# Package 'alphashape3d'

May 7, 2026

**Version** 1.3.3

**Date** 2025-10-10

**Title** Implementation of the 3D Alpha-Shape for the Reconstruction of 3D Sets from a Point Cloud

**Depends** geometry, rgl

**Imports** RANN

**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

**Encoding** UTF-8

**Author** Beatriz Pateiro-López [aut, cre], Thomas Lafarge [aut]

**Maintainer** Beatriz Pateiro-López <beatriz.pateiro@usc.es>

**Repository** CRAN

**Date/Publication** 2025-10-10 12:10:02 UTC

## Contents

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 it 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 eps*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

Edelsbrunner, H., Mucke, E. P. (1994). Three-Dimensional Alpha Shapes. *ACM Transactions on Graphics*, 13(1), pp.43-72.

## 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)
# 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 *Connected subsets computation*

## 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)
plot(ashape3d.obj, indexAlpha = "all")
```

```
# Connected components of the alpha-shape for both values of alpha
comp <- components_ashape3d(ashape3d.obj, indexAlpha = "all")
class(comp)
# Number of components and points in each component for alpha=0.25
table(comp[[1]])
# Number of components and points in each component for alpha=2
table(comp[[2]])

# Plot the connected components for alpha=0.25
plot(ashape3d.obj, byComponents = TRUE, indexAlpha = 1)
```

---

`inashape3d` *Test of the inside of an $\alpha$-shape*

---

## Description

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

## Usage

```
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).

## See Also

ashape3d

## Examples

```
T1 <- rtorus(2000, 0.5, 2)
T2 <- rtorus(2000, 0.5, 2, ct = c(2, 0, 0), rotx = pi/2)
x <- rbind(T1, T2)
ashape3d.obj <- ashape3d(x, alpha = 0.4)
# Random sample of points in a plane
points <- matrix(c(5*runif(10000) - 2.5, rep(0.01, 5000)), nc = 3)
in3d <- inashape3d(ashape3d.obj, points = points)
plot(ashape3d.obj, transparency = 0.2)
colors <- ifelse(in3d, "blue", "green")
points3d(points, col = colors)
```

---

# plot.ashape3d *Plot the $\alpha$-shape in 3D*

---

## Description

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

## Usage

```
## 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. |
| triangles | Logical, if TRUE triangles are plotted. |

| | |
|---|---|
| edges | Logical, if TRUE edges are plotted. |
| vertices | Logical, if TRUE vertices are plotted. |
| ... | Material properties. See material3d for details. |

## Details

The function plot.ashape3d opens a rgl device for each value of $\alpha$ in x$alpha[indexAlpha]. Device information is displayed in the console.

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

## See Also

ashape3d, components_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(0.15, 0.25, 1)
ashape3d.obj <- ashape3d(x, alpha = alpha)

# Plot the alpha-shape for all values of alpha
plot(ashape3d.obj, indexAlpha = "all")

# Plot the connected components of the alpha-shape for alpha=0.25
plot(ashape3d.obj, byComponents = TRUE, indexAlpha = 2)
```

---

# 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

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

### 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

```
T1 <- rtorus(2000, 0.5, 2.5)
bbox3d(color = c("white", "black"))
points3d(T1, col = 4)


T2 <- rtorus(2000, 0.5, 2.5, ct = c(2, 0, 0.5), rotx = pi/2)
points3d(T2, col = 2)
```

---

## surfaceNormals *Normal vectors computation*

---

### Description

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

### Usage

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

### 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. |
| 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 $\alpha$-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 material3d 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

```
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

```
volume_ashape3d(as3d, byComponents = FALSE, 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. |
| 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