R Implementation of a Polyhedral Approximation to a 3D Set of Points Using the α-shape

Thomas Lafarge
National Institute of Standards and Technology

Beatriz Pateiro-López
Universidad de Santiago de Compostela

Antonio Possolo
National Institute of Standards and Technology

Joy P. Dunkers
National Institute of Standards and Technology

Abstract

This introduction to the R package **alphashape3d** is a (slightly) modified version of Lafarge, Pateiro-López, Possolo, and Dunkers (2014), published in the *Journal of Statistical Software*. This work presents the implementation in R of the α-shape of a finite set of points in the three–dimensional space $\mathbb{R}^3$. This geometric structure 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 α-shape, the R package **alphashape3d** provides users with tools to facilitate the three–dimensional graphical visualization of the estimated set as well as the computation of important characteristics such as the connected components or the volume, among others.

Keywords: set estimation, 3D surface reconstruction, α-convexity, α-shape, R software.

1. Introduction

Point sets are a fundamental source of information in many disciplines: they may be samples from probability distributions, portions of realizations of point processes observed within bounded sampling windows, or have substantive meaning (for example, landmarks in industrial parts or in biological structures). In many cases, a primary goal is to recover the size and shape of the set the points originate from (set estimation). In some applications the set is known but a sample of points is taken from it anyway, and used as a discrete approximation that facilitates the estimation of selected geometric attributes of the set (morphometry).

Besides the long-standing models for two-dimensional (2D) data, there is now an increasing interest in modeling techniques for three-dimensional (3D) data. Widely available technolo-

gies for measurement of 3D structures — including 3D laser-scanning, confocal microscopy, and computed tomography from nano to macro scales — are generating high-quality digital datasets that may be point clouds, or sets whose geometry can be probed using point clouds, in cellular biology, medicine, material science, engineering, etc. Consequently, a rich branch of image analysis has been developing rapidly to deal with the reconstruction of 3D objects from point clouds, and to study geometrical attributes after discretization by sampling with point clouds.

Statistical morphometry seeks to characterize the variability of shape and size, which is essential in a wide range of disciplines that include biology, material science, geology, archaeology, geography, etc. We refer to Small (1996) and Grenander and Miller (2007) for an extensive review of the methods and applications of shape analysis. See also Claude (2008) for a guide on how to perform morphometrics with R. The geometric characterization of an object includes the computation of volumes and surface areas, surface roughness, deviations from convexity, porosity and permeability, among many other attributes, in many cases from points sampled within the object. In practice, the user defines a sampling procedure to obtain points in the set of interest, and thus the problem relates to the problem of set estimation.

The goal of set estimation is to produce estimates of compact subsets of an Euclidean space based on random samples of points from probability distributions whose supports are those sets, and then to study the statistical properties of the corresponding estimators (asymptotic properties, types of convergence, convergence rates, etc.). Cuevas and Fraiman (2009) surveys the most relevant theory on set estimation.

There are different geometrical structures that can capture the shape of a set from a sample of points taken from it. The extent to which these structures manage faithfully to reproduce the original set depends heavily on the geometrical characteristics of the set. For instance, if the original set is convex, the convex hull of the sample is the natural estimator. See Geffroy (1964), Rényi and Sulanke (1963, 1964), Schneider (1988), Dümbgen and Walther (1996) for results about the convex hull estimator. However, in many applications the sets or objects of interest are not strictly convex, but only approximately so. There is, therefore, a need to define relaxed convexity shape restrictions.

The α-convexity is one such restriction that mildly relaxes the assumption of convexity, and does so in a tunable manner (determined by the value of the parameter α): it appears in the literature on set estimation, see Walther (1997, 1999), Rodríguez-Casal (2007). By analogy with the characterization of a convex set as the intersection of half-spaces defined by planes tangent to the set, a set is said to be α-convex if it can be expressed as an intersection of the complements of open balls of radius α not intersecting the set. Section 2 provides details on α-convex sets. The so-called α-convex hull of the sample is a suitable estimator of α-convex sets (think of what's left if carving out all the space with a spherical scoop of radius α without removing any of the points in the sample). Rodríguez-Casal (2007) studies the asymptotic performance of this estimator.

The boundary of the α-convex hull comprises arcs of circles (in 2D), or spherical caps (in 3D), and intersections thereof. If now we approximate the boundary of the α-convex hull by a polygonal curve (in 2D) or by a polyhedral surface (in 3D), then we get another object that approximates the original set, called the α-shape. Edelsbrunner, Kirkpatrick, and Seidel (1983) defined it in the case of $\mathbb{R}^2$, and Edelsbrunner and Mücke (1994) generalized it to three-dimensional Euclidean space. The α-shapes have been widely used in several different

fields: for example, to characterize biological systems, and the structure, surface and cavities of molecules and proteins, as described by Liang, Edelsbrunner, Fu, Sudhakar, and Subramaniam (1998a,b), Edelsbrunner and Facello (1998), Zhou and Yan (2010), among others. Also in cosmology, by van de Weygaert, Platen, Vegter, Eldering, and Kruithof (2010), to study the topology of the distribution of galaxies. Vauhkonen, Tokola, Maltamo, and Packalén (2008) describe applications of α-shapes in remote sensing, where tree characteristics are determined from laser scanning data. There are also connections between α-shapes and pattern recognition and clustering techniques. Given a porous material, the α-shape of a sample of points drawn from the material can be used to characterize the geometry of the pores; alternatively, the α-shape of a sample of points drawn from the voids can be used to study the permeability of the material.

It is becoming increasingly clear that R (R Core Team (2013)) has grown to become one of the most powerful tools for statistical data analysis, and in fact the *lingua franca* for the interchange of ideas and methods in computational statistics and probability. R is a high-level language that offers powerful graphical capabilities and a wide variety of facilities for statistical modeling and data analysis in many fields of research. Many of the techniques discussed in this paper have been developed over the last couple of decades, and there exist several R packages that, in one form or another, address issues of set estimation. For example, the convex hull of a set of points in d–dimensional Euclidean space can be computed using functions in package **geometry** (Barber, Habel, Grasman, Gramacy, Stahel, and Sterratt 2013). Objects analyzed in the disciplines mentioned above are mostly two-dimensional or three-dimensional. The two-dimensional α-shape and the α-convex hull are implemented in the R package **alphahull** by Pateiro-Lopez and Rodriguez-Casal. (2011). Edelsbrunner *et al.* (1983) describe the algorithm for the computation of the α-shape and α-convex hull in 2D. Pateiro-López and Rodríguez-Casal (2010) provide further details on the library and applications. However, and in spite of its potential usefulness in statistical analysis, the α-shape in 3D had not yet been implemented in R. Cognizant of the importance of the development of a free, open-source software environment for statistical computing, we have produced the R package **alphashape3d**, see Lafarge and Pateiro-Lopez (2014).

The paper is organized as follows. In Section 2 we provide some background in set estimation theory and describe the estimator under study, the α-shape of a random sample of points in the three-dimensional Euclidean space. In Section 3 we briefly describe the algorithm by Edelsbrunner and Mücke (1994) and give the details about its implementation in R. Further computational details are given in Section 4.

2. The α-shape in 3D

This work is based on the computation of α-shapes in $\mathbb{R}^3$ as defined by Edelsbrunner and Mücke (1994). The α-shape is computationally practicable, and it is closely related to the α-convex hull estimator, which has been studied predominantly from a theoretical viewpoint. As mentioned in the Introduction, in order to estimate an unknown set effectively, from points drawn from it, some geometrical assumptions have to be made. Even though convex sets have many useful and well-known properties, restricting attention to estimators that are convex sets rules out consideration of many sets of great practical interest. There are meaningful extensions of the notion of convexity, such as α-convexity. A set $A \subset \mathbb{R}^d$ is said to be α-convex,

for $\alpha > 0$, if $A = C_\alpha(A)$, where

$$C_\alpha(A) = \bigcap_{\{\mathring{B}(x,\alpha):\ \mathring{B}(x,\alpha)\cap A=\emptyset\}} \left(\mathring{B}(x,\alpha)\right)^c \tag{1}$$

is called the α-convex hull of A. In (1), $\mathring{B}(x,\alpha)$ denotes the open ball with center x and radius α. Walther (1999) describes properties of α-convex sets and relations between convexity and α-convexity. Let S be a nonempty compact subset in $\mathbb{R}^d$, equipped with the norm $\|\cdot\|$. Assume that we are given a random sample $\mathcal{X}_n = \{X_1, \ldots, X_n\}$ from a probability distribution P_X on $\mathbb{R}^d$ with support S. The goal is to approximate S based on the sample. If we assume that the set S is α-convex then the natural estimator of S is $C_\alpha(\mathcal{X}_n)$, the α-convex hull of the sample. Rodríguez-Casal (2007) characterizes the performance of the estimator. In Figure 1 we represent the α-convex hull estimator in the two-dimensional case. If we straighten the α-convex hull in Figure 1 by substituting straight edges for the circular arcs, we obtain a straight line graph known as α-shape: this is computationally more practicable than the α-convex hull. Edelsbrunner *et al.* (1983) give a formal definition of the α-shape in $\mathbb{R}^2$.

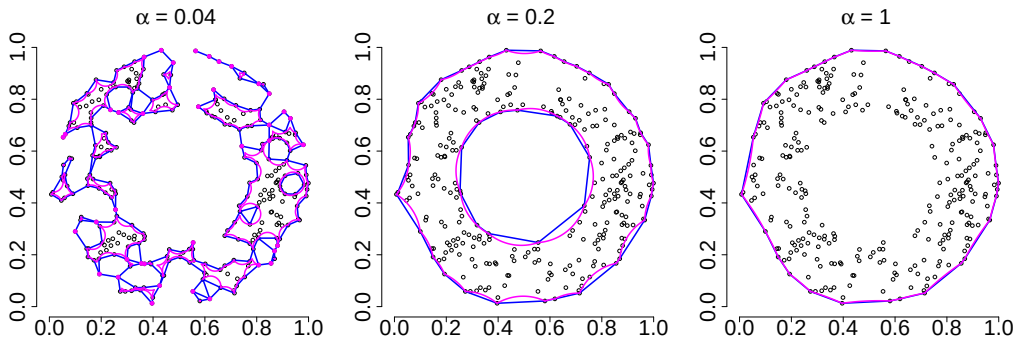

Figure 1: In pink, boundary of the α-convex hulls of a uniform random sample of size $n = 300$ on a ring in $\mathbb{R}^2$ for different values of α. In blue, the corresponding α-shapes.

In $\mathbb{R}^3$, one can visualize the α-convex hull estimator by thinking of a sphere of radius α scooping up space around the points without removing any of them. By replacing each spherical cap with the largest triangle circumscribed by the cap we obtain the α-shape, see Figure 2. The formal definition of the α-shape in 3D can be found in Edelsbrunner and Mücke (1994). It can be computed from the α-complex of the sample, also defined in Edelsbrunner and Mücke (1994), which is a subcomplex of the Delaunay triangulation. The α-shape is the polytope (in a general sense) formed by the union of all simplices of the α-complex. Further details are given in Section 3. The value of α controls the spatial scale of the shape of the estimator. As α decreases, the α-shape shrinks and cavities appear among the sample points. As α tends to ∞, the α-shape tends to the the convex hull of the sample, see Figure 2. This emphasizes the fact that the α-shape can be considered a generalization of the convex hull. The flexibility that varying α affords is very useful to characterize the geometry of porous materials. When studying some of the properties of these materials, it may be preferable to focus on the α-shape of the voids: for example, to characterize the geometry of tunnels that link different pores, as will be illustrated in Section **??**.

Figure 2: From left to right, α-shapes of a point cloud in $\mathbb{R}^3$ for increasing values of α.

3. Implementation

The R package **alphashape3d** comprises functions to compute, represent and display the α-shape of a given sample of points in three-dimensional Euclidean space. We restrict ourselves to point sets in $\mathbb{R}^3$ because we are mainly motivated by the applications of the α-shape in 3D, even though the definition of the α-shape can be generalized to higher dimensions (increasing the difficulty of the implementation).

The main function in the package is `ashape3d`, that implements in R the algorithm specified by Edelsbrunner and Mücke (1994). The α-shape is computed from the three-dimensional Delaunay triangulation of the point set. Subsection 3.1 describes the computation of that triangulation, and Section 3.2 describes the computation of the α-shape. Subsection 3.3 gives details about the three-dimensional graphical representation of the α-shape. Subsection 3.4 describes functions to compute and visualize the connected components of the α-shape. Subsection 3.5 describes other important features of the α-shape that can be computed with functions in the library **alphashape3d**.

3.1. Delaunay triangulation in 3D. The package geometry

The computation of the α-shape is closely related to the construction of the Delaunay triangulation of the given point set. Edelsbrunner *et al.* (1983) makes precise the relationship that exists between these two geometric structures in the bidimensional case by proving that the α-shape is a subgraph of the Delaunay triangulation. This simplifies the computation of the α-shape since we only have to consider the edges of the Delaunay triangulation as candidates to form the α-shape. The relationship between α-shape and Delaunay triangulation is also decisive in $\mathbb{R}^3$. The problem of triangulation in $\mathbb{R}^3$ arises in the decomposition of three-dimensional objects, for example, in solid modeling. The Delaunay triangulation is an essential geometric structure with a large number of applications, such as mesh generation and cluster analysis, crystallography, metallurgy, image processing, etc. For in-depth discussions of these problems we refer the reader to Aurenhammer (1991) and Preparata and Shamos (1985).

Similarly to the two-dimensional case, the Delaunay triangulation in $\mathbb{R}^3$ is defined as the dual of the Voronoi diagram (or, tessellation). Recall that the Voronoi diagram of a finite sample of points is the partition of space into cells such that each cell comprises the points which are

closer to one particular sample point than to any other sample point. In the simplest, two-dimensional case, the data are a sample of points in the plane, which are the Voronoi point sites. Each point site has associated a closed and convex Voronoi cell, defined by an open and convex Voronoi region, Voronoi edges and Voronoi vertices, as illustrated in Figure 3. In $\mathbb{R}^3$, each Voronoi cell is defined by a Voronoi region that is a three-dimensional convex polyhedron, Voronoi facets, Voronoi edges, and Voronoi vertices. The dual complex Delaunay "triangulation" consists of Delaunay tetrahedra and their triangular faces, edges and vertices, hence may more properly be called a "tetrahedralization".

An important feature in the Delaunay triangulation is that the circumsphere of each tetrahedron includes no other point sites (empty sphere property). The centers of these empty circumspheres are just the vertices of the Voronoi diagram, see Figure 3. Okabe, Boots, Sugihara, and Chiu (2009) provide an extensive discussion on the properties of Voronoi diagrams, Delaunay triangulations and their many applications in computational geometry. It is should be noted that for a set of n points in $\mathbb{R}^3$ the number of Delaunay tetrahedra generally is proportional to n^2.

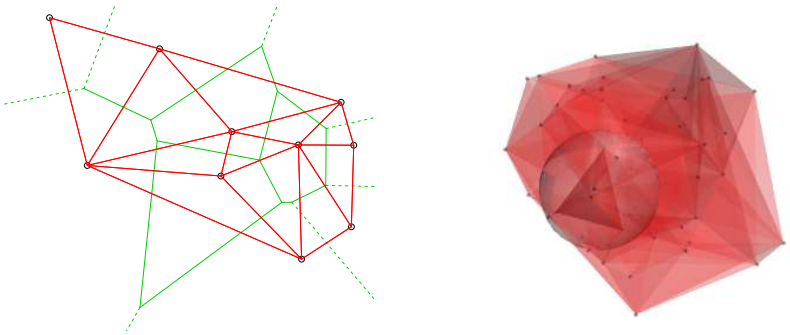

Figure 3: The left panel shows the Voronoi diagram of a set of point in $\mathbb{R}^2$: each point has an associated closed and convex Voronoi polygon (whose boundary is outlined in green), delimited by Voronoi edges abutting on Voronoi vertices; the corresponding Delaunay triangulation is drawn in red. The right panel shows the Delaunay triangulation of a uniform random sample of size $n = 50$ drawn from the unit cube in $\mathbb{R}^3$: the Delaunay diagram comprises tetrahedra and their triangular faces, edges and vertices; a tetrahedron's circumsphere, drawn in gray, contains no data points besides those at its vertices.

In R, the Delaunay triangulation in $\mathbb{R}^3$ can be computed with the package **geometry** (Barber *et al.* 2013), which makes the `qhull` library (`www.qhull.org`) available in R. The function `qhull` computes convex hulls, Delaunay triangulations, Voronoi vertices, farthest-site Voronoi vertices, and half-space intersections. Barber, Dobkin, and Huhdanpaa (1996) describe the Quickhull algorithm and review literature on the subject exhaustively. Discussion on efficient methods for the computation of Delaunay triangulation and Voronoi diagrams can be found in Aurenhammer and Klein (2000) and references therein.

The function `delaunayn` in package **geometry** plays an important role in the computation of the α-shape described in Section 3.2. Given a sample of points in $\mathbb{R}^3$, it returns for each

tetrahedron in the triangulation the indices of the sample points defining the tetrahedron.

3.2. The function `ashape3d`

Here we describe the function `ashape3d`, included in the library **alphashape3d**, that implements the algorithm to construct the three-dimensional α-shape in R. The function `ashape3d` computes the α-shape of a given sample $\mathcal{X}_n$ in $\mathbb{R}^3$ for given values of $\alpha > 0$. As an example to illustrate the use of the function we consider a point cloud of size $n = 4000$ within two linked tori, see Figure 4.

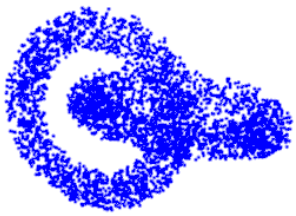

Figure 4: Point cloud of size $n = 4000$ within two linked tori.

Using function `rtorus`, we generated a uniform random sample of size 2000 in each of two tori with minor radius $r = 0.5$ and major radius $R = 2$. The computation of the α-shape requires that the input points be in the so-called *general position* (no quadruplet of points lies on the same plane, no quintuplet lies on the same sphere, and, for any particular α, the smallest sphere through any two, three, or four points has radius different from α). The function determines whether the given points are in such general position, and it gives the user the option of slightly disturbing their coordinates to remove the degeneracy and compute the α-shape, see Section 4.

```
R> n <- 4000
R> T1 <- rtorus(n/2, 0.5, 2)
R> T2 <- rtorus(n/2, 0.5, 2, ct = c(2, 0, 0), rotx = pi/3)
R> x <- rbind(T1, T2)
R> points3d(x, col = 4)
```

We compute the α-shape for $\alpha = 0.3$ and $\alpha = 0.5$.

```
R> alphashape3d <- ashape3d(x, alpha = c(0.3, 0.5))
R> class(alphashape3d)

[1] "ashape3d"

R> names(alphashape3d)
```

```
[1] "tetra"  "triang" "edge"   "vertex" "x"      "alpha"
```

The output of the function is an object of class 'ashape3d'. The components `x` and `alpha` store the input information. For each simplex (tetrahedron, triangle, edge or vertex) in the three-dimensional Delaunay triangulation there is a single interval so that the simplex is a face of the α-shape if and only if α is contained in this interval. Moreover, see Table I in Edelsbrunner and Mücke (1994)[1], each interval can be broken into three parts that classify a simplex as an interior, regular or singular simplex of the α-complex. As defined in Edelsbrunner and Mücke (1994), a simplex in the α-complex is interior if it does not belong to the boundary of the α-shape. A simplex in the α-complex is regular if it is part of the boundary of the α-shape and bounds some higher-dimensional simplex in the α-complex. A simplex in the α-complex is singular if it is part of the boundary of the α-shape but does not bounds any higher-dimensional simplex in the α-complex. This classification allows us to identify the simplices that define the boundary of the α-shape.

Throughout this paper, numerical results are listed as R would produce them by default, irrespective of whether all the digits reported are substantively significant.

```
R> head(alphashape3d$tetra)
```

```
   v1   v2   v3   v4        rhoT fc:0.3 fc:0.5
  782 1242 3203 3825   7.6716190      0      0
   22  868 2264 2349   1.9257596      0      0
 1027 3195 3203 3825 190.7969561      0      0
 2389 2519 3507 3540   0.5654926      0      0
 1182 1485 1522 2109  26.7345169      0      0
  102  473 2114 3792   2.6445879      0      0
```

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 `rhoT` that defines the intervals for which the tetrahedron belongs to the α-complex (column 5) and for each α a value (1 or 0) indicating whether the tetrahedron belongs to the α-shape (columns 6 to last). Thus, for the sample in Figure 5 (left), the tetrahedron defined by the points with indexes $(782, 1242, 3203, 3825)$ belongs to the α-shape for $\alpha > 7.6716$.

```
R> head(alphashape3d$triang)
```

```
 tr1 tr2  tr3 on.ch attached      rhoT       muT       MuT fc:0.3 fc:0.5
   1  15  722     0        1 0.1749262 0.1758546 0.1911751      1      1
   1  15  831     0        0 0.1382563 0.1384008 0.1453170      1      1
   1  15 1022     0        1 0.1625292 0.1691324 0.1758546      1      1
   1  15 1049     0        0 0.1523811 0.1691324 0.1773384      1      1
   1  15 1212     0        0 0.1016582 0.1028032 0.1587552      1      1
   1  15 1294     0        0 0.1094436 0.1261041 0.1911751      1      1
```

[1]The end-points of these sub-intervals, ρ_T, $\underline{\mu}_T$ and $\overline{\mu}_T$ defined in Edelsbrunner and Mücke (1994), are stored in columns `rhoT`, `muT` and `MuT`, respectively

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[2] (column 5), values `rhoT`, `muT` and `MuT` that define the intervals for which the triangle belongs to the α-complex (columns 6 to 8) and for each α a value (0, 1, 2 or 3) indicating, respectively, that the triangle is not in the α-shape or it is interior, regular or singular (columns 9 to last).

```
R> head(alphashape3d$edge)
```

```
     ed1 ed2 on.ch attached       rhoT        muT       MuT fc:0.3 fc:0.5
[1,]   1  15     0        0 0.05985055 0.10165816 0.1911751      1      1
[2,]   1 518     0        1 0.13164855 0.13227398 0.1522371      1      1
[3,]   1 540     0        1 0.08830179 0.14329671 0.1729808      1      1
[4,]   1 617     0        0 0.07735618 0.08851678 0.1684250      1      1
[5,]   1 722     0        1 0.16786176 0.17585456 0.1913174      1      1
[6,]   1 831     0        0 0.11897811 0.12239988 0.1522371      1      1
```

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 `rhoT`, `muT` and `MuT` that define the intervals for which the edge belongs to the α-complex (columns 5 to 7) and for each α a value (0, 1, 2 or 3) indicating, respectively, that the edge is not in the α-shape or it is interior, regular or singular (columns 8 to last).

```
R> head(alphashape3d$vertex)
```

```
     v1 on.ch        muT       MuT fc:0.3 fc:0.5
[1,]  1     0 0.05243670 0.1913174      1      1
[2,]  2     0 0.04260460 0.1764762      1      1
[3,]  3     0 0.04811865 0.1743870      1      1
[4,]  4     0 0.05273586 0.2433215      1      1
[5,]  5     0 0.07262488 0.1807329      1      1
[6,]  6     0 0.03250811 0.1653096      1      1
```

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 `muT` and `MuT` that define the intervals for which the point belongs to the α-complex (columns 3 and 4) and for each α a value (1, 2 or 3) indicating, respectively, if the point is interior, regular or singular (columns 5 to last).

3.3. Graphical representation of the α–shape in 3D

Graphical visualization is an essential part of data analysis and statistical modelling. The function `plot.ashape3d` (S3 method for class 'ashape3d') performs 3D visualization of the α-shape.

[2]See nomenclature of Edelsbrunner and Mücke (1994)

```
R> rgl.viewpoint(20)
R> rgl.light(120, 60)
R> bg3d("white")
R> plot(alphashape3d, col = c(4, 4, 4), indexAlpha = "all")
```

The input must be an object of class 'ashape3d'. The argument `col` is a vector of colour numbers for the triangles, edges and vertices composing the α-shape (the default colour is red). The argument `indexAlpha` can be a single value or a vector specifying the indexes of `alphashape3d$alpha` to be used for representing the α-shape. If `indexAlpha="all"` or `indexAlpha="ALL"` then the function opens a new rgl device for each value of α in `alphashape3d$alpha`. Other arguments are available. The argument `transparency` controls the transparency properties of the α-shape. Its value ranges from zero (transparent) to one (opaque). The scene and appearance of shapes, backgrounds and bounding box objects can be controlled by functions and parameters in package **rgl**, see Adler and Murdoch (2013). The package **rgl** adds three-dimensional, real-time visualization functionality to the R programming environment. Figure 5 shows snapshots of the above call to `plot.ashape3d`.

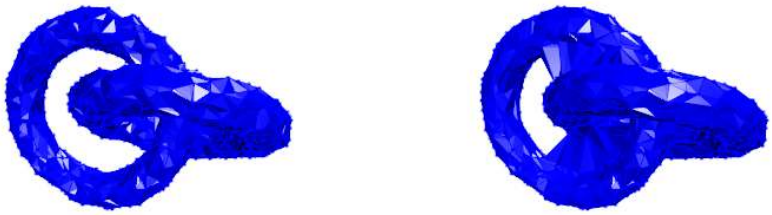

Figure 5: Snapshots from `plot.ashape3d` representing the α-shape of the point cloud for $\alpha = 0.3$ (middle) and $\alpha = 0.5$ (right).

3.4. Connected components of the α-shape

The library **alphashape3d** includes the function `components_ashape3d` that computes the connected components of the α-shape of a given point cloud in $\mathbb{R}^3$ and identifies the component that each sample point belongs to. We find this function potentially useful for partitioning point clouds into clusters. See Chapter 2 in Marchette (2004) for a discussion on the usefulness of the α-shape for clustering in the planar case. The visualization of the connected components of the α-shape provides clues about the reality of clusters in the data set. Owing to the characteristics of the α-shape, these clusters may have rather irregular shapes. Therefore, the connected components defined based on the α-shape may be more meaningful and accurate than those produced by classification algorithms that are entirely based on distance calculations.

In our implementation the α-shape is represented as a set of tetrahedrons, triangles, edges and vertices. This representation facilitates the computation of the adjacency matrix of the

vertices. Using this information, we can then identify the connected components of the α-shape using an algorithm inspired by the strongly connected components algorithm described in Tarjan (1972). Our algorithm is a modification of depth-first search (DFS) to label all the vertices of a component and then to jump to another component when needed.

For the example whose results are depicted in Figure 6, we compute the connected components corresponding to the α-shapes for $\alpha = 0.3$ and $\alpha = 0.5$ stored in `alphashape3d`.

```
R> comp <- components_ashape3d(alphashape3d, indexAlpha = "all")
```

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`). Otherwise, `components_ashape3d` returns a list of vectors describing the connected components of the α-shape for each selected value of α.

```
R> table(comp[[1]])

   1    2
2000 2000

R> table(comp[[2]])

   1
4000
```

This shows that the α-shape for $\alpha = 0.3$ has two connected components with 2000 sample points each, and that the α-shape for $\alpha = 0.5$ has only one connected component. The function `plot.ashape3d`, described in Subsection 3.3, displays the connected components of the α-shape in different colours by setting the argument `byComponents = TRUE`. The following code produces Figure 6.

```
R> rgl.viewpoint(20)
R> rgl.light(120, 60)
R> bg3d("white")
R> plot(alphashape3d, byComponents = TRUE, indexAlpha = "all")
```

3.5. Volume and other attributes of the α-shape

Here we describe other functions in the library **alphashape3d** that compute the values of several attributes of the α-shape. The function `volume_ashape3d` calculates the volume of the α-shape of a point cloud. The input must be an object of class 'ashape3d'. As described before, the argument `indexAlpha` can be a single value or a vector specifying the indexes of `alphashape3d$alpha` to be considered. For our example, we compute the volume of the α-shape for $\alpha = 0.3$.

```
R> volume_ashape3d(alphashape3d, indexAlpha = 1)
```

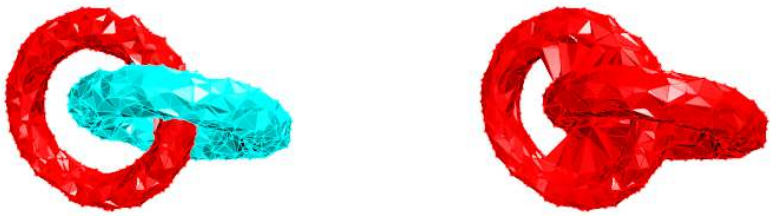

Figure 6: Connected components of the α-shape of the point cloud for $\alpha = 0.3$ (left) and $\alpha = 0.5$ (right).

```
[1] 14.75831
```

We can also compute the volume of each connected component of the α-shape separately, by setting the argument `byComponents = TRUE`.

```
R> volume_ashape3d(alphashape3d, indexAlpha = 1, byComponents = TRUE)
```

```
[1] 7.382174 7.376132
```

These add up to 14.7583, which is the total volume of the α-shape in Figure 6 (left).

The function `inashape3d` checks whether one or several points belong to the interior of the α-shape. For example, the point $(-2.5, 2.5, 0)$ does not belong to any of the α-shapes computed above.

```
R> inashape3d(alphashape3d, indexAlpha = "all", c(-2.5, 2.5, 0))
```

```
[[1]]
[1] FALSE

[[2]]
[1] FALSE
```

The following code produces Figure 7. We have considered a sequence of points in the line $y = -x$ ($z = 0$). After using the function `inashape3d` we represent in red the points that do not belong to the α-shape and in green the points that belong to the α-shape for $\alpha = 0.3$.

```
R> np <- 50
R> x <- seq(-2.5, 2.5, length = np)
R> points <- cbind(x, -x, rep(0, np))
R> in3d <- inashape3d(alphashape3d, indexAlpha = 1, points = points)
```

```
R> rgl.viewpoint(20)
R> bg3d("white")
R> plot(alphashape3d, col = c(4, 4, 4), indexAlpha = 1, transparency = 0.2)
R> colours <- ifelse(in3d, "green", "red")
R> rgl.points(points, col = colours)
```

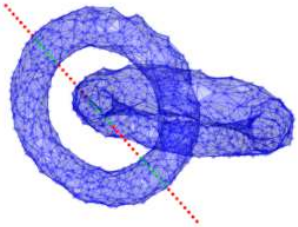

Figure 7: In blue, α-shape of the input point cloud for $\alpha = 0.3$. In red, points in the line $y = -x$ ($z = 0$) that do not belong to the α-shape. In green, points in the line $y = -x$ ($z = 0$) that belong to the α-shape.

The algorithm used to determine whether a given point p is inside the α-shape is an extension of the point-in-polygon ray-casting algorithm. The idea is to cast a ray from p in a randomly chosen direction and compute the intersections between the ray and the *regular* triangles of the α-shape. Recall that, according to the nomenclature of Edelsbrunner and Mücke (1994), a triangle is regular if it belongs to the boundary of the α-shape and is a face of one of its tetrahedra. If there is an odd number of intersections between the ray and the regular triangles, then the tested point p is inside the α-shape. If there is an even number of intersections, then p is outside the α-shape. To determine whether a half ray intersects a triangle, we compute the intersection of four half-spaces, which is considerably faster than testing whether p belongs to any of the tetrahedra that form the α-shape.

Finally, the function `surfaceNormals` calculates the normal vectors to the triangles in the boundary of the α-shape. For the considered example, we compute the normal vectors for the α-shape with $\alpha = 0.3$.

```
R> normv <- surfaceNormals(alphashape3d, indexAlpha = 1)
R> class(normv)

[1] "normals"

R> names(normv)

[1] "normals" "centers"
```

```
R> head(normv$normals)
```

```
              [,1]         [,2]        [,3]
[1,]  0.0029585072 -0.001143980 0.033783489
[2,] -0.0036817298 -0.010252499 0.044488124
[3,]  0.0044536541 -0.017187789 0.000885211
[4,]  0.0008754820 -0.015346566 0.008283824
[5,]  0.0034202517 -0.017412856 0.021822548
[6,] -0.0001668974 -0.002933998 0.023543594
```

```
R> head(normv$centers)
```

```
           [,1]     [,2]      [,3]
[1,] -0.3926686 1.695505 0.3534856
[2,] -0.2716099 1.803067 0.3716013
[3,] -0.3339178 1.664702 0.3294204
[4,] -0.1594104 1.657410 0.2680781
[5,] -0.1077608 1.681371 0.2988462
[6,] -0.1612096 1.796225 0.3783043
```

The function returns an object of class 'normals'. The component `normals` is a three-column matrix with the euclidean coordinates of the normal vectors (the norm of each vector is equal to the area of its associated triangle). The component `centers` is a three-column matrix with the euclidean coordinates of the centers of the triangles that form the boundary of the α-shape. The normal vectors can be represented by setting the argument `display = TRUE` in the function `surfaceNormals`.

4. Computational details

About the general position assumption

For the computation of the α-shape, the input points in function `ashape3d` must be in general position. Otherwise, the function `ashape3d` prints an error message reporting that the general position assumption is not satisfied. By setting the argument `pert = TRUE` the function `ashape3d` applies a slight perturbation to the input points in case they are nor in general position. Both, the original points and the perturbed data, are stored in components `x` and `xpert` of the output object, respectively.

Exporting as OFF files

Object File Format (.off) files are commonly used to represent the geometry of an object. OFF files represent collections of planar polygons with possibly shared vertices. They are specially useful to describe polyhedra and many 3D graphics programs use this format for input or output. An object of class 'ashape3d' can be easily exported as an OFF file. Each OFF file begins with the keyword "OFF". The next line includes the number of vertices, faces, and edges of the object. Next, the coordinates of the vertices are listed and following these are the face descriptions. The following code saves the α-shape computed in our example for

$\alpha = 0.3$ as an OFF file. OFF files can be visualized and manipulated quite easily using 3D viewing programs.

```
R> offFilename <- "alphashape3d.off"
R> write("OFF", file = offFilename, append = FALSE, sep = " ")
R> aux <- alphashape3d$triang
R> tr <- aux[aux[, 9] == 2 | aux[, 9]==3, c("tr1", "tr2", "tr3")]
R> write(c(dim(alphashape3d$x)[1], dim(tr)[1], 0), file = offFilename,
+  append = TRUE, sep = " ")
R> write(t(alphashape3d$x), file = offFilename, ncolumns = 3,
+  append = TRUE, sep = " ")
R> write(t(cbind(rep(3, dim(tr)[1]), tr - 1)),
+  file = offFilename, ncolumns = 4, append = TRUE, sep = " ")
```

Software

The **alphashape3d** package is available from the Comprehensive R Archive Network at the url http://CRAN.R-project.org/package=alphashape3d. This document was generated with version 1.1 of the package.

References

Adler D, Murdoch D (2013). ***rgl**: 3D Visualization Device System (OpenGL)*. R package version 0.93.935, URL http://CRAN.R-project.org/package=rgl.

Aurenhammer F (1991). "Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure." *ACM Comput. Surv.*, **23**(3), 345–405.

Aurenhammer F, Klein R (2000). "Voronoi Diagrams." In *Handbook of Computational Geometry*, pp. 201–290. North-Holland, Amsterdam.

Barber CB, Dobkin DP, Huhdanpaa H (1996). "The Quickhull Algorithm for Convex Hulls." *ACM Trans. Math. Softw.*, **22**, 469–483.

Barber CB, Habel K, Grasman R, Gramacy RB, Stahel A, Sterratt DC (2013). ***geometry**: Mesh Generation and Surface Tesselation*. R package version 0.3-3, URL http://CRAN.R-project.org/package=geometry.

Claude J (2008). *Morphometrics with R*. Springer-Verlag.

Cuevas A, Fraiman R (2009). *New Perspectives on Stochastic Geometry*, chapter Set estimation, pp. 366–385. Oxford University Press.

Dümbgen L, Walther G (1996). "Rates of Convergence for Random Approximations of Convex Sets." *Adv. in Appl. Probab.*, **28**(2), 384–393.

Edelsbrunner H, Facello M (1998). "On the Definition and the Construction of Pockets in Macromolecules." *Discrete Applied Mathematics*, **88**, 83–102.

Edelsbrunner H, Kirkpatrick DG, Seidel R (1983). "On the Shape of a Set of Points in the Plane." *IEEE Trans. Inform. Theory*, **29**(4), 551–559.

Edelsbrunner H, Mücke EP (1994). "Three-dimensional Alpha Shapes." *ACM Trans. Graph.*, **13**(1), 43–72.

Geffroy J (1964). "Sur un Problème d'estimation Géométrique." *Publ. Inst. Statist. Univ. Paris*, **13**, 191–210.

Grenander U, Miller MI (2007). *Pattern Theory: from Representation to Inference*. OUP Oxford.

Lafarge T, Pateiro-Lopez B (2014). ***alphashape3d**: Implementation of the 3D Alpha-Shape for the Reconstruction of 3D Sets from a Point Cloud*. R package version 1.1, URL http://CRAN.R-project.org/package=alphashape3d.

Lafarge T, Pateiro-López B, Possolo A, Dunkers J (2014). "R Implementation of a Polyhedral Approximation to a 3D Set of Points Using the α-Shape." *Journal of Statistical Software*, **56**(4), 1–19. ISSN 1548-7660. URL http://www.jstatsoft.org/v56/i04.

Liang J, Edelsbrunner H, Fu P, Sudhakar PV, Subramaniam S (1998a). "Analytical Shape Computation of Macromolecules: I." *Proteins-structure Function and Bioinformatics*, **33**, 117.

Liang J, Edelsbrunner H, Fu P, Sudhakar PV, Subramaniam S (1998b). "Analytical Shape Computation of Macromolecules: II. Inaccessible Cavities in Proteins." *Proteins-structure Function and Bioinformatics*, **33**, 18–29.

Marchette DJ (2004). *Random Graphs for Statistical Pattern Recognition*. John Wiley & Sons.

Okabe A, Boots B, Sugihara K, Chiu S (2009). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley Series in Probability and Statistics. John Wiley & Sons.

Pateiro-López B, Rodríguez-Casal A (2010). "Generalizing the Convex Hull of a Sample: The R Package **alphahull**." *Journal of Statistical Software*, **34**(5), 1–28.

Pateiro-Lopez B, Rodriguez-Casal A (2011). ***alphahull**: Generalization of the Convex Hull of a Sample of Points in the Plane*. R package version 0.2-1, URL http://CRAN.R-project.org/package=alphahull.

Preparata FP, Shamos MI (1985). *Computational Geometry: An Introduction*. Texts and Monographs in Computer Science. Springer-Verlag.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Rényi A, Sulanke R (1963). "Über die Konvexe Hülle von n Zufällig Gewählten Punkten." *Z. Wahrscheinlichkeitstheorie und Verw. Gebiete*, **2**, 75–84.

Rényi A, Sulanke R (1964). "Über die Konvexe Hülle von n Zufällig Gewählten Punkten. II." *Z. Wahrscheinlichkeitstheorie und Verw. Gebiete*, **3**, 138–147.

Rodríguez-Casal A (2007). "Set Estimation under Convexity Type Assumptions." *Annales de l'I.H.P.- Probabilités & Statistiques*, **43**, 763–774.

Schneider R (1988). "Random Approximation of Convex Sets." *J. Microscopy*, **151**, 211–227.

Small C (1996). *The Statistical Theory of Shape*. Springer series in statistics. Springer-Verlag.

Tarjan R (1972). "Depth-first Search and Linear Graph Algorithms." *SIAM J. Comput.*, **1**(2), 146–160.

van de Weygaert R, Platen E, Vegter G, Eldering B, Kruithof N (2010). "Alpha Shape Topology of the Cosmic Web." *International Symposium on Voronoi Diagrams in Science and Engineering*, **0**, 224–234. doi:http://doi.ieeecomputersociety.org/10.1109/ISVD.2010.24.

Vauhkonen J, Tokola T, Maltamo M, Packalén P (2008). "Effects of Pulse Density on Predicting Characteristics of Individual Trees of Scandinavian Commercial Species Using Alpha Shape Metrics Based on Airborne Laser Scanning Data." *Canadian Journal of Remote Sensing*, **34**, 441–459.

Walther G (1997). "Granulometric Smoothing." *Ann. Statist.*, **25**(6), 2273–2299.

Walther G (1999). "On a Generalization of Blaschke's Rolling Theorem and the Smoothing of Surfaces." *Math. Methods Appl. Sci.*, **22**(4), 301–316.

Zhou W, Yan H (2010). "A Discriminatory Function for Prediction of Protein-DNA Interactions Based on Alpha Shape Modeling." *Bioinformatics*, **26**, 2541–2548.

Affiliation:

Beatriz Pateiro-López.
Departamento de Estadística e Investigación Operativa
Facultad de Matemáticas. Rúa Lope Gómez de Marzoa s/n
15782 Santiago de Compostela, Spain
E-mail: beatriz.pateiro@usc.es