CGALmesh
A Simplicial Mesh Generator based on the library CGAL

Clément Jamin, Mariette Yvinec

INRIA Sophia Antipolis

Meshing day of the C2S@EXA project
Paris, May 2013

The library CGAL

http://www.cgal.org

The Computational
Geometric
Algorithms
Library

Mission statement

Make the large body of geometric algorithms developed in the field of computational geometry available for research and industrial applications.

5 European Funding projects

CGAL 96-97 GALIA 98-99
ECG ACS AIM@SHAPE

7 research institutes involved in development

ETH Zürich , INRIA Sophia Antipolis, MPI Saarbrücken, Tel Aviv University, Utrecht University , Groeningen University, Forth Heraklion

CGAL today

600 000	lines of C++ code
20	active developers
60	commercial users
2	public mailing lists :
	cgal_announce: 3 000 subscribers
	cgal_discuss: 1 000 subscribers
1	editorial board
6	month release cycle
daily	testsuite on 20 platforms

Licences

Open source : GPLv3+ (CGAL kernel), LGPLv3+

Commercial : selled by Geometry Factory

Interfaces

Scilab, Java, Python interfaces,

CGAL triangulations in Matlab

CGALmesh

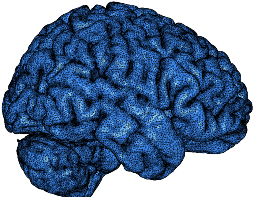
Why a CGAL package for mesh generation

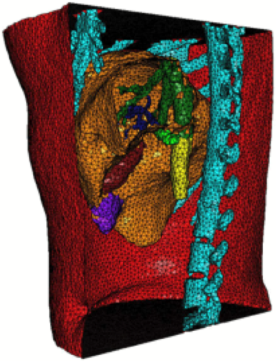
CGALmesh is a simplicial mesh generator

- Delaunay refinement
- Mesh optimization

The main advantages of CGALmesh

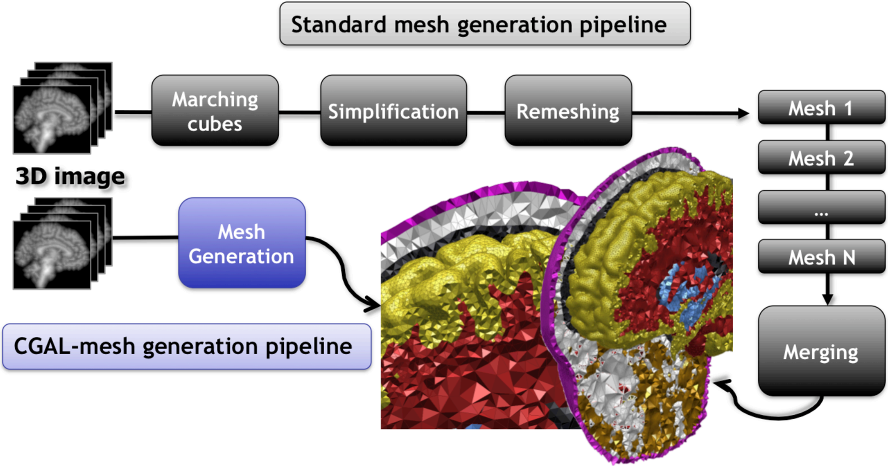
- It meshes 3D volumes bounded by smooth or piecewise smooth straight or curved surfaces.
- It meshes in the same process surfaces and 3D volumes, single or multidomains
- Highly flexible. Input may be: polyhedral, implicit surfaces 3D grey-level or segmented 3D images, ...



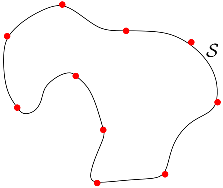


The main advantages of CGALmesh

Meshing surfaces and multivolums in a single process

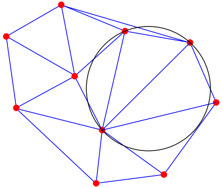


Restricted Delaunay triangulation



sampling P, surface $\mathcal{S}$

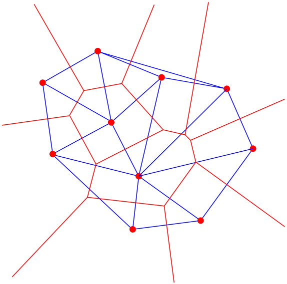
Restricted Delaunay triangulation



sampling P, surface $\mathcal{S}$

- $\mathrm{Del}(P)$ Delaunay triangulation

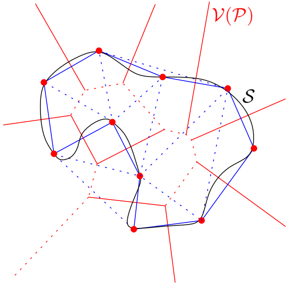
Restricted Delaunay triangulation



sampling P, surface $\mathcal{S}$

- $\mathrm{Del}(P)$ Delaunay triangulation
- $\mathrm{Vor}(P)$ its dual Voronoi diagram

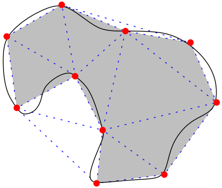
Restricted Delaunay triangulation



sampling P, surface $\mathcal{S}$

- $\mathrm{Del}(P)$ Delaunay triangulation
- $\mathrm{Vor}(P)$ its dual Voronoi diagram
- $\mathrm{Del}_{|\mathcal{S}}(P)$ facets of $\mathrm{Del}(P)$ whose dual intersect $\mathcal{S}$

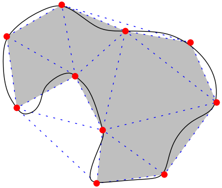
Restricted Delaunay triangulation



sampling P, surface $\mathcal{S}$

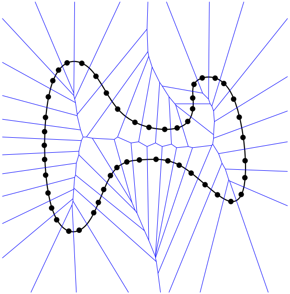
- $\mathrm{Del}(P)$ Delaunay triangulation
- $\mathrm{Vor}(P)$ its dual Voronoi diagram
- $\mathrm{Del}_{|\mathcal{S}}(P)$ facets of $\mathrm{Del}(P)$ whose dual intersect $\mathcal{S}$
- $\mathrm{Del}_{|O}(P)$ tetrahedra of $\mathrm{Del}(P)$ whose circumcenter lie in O

Restricted Delaunay triangulation



sampling P, surface $\mathcal{S}$

- $\mathrm{Del}(P)$ Delaunay triangulation
- $\mathrm{Vor}(P)$ its dual Voronoi diagram
- $\mathrm{Del}_{|\mathcal{S}}(P)$ facets of $\mathrm{Del}(P)$ whose dual intersect $\mathcal{S}$
- $\mathrm{Del}_{|O}(P)$ tetrahedra of $\mathrm{Del}(P)$ whose circumcenter lie in O



[ES 97, AB 98, BO 05]

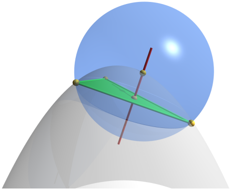
If P is dense enough on $\mathcal{S}$

- $\mathrm{Del}_{|\mathcal{S}}(P)$ is homeomorphic to $\mathcal{S}$
- Hausdorff-distance$(\mathrm{Del}_{|\mathcal{S}}(P), \mathcal{S}) \approx 0$
- $\mathrm{Del}_{|\mathcal{S}}(\mathcal{S})$ provides good estimation of normals, area, curvature
- $\mathrm{Del}_{|\mathcal{S}}(P)$ is the boundary of $\mathrm{Del}_{|\mathcal{S}}(P)$

Meshing algorithm

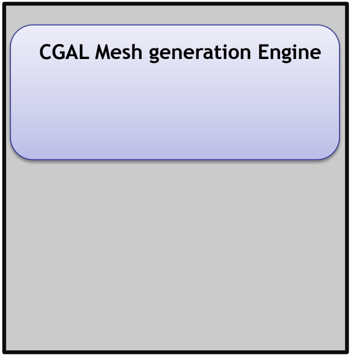
The algorithm

- Initialize P, $\mathrm{Del}_{|\mathcal{S}}(P)$ and $\mathrm{Del}_{|O}(P)$
- Delaunay refinement
  Apply the following rules with priority order.
  - Rule 1: If there is a bad facet f in $\mathrm{Del}_{|\mathcal{S}}(P)$,
    refine_facet(f)
  - Rule 2: If there is a bad tetrahedron t in $\mathrm{Del}_{|O}(P)$,
    refine_tetrahedron_or_facet(t)
- Mesh optimization

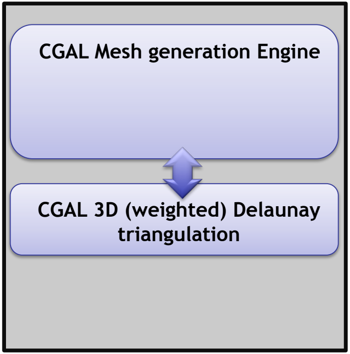


refine_facet(f)

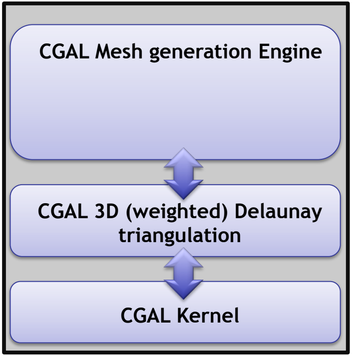
CGALmesh design



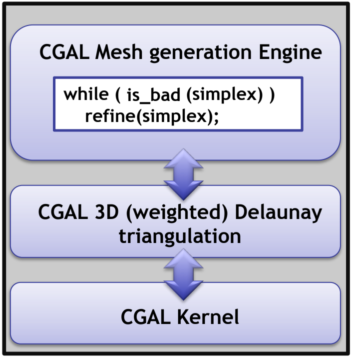
CGALmesh design



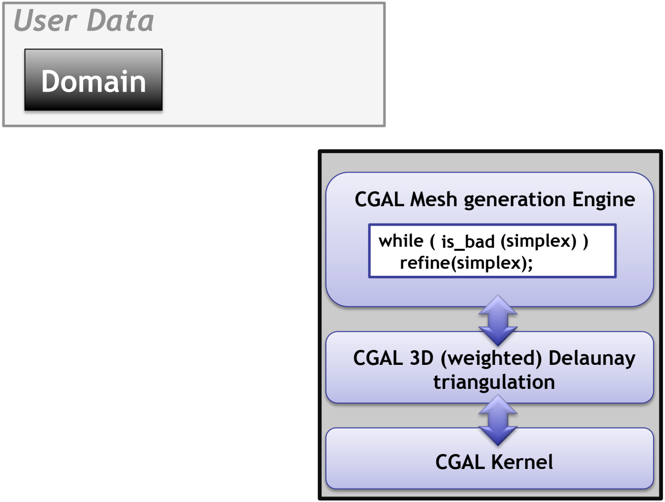
CGALmesh design



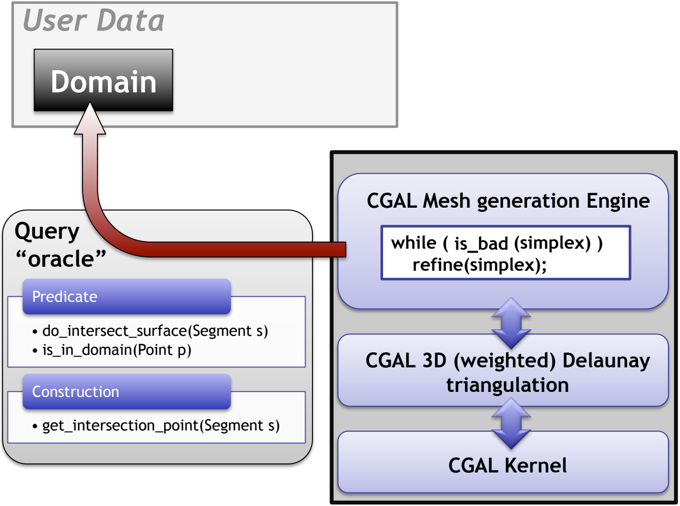
CGALmesh design



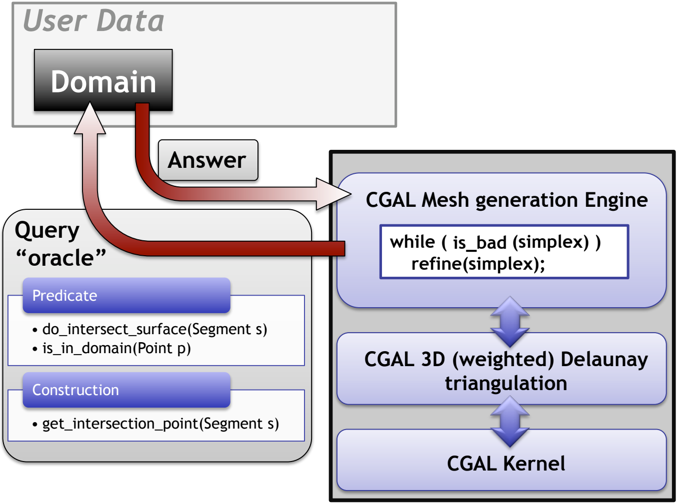
CGALmesh design



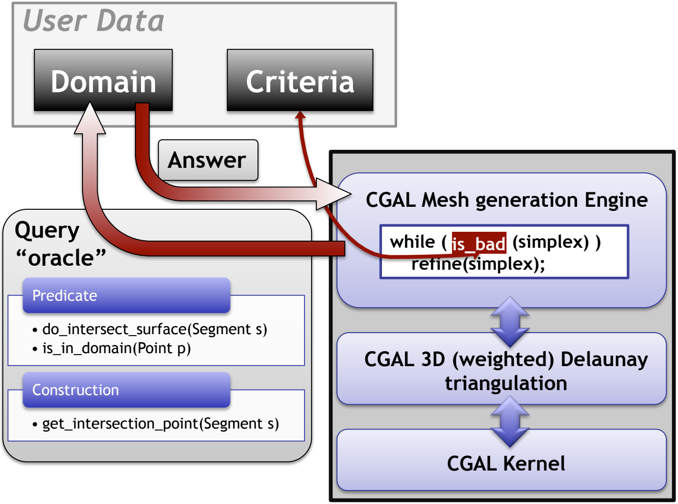
CGALmesh design



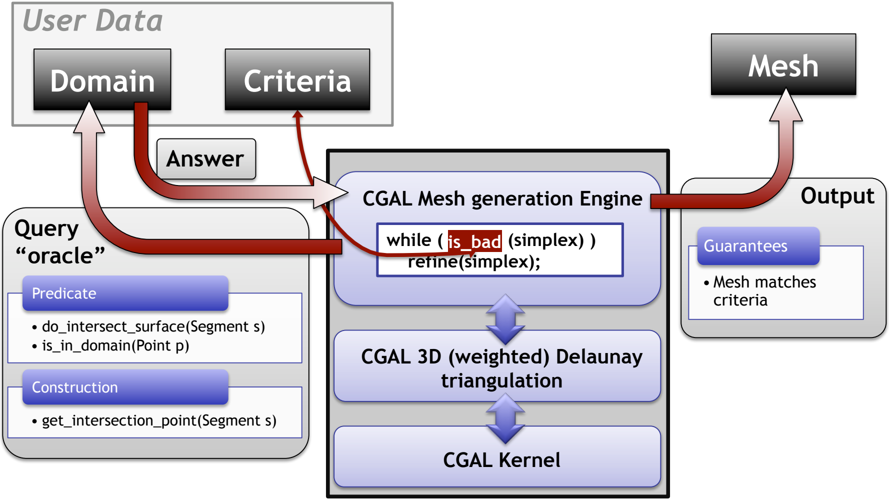
CGALmesh design



CGALmesh design



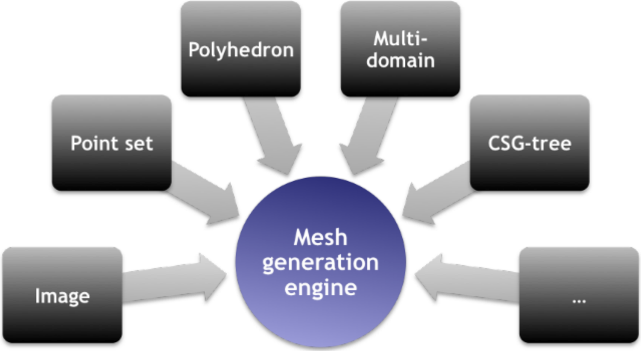
CGALmesh design



Flexibility : the domain oracle

Required interface with the domain is gathered in a single concept including:

- points initialization in domain and on surfaces
- intersections between segments (Voronoi edges) and surfaces
- location of points in domain and subdomains

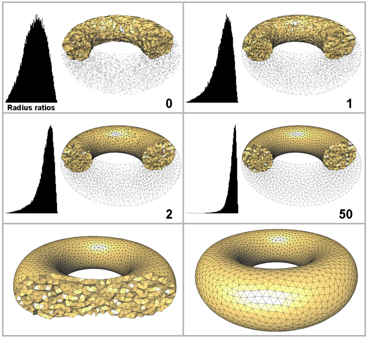


Mesh optimization

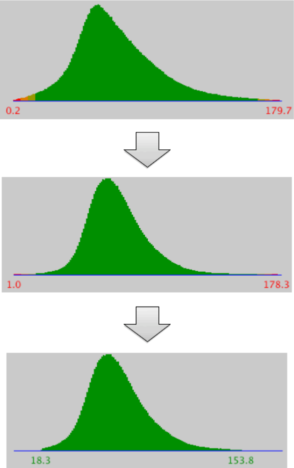
Optimization processes in CGALmesh

4 optimization processes can be sequentially combined.

- Global optimization processes
  - Lloyd relaxation
  - ODT relaxation (Optimal Delaunay Triangulation)
- Local optimization
  - Vertex perturbation
  - Sliver exudation



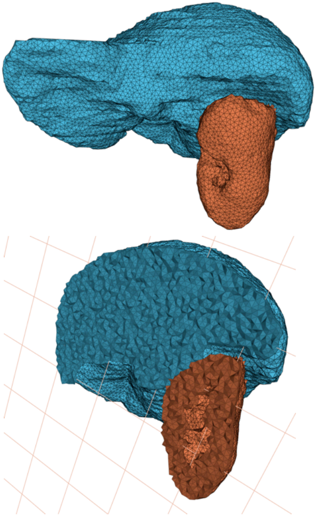
Mesh optimization



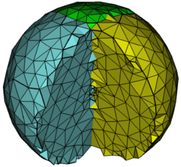
Original Mesh
(50k vertices, 290k tets, 10 seconds)

ODT smoothing
(global optimization, 110s)

ODT + Sliver perturbation
(local optimization, 40s)

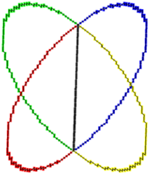


Handling sharp features

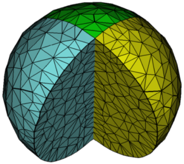


Sharp features are handled through
the method of protecting balls
[Cheng, Dey et al 07]

1. Cover sharp edges with protecting balls
   - balls cover the edges
   - balls do not include each other centers
   - two balls centered on different edges do not intersect
   - three balls do not intersect

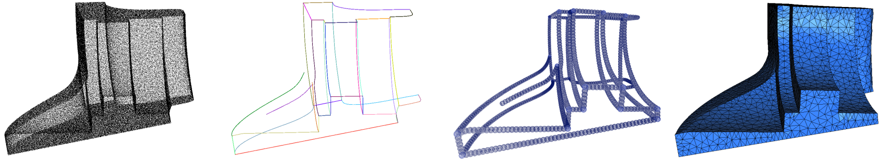


2. Use a weighted Delaunay triangulation
Initialize mesh with protecting balls
Run weighted version of Delaunay refinement



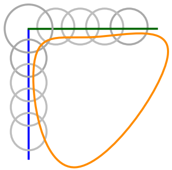
Segments joining centers of consecutive protecting balls are guaranteed to be edges in the mesh

Handling sharp features



Weights ensure:

- No refinement point inserted inside the balls
- Termination of Delaunay refinement granted whatever may be angles formed by sharp edges
- No queries on the surface part inside the ball

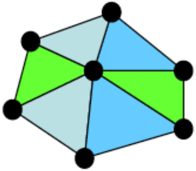


Remarks :

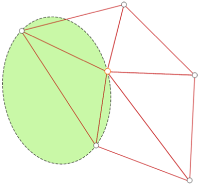
- Protecting balls influence mesh sizing
- Protecting balls should cover the gap between the surface and the sharp edges

Locally Uniform Anisotropic Delaunay Meshes

A mesh such that:
the star of each vertex is
Delaunay and well shaped
wrt the metric at that vertex.

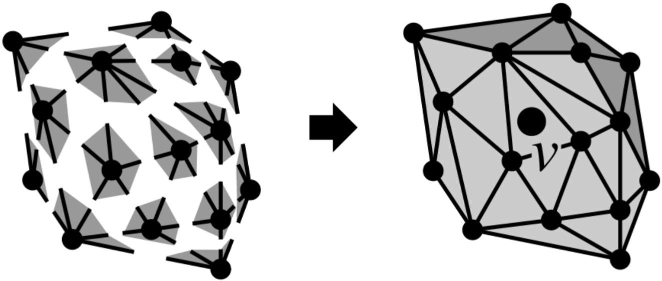


V set of vertices, $v \in V$:
M_v metric at v
$\mathrm{Del}_v(V)$ Delaunay triangulation of V
computed with metric M_v
S_v: the star of v in $\mathrm{Del}_v(V)$



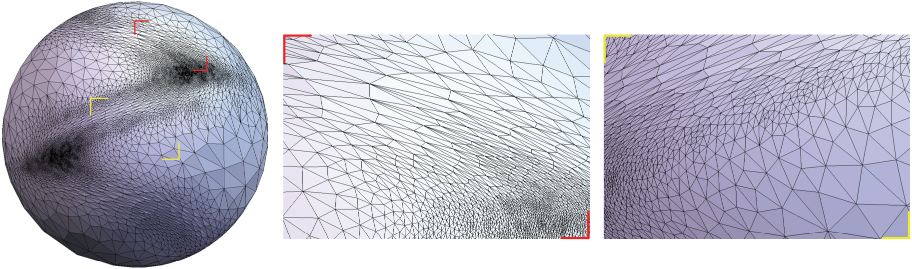
Overview of the meshing algorithm

- Maintain the set of stars $S(V) = \{S_v : v \in V\}$
- Refine V until stars are well shaped and consistent.
- Consistent stars are stitched



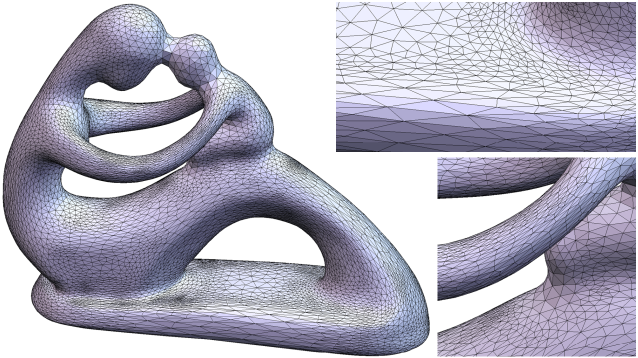
Anisotropic Surface Meshes

Metric according to a shock wave function

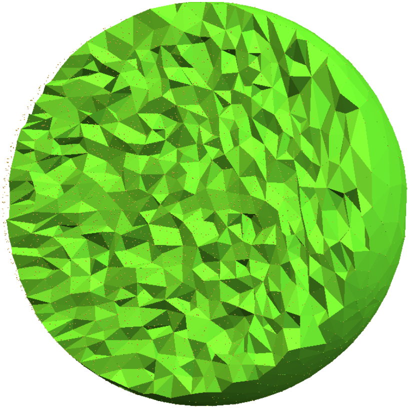


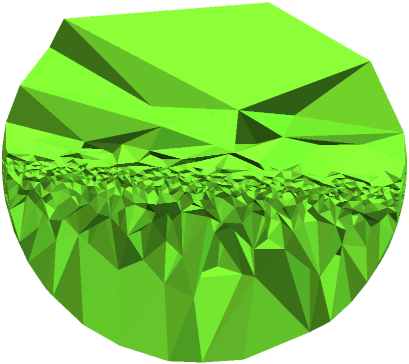
Anisotropic Surface Mesh

Estimated curvature tensor onpolyhedral surfaces



Anisotropic 3D Meshes





Anisotropic 3D Meshes