUE, `surfaceNormals` open a new rgl device and display the related α-shape and its normals vectors.
- **col**: Color of the normal vectors.
- **scale**: Scale parameter to control the length of the surface normals, only affect display.
- **...**: Material properties. See `rgl.material` for details.
Details

The function `surfaceNormals` computes the normal vectors of all the triangles which belong to the boundary of the $\alpha$-shape for each value of $\alpha$ in `x$alpha[indexAlpha]`. The magnitude of each vector is equal to the area of its associated triangle.

If `indexAlpha="all"` or `indexAlpha="ALL"` then the function computes the normal vectors for all values of $\alpha$ in `as3d$alpha`.

Value

If `indexAlpha` is a single value then the function returns an object of class "normals" with the following components:

- `normals`: Three-column matrix with the euclidean coordinates of the normal vectors.
- `centers`: Three-column matrix with the euclidean coordinates of the centers of the triangles that form the $\alpha$-shape.

Otherwise `surfaceNormals` returns a list of class "normals-List" (each object in the list as described above).

See Also

`ashape3d`

Examples

```r
x <- rtorus(1000, 0.5, 1)
alpha <- 0.25
ashape3d.obj <- ashape3d(x, alpha = alpha)
surfaceNormals(ashape3d.obj, display = TRUE)
```

---

volume_ashape3d

### Volume computation

Description

This function calculates the volume of the $\alpha$-shape of a given sample of points in the three-dimensional space.

Usage

```r
volume_ashape3d(as3d, byComponents = FALSE, indexAlpha = 1)
```
volume_ashape3d

Arguments

as3d An object of class "ashape3d" that represents the $\alpha$-shape of a given sample of points in the three-dimensional space, see ashape3d.

byComponents Logical, if FALSE (default) volume_ashape3d computes the volume of the whole $\alpha$-shape. If TRUE, volume_ashape3d computes the volume of each connected component of the $\alpha$-shape separately.

indexAlpha A single value or vector with the indexes of as3d$alpha that should be used for the computation, see Details.

Details

The function volume_ashape3d computes the volume of the $\alpha$-shape for each value of $\alpha$ in as3d$alpha[indexAlpha] when indexAlpha is numeric.

If indexAlpha="all" or indexAlpha="ALL" then the function computes the volume of the $\alpha$-shape for all values of $\alpha$ in as3d$alpha.

Value

If indexAlpha is a single value then the function returns the volume of the $\alpha$-shape (if the argument byComponents is set to FALSE) or a vector with the volumes of each connected component of the $\alpha$-shape (if the argument byComponents is set to TRUE).

Otherwise volume_ashape3d returns a list (each object in the list as described above).

See Also

ashape3d, components_ashape3d

Examples

C1 <- matrix(runif(6000), nc = 3)
C2 <- matrix(runif(6000), nc = 3) + 2
x <- rbind(C1, C2)
ashape3d.obj <- ashape3d(x, alpha = 0.75)
plot(ashape3d.obj, byComponents = TRUE)

# Compute the volume of the alpha-shape
volume_ashape3d(ashape3d.obj)
# Compute the volumes of the connected components of the alpha-shape
volume_ashape3d(ashape3d.obj, byComponents = TRUE)
Index

ashape3d, 2, 4–10
components_ashape3d, 4, 6, 7, 10
inashape3d, 5
plot.ashape3d, 4, 6
rgl, 6
rgl.material, 7, 8
rtorus, 7
surfaceNormals, 8
volume_ashape3d, 9