Computational topology: Lecture 18

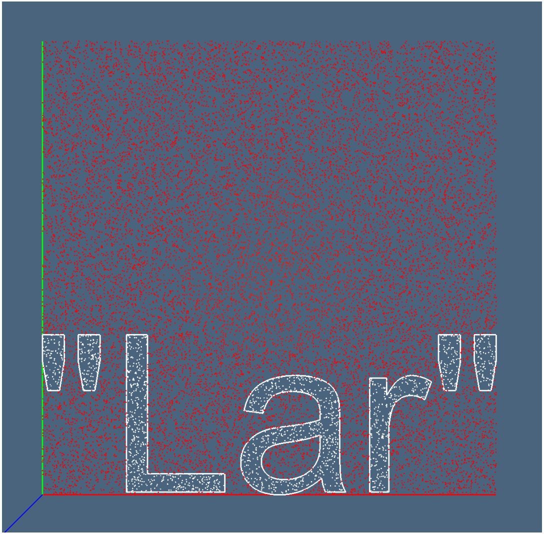
Alberto Paoluzzi

May 17, 2019

Implementing Alpha-shapes in 2D

Lab work: implement the Alpha complex of points

Get the points inside the polygonal cheracters and generate their parametric alpha shape



Alpha shape and Alpha complex (from Wikipedia)

In computational geometry, an alpha shape, or α-shape, is a family of piecewise linear simple curves in the Euclidean plane associated with the shape of a finite set of points

Alpha shapes are closely related to alpha complexes, subcomplexes of Delaunay triangulation of the point set.

- Each edge or triangle of the Delaunay triangulation may be associated with a characteristic radius, the radius of the smallest empty circle containing the edge or triangle

Alpha shape and Alpha complex (from Wikipedia)

In computational geometry, an alpha shape, or α-shape, is a family of piecewise linear simple curves in the Euclidean plane associated with the shape of a finite set of points

Alpha shapes are closely related to alpha complexes, subcomplexes of Delaunay triangulation of the point set.

- Each edge or triangle of the Delaunay triangulation may be associated with a characteristic radius, the radius of the smallest empty circle containing the edge or triangle
- For each real number α, the α-complex of the given set of points is the simplicial complex formed by the set of edges and triangles whose radii are at most $1/\alpha$.

Alpha shape and Alpha complex (from Wikipedia)

In computational geometry, an alpha shape, or α-shape, is a family of piecewise linear simple curves in the Euclidean plane associated with the shape of a finite set of points

Alpha shapes are closely related to alpha complexes, subcomplexes of Delaunay triangulation of the point set.

- Each edge or triangle of the Delaunay triangulation may be associated with a characteristic radius, the radius of the smallest empty circle containing the edge or triangle
- For each real number α, the α-complex of the given set of points is the simplicial complex formed by the set of edges and triangles whose radii are at most $1/\alpha$.
- The union of the edges and triangles in the α-complex forms a shape closely resembling the α-shape; however it differs in that it has polygonal edges rather than edges formed from arcs of circles

Implementation? (in class)

1. Start from a file
   `<yourrepo>/LinearAlgebraicRepresentation.jl/examples/2d/alphashape.jl`

[1] Useful link: table-of-8-bit-ascii-character-codes

Implementation? (in class)

1. Start from a file
   `<yourrepo>/LinearAlgebraicRepresentation.jl/examples/2d/alphashape.jl`
2. Insert the header:

```
using Plasm, Triangle
using LinearAlgebraicRepresentation
Lar = LinearAlgebraicRepresentation
```

[1] Useful link: table-of-8-bit-ascii-character-codes

Implementation? (in class)

1. Start from a file
   `<yourrepo>/LinearAlgebraicRepresentation.jl/examples/2d/alphashape.jl`
2. Insert the header:

```
using Plasm, Triangle
using LinearAlgebraicRepresentation
Lar = LinearAlgebraicRepresentation
```

3. Using some drawing program of your machine, write some graphic text:
   `"Lar"` is OK (including double quotes)[1]

[1] Useful link: table-of-8-bit-ascii-character-codes

Implementation? (in class)

1. Start from a file
`<yourrepo>/LinearAlgebraicRepresentation.jl/examples/2d/alphashape.jl`
2. Insert the header:

```
using Plasm, Triangle
using LinearAlgebraicRepresentation
Lar = LinearAlgebraicRepresentation
```

3. Using some drawing program of your machine, write some graphic text: `"Lar"` is OK (including double quotes)[1]
4. Make the graphic text fairly big, and export as .SVG file

[1] Useful link: table-of-8-bit-ascii-character-codes

Implementation? (in class)

1. Start from a file
   `<yourrepo>/LinearAlgebraicRepresentation.jl/examples/2d/alphashape.jl`
2. Insert the header:

```
using Plasm, Triangle
using LinearAlgebraicRepresentation
Lar = LinearAlgebraicRepresentation
```

3. Using some drawing program of your machine, write some graphic text:
   `"Lar"` is OK (including double quotes)[1]
4. Make the graphic text fairly big, and export as .SVG file
5. Import the file and transform in LAR model format using
   https://github.com/cvdlab/Plasm.jl/tree/julia-1.0/src

[1] Useful link: table-of-8-bit-ascii-character-codes

Implementation? (in class)

1. Start from a file
`<yourrepo>/LinearAlgebraicRepresentation.jl/examples/2d/alphashape.jl`
2. Insert the header:

```
using Plasm, Triangle
using LinearAlgebraicRepresentation
Lar = LinearAlgebraicRepresentation
```

3. Using some drawing program of your machine, write some graphic text: `"Lar"` is OK (including double quotes)[1]
4. Make the graphic text fairly big, and export as .SVG file
5. Import the file and transform in LAR model format using
https://github.com/cvdlab/Plasm.jl/tree/julia-1.0/src
6. Generate a random set of points inside the shape. HINT: use code from
https://github.com/cvdlab/LinearAlgebraicRepresentation.jl/blob/julia-1.0/examples/2d/svg2lar.jl

[1] Useful link: table-of-8-bit-ascii-character-codes

Implementation? (in class)

1. Start from a file
   `<yourrepo>/LinearAlgebraicRepresentation.jl/examples/2d/alphashape.jl`
2. Insert the header:

```
using Plasm, Triangle
using LinearAlgebraicRepresentation
Lar = LinearAlgebraicRepresentation
```

3. Using some drawing program of your machine, write some graphic text: `"Lar"` is OK (including double quotes)[1]
4. Make the graphic text fairly big, and export as .SVG file
5. Import the file and transform in LAR model format using https://github.com/cvdlab/Plasm.jl/tree/julia-1.0/src
6. Generate a random set of points inside the shape. HINT: use code from https://github.com/cvdlab/LinearAlgebraicRepresentation.jl/blob/julia-1.0/examples/2d/svg2lar.jl
7. finally write a parametric function ($0 \leq \alpha \leq 1$) to compute the α-complex of any set `V` of 2D points:

[1] Useful link: table-of-8-bit-ascii-character-codes