# The Computational Geometry Algorithms Library

**Andreas Fabri**
**GeometryFactory**

# Mission Statement

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

CGAL Project Proposal, 1996

# Algorithms and Datastructures

**Convex Hull Algorithms**
• Convex hull and extreme points (2D)
• Convex hull of points (3D)
• Convex hull and Delaunay triangulation (dD)
• Convex hull of circles

**Polygons and Polyhedra**
• Polygon (2D)
• Polygon partitioning (2D)
• Polyhedral surface
• Halfedge data structure

**Polygon and Polyhedron Operations**
• Regularized Boolean set-operations (2D)
• Boolean operations on Nef polygons (2D)
• Boolean operations on Nef polygons on the sphere
• Boolean operations on Nef polyhedra (3D)
• Straight skeleton and polygon offsetting (2D)

**Arrangements**
• Arrangements of arcs of lines and circles (2D)
• Intersection of curves (2D)
• Snap rounding (2D)

**Triangulations**
• Delaunay (2D,3D), Constrained Delaunay (2D)
• Regular triangulation (2D,3D)
• Triangulation data structure (2D,3D)
• Alpha shape (2D,3D)

**Voronoi Diagrams**
• Segment Delaunay graph (2D)
• Delaunay graph of disks (2D)
• Voronoi diagram adaptor (2D)

**Mesh Generation**
• Conforming triangulations and meshes (2D)
• Surface mesh generator (3D)
• Iso surface extraction from voxel data (3D)
• Surface subdivision methods (3D)
• Parameterization of triangulated surfaces

**Search Structures**
• Interval skip list
• Nearest/furthest neighbor search (dD)
• Range and segment trees (dD)
• Intersecting sequences of iso-oriented boxes (dD)

**Geometric Optimization**
• Polytope distance (3D)
• Minimum enclosing sphere/ellipsoid of points (dD)
• Minimum enclosing sphere of spheres (3D)
• Principal component analysis (2D,3D)

**Interpolation**
• Natural neighbor interpolation (2D)
• Placement of streamlines (2D)

**Kinetic Data Structures**
• Kinetic triangulations (2D,3D)
• Kinetic framework

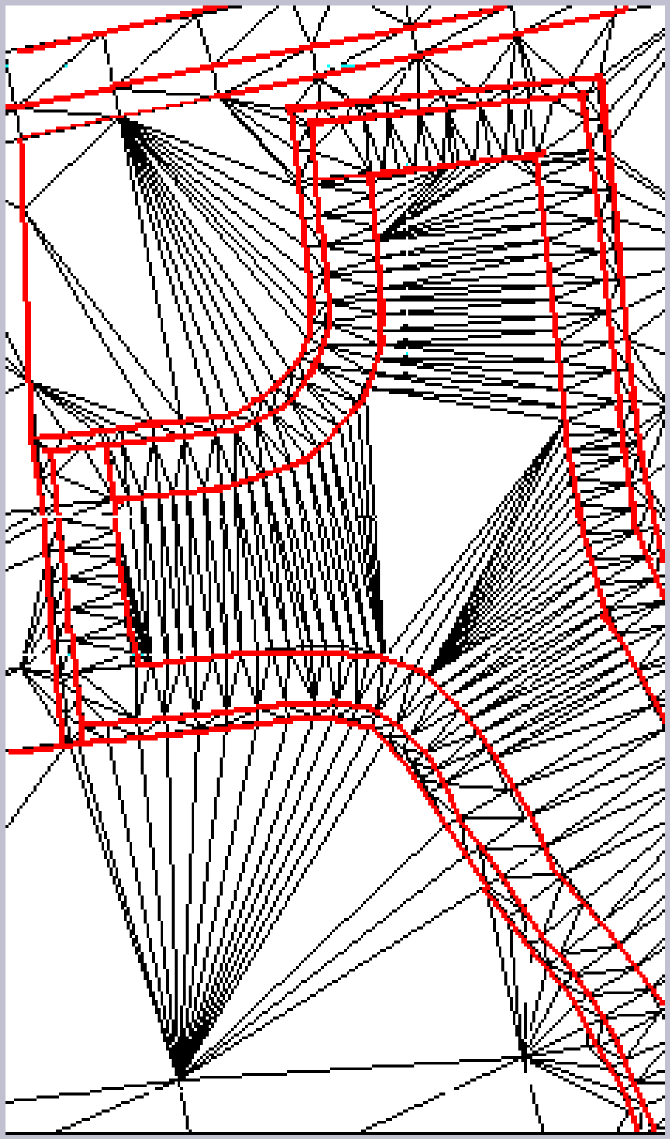
**Under Development**
• Mesh simplification (3D)
• Polyline simplification (2D)
• Volume mesh generator (3D)
• Minkowski sum (2D)
• Voronoi diagram of spheres (3D)
• Multiplicatively weighted Voronoi diagram (2D)
• Polyhedral surface reconstruction (3D)
• Ridge and crest line detection on a surface (3D)

# 2D Triangulations

- Delaunay and Regular
- Can handle constraints
- Can deal with elevation (terrains)
- Kinetic version

- Datastructures are fully dynamic
- 1 mio 2D points in 20 sec

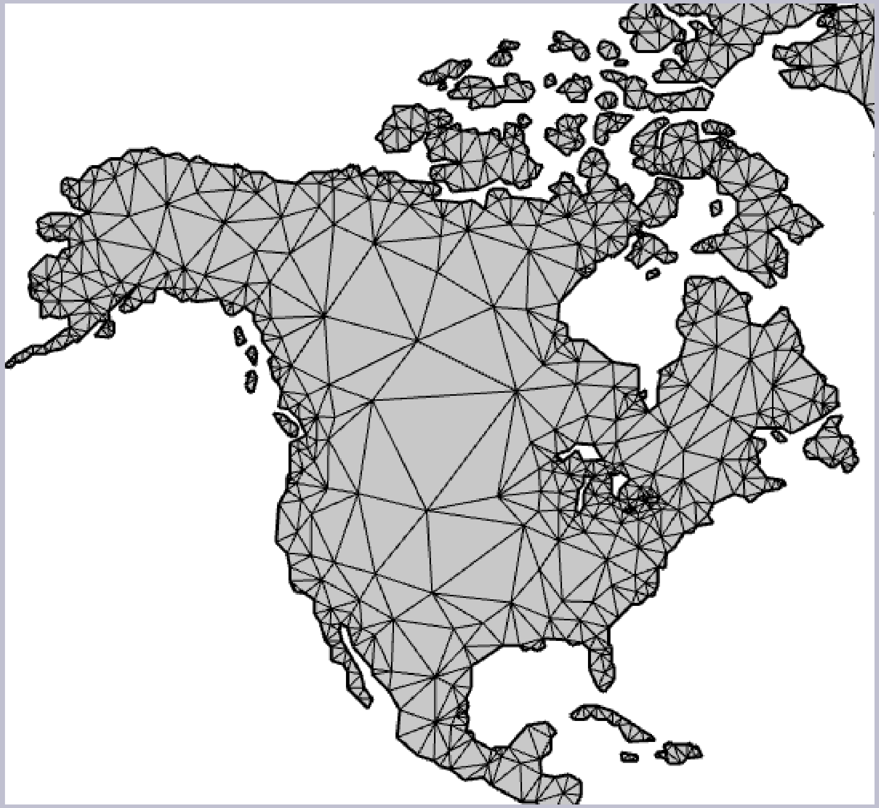
- Details in an afternoon session

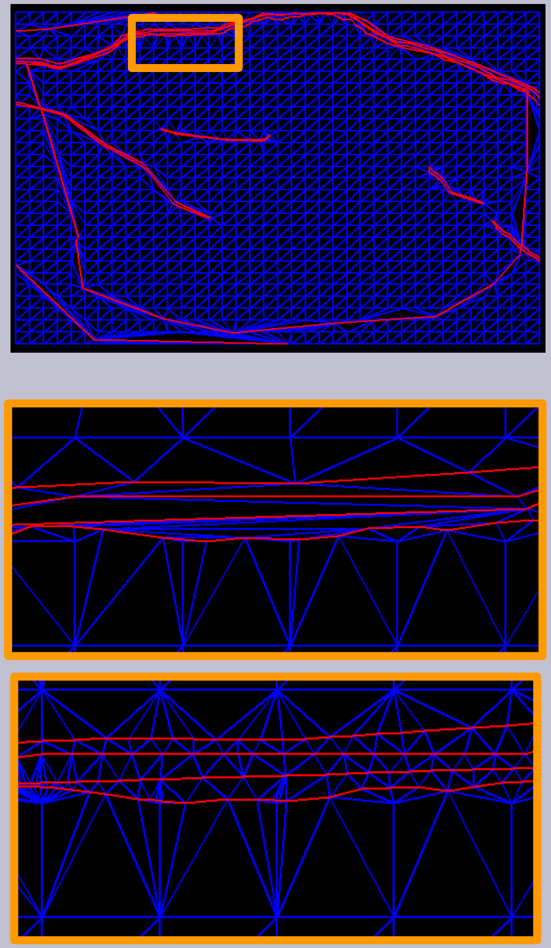


Courtesy: ECL

# 2D Mesh Generation

- Conforming triangulations
- Meshing the interior of polygons



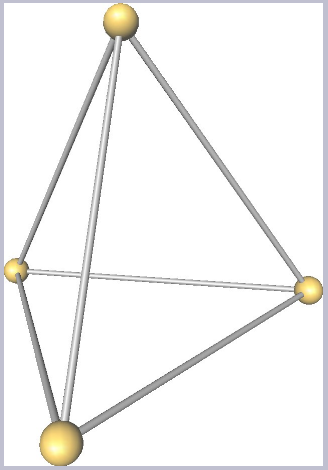


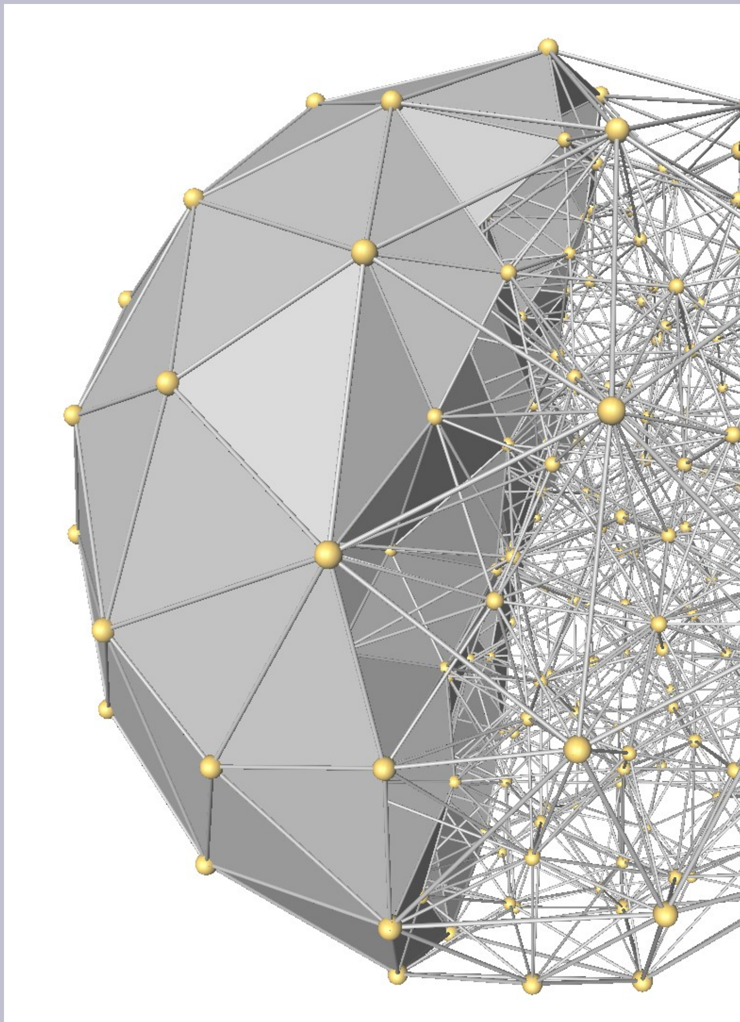
Courtesy: Agip

# 3D Triangulations

- Delaunay and Regular Triangulation
- Kinetic version

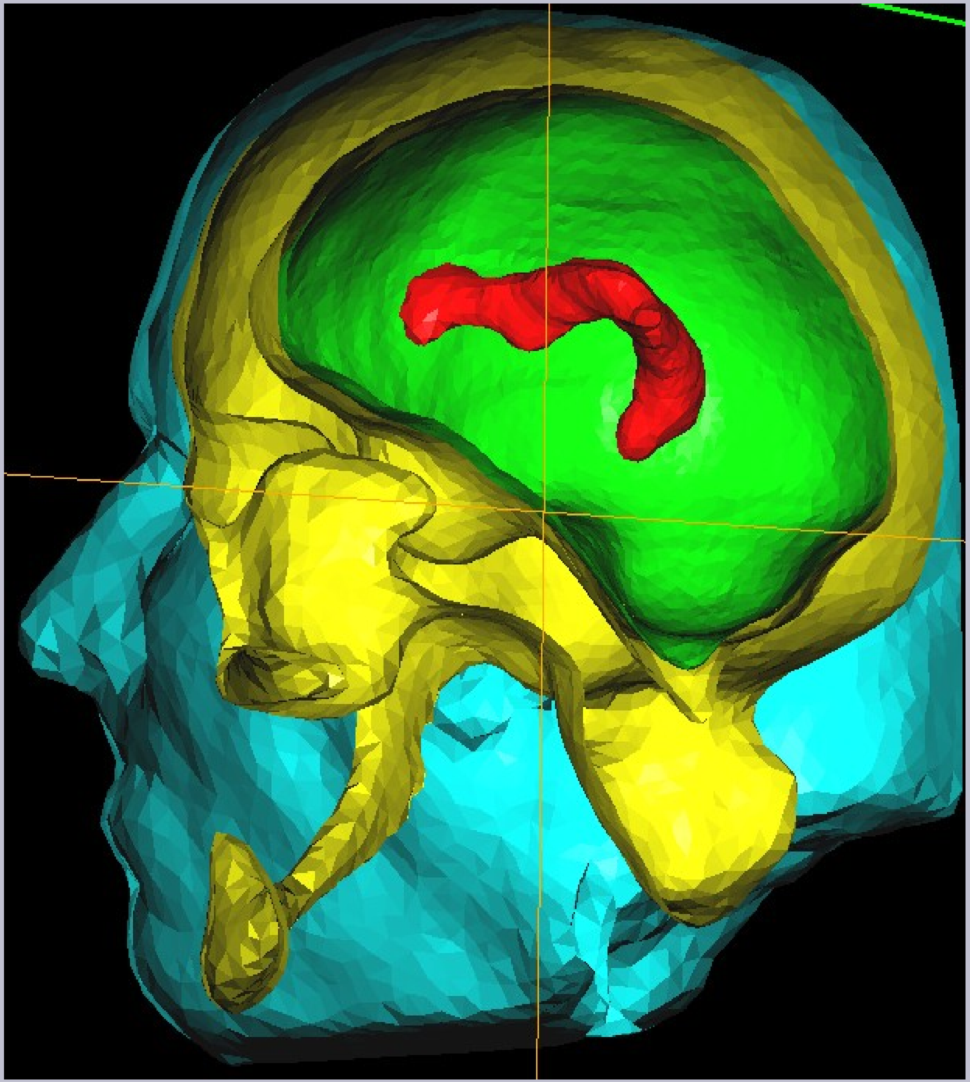
- fully dynamic datastructure
- 1 mio 3D points in 60 sec





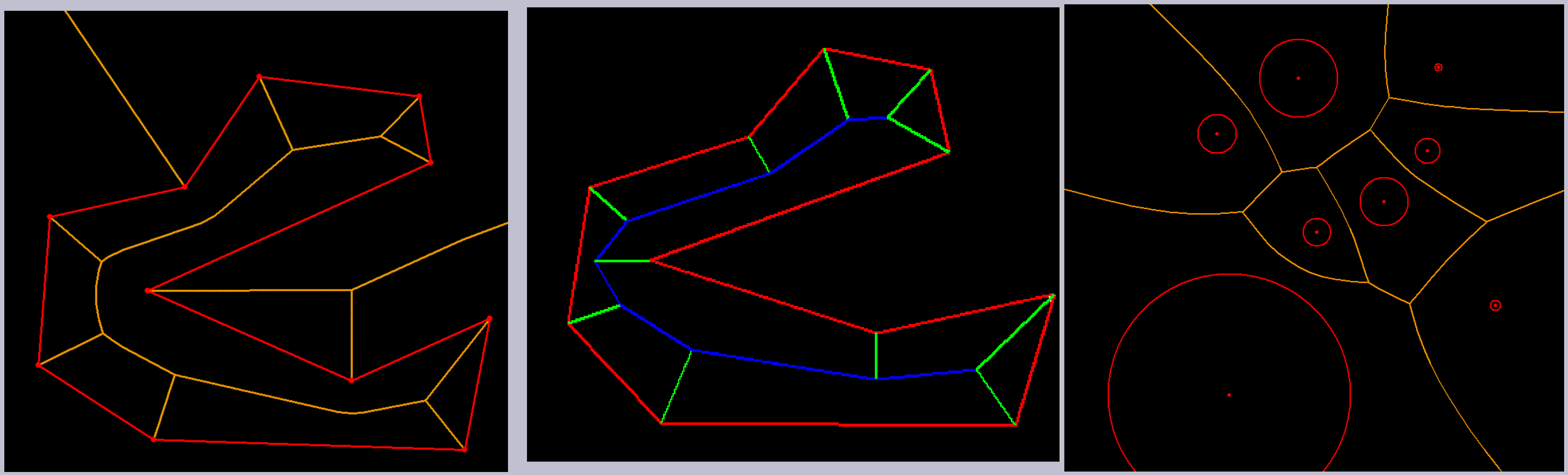
# 3D Mesh Generation

- Surface mesh
  - of implicit surface
  - from image
  - for polyhedral mesh*
- Volume mesh*



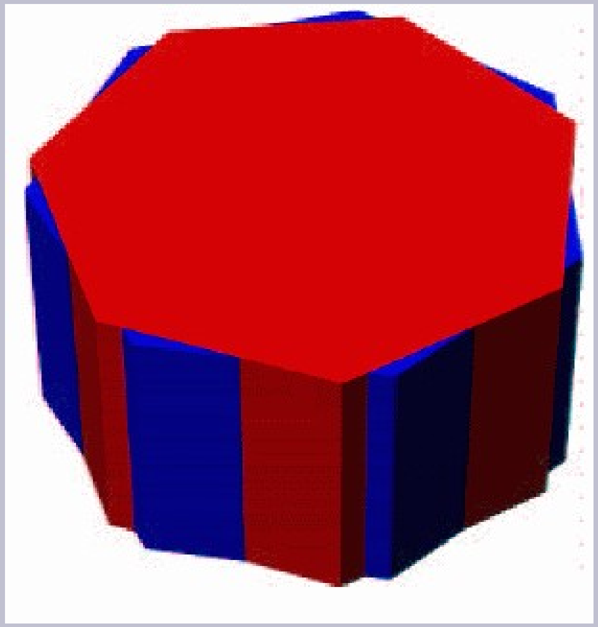
# Voronoi Diagrams

- 2D, 3D, dD for points and weighted points
- 2D for segments and polygons
- 2D Straight Skeleton
- 2D for circles



# 2D and 3D Boolean Operations

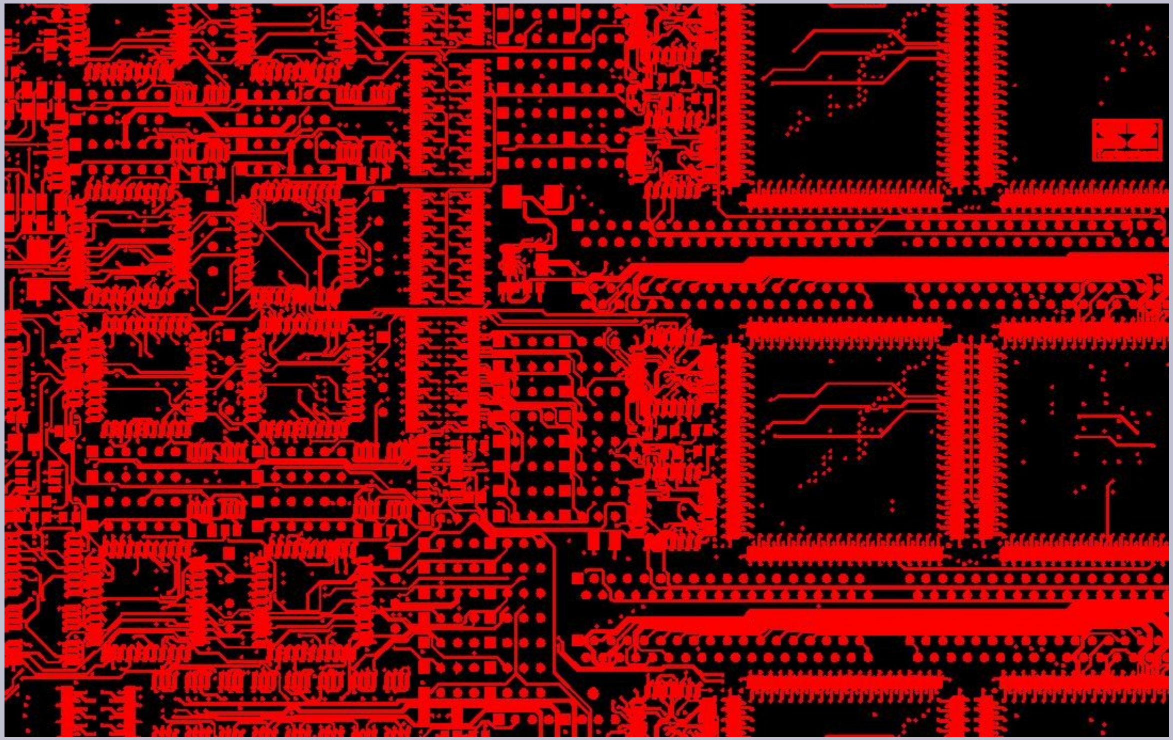
Operation union (n-gon, rotate(n-gon, alpha))



| n | $\alpha$ | time Acis R13 | runtime [s] Nef 3D |
|---|---|---|---|
| 100 | $10^{-1}$ | 1.08s | 3.47s |
| | $10^{-2}$ | 1.05s | 3.50s |
| | $10^{-3}$ | 1.08s | 3.59s |
| | $10^{-4}$ | 1.07s | 3.64s |
| | $10^{-5}$ | not executable | 3.72s |
| | $10^{-6}$ | not executable | 3.77s |
| 1000 | $10^{-1}$ | 61s | 67s |
| | $10^{-2}$ | 61s | 68s |
| | $10^{-3}$ | 61s | 69s |
| | $10^{-4}$ | not executable | 69s |
| | $10^{-5}$ | not executable | 71s |
| | $10^{-6}$ | not executable | 71s |
| 2000 | $10^{-1}$ | 252s | 195s |
| | $10^{-2}$ | 253s | 198s |
| | $10^{-3}$ | 255s | 203s |
| | $10^{-4}$ | not executable | 205s |
| | $10^{-5}$ | not executable | 207s |
| | $10^{-6}$ | not executable | 210s |
| 10000 | $10^{-7}$ | not executable | 3219 s |

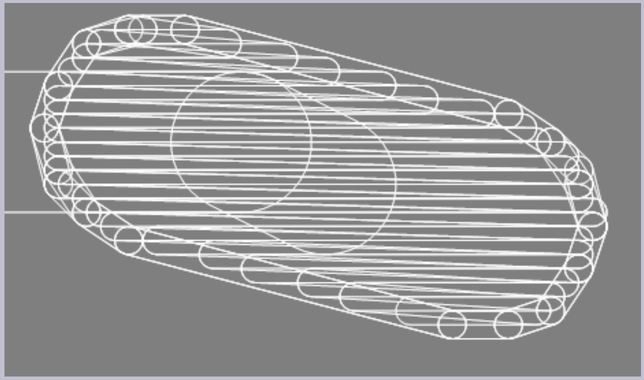
# Boolean Operations on Circular Arcs

- Based on arrangements of 2D curves
- Without discretization

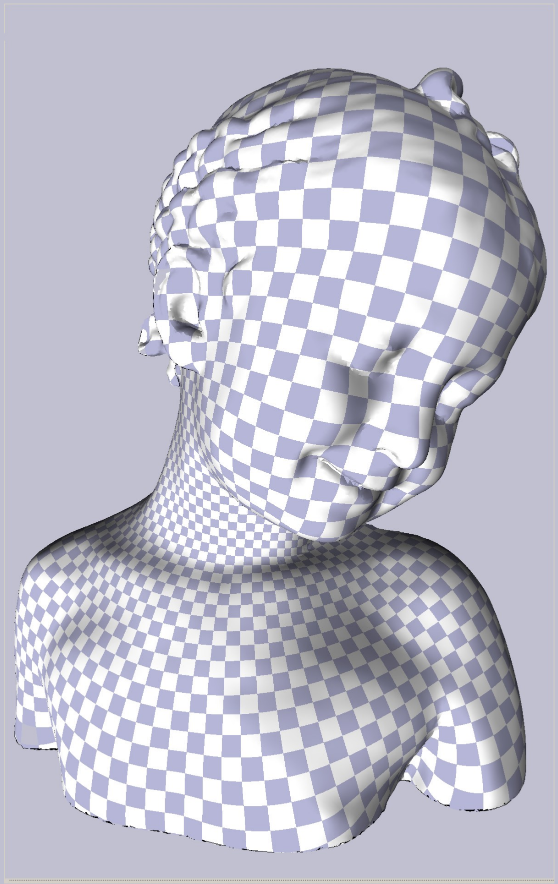


Courtesy: Maniabarco

Union of polygons consisting of segment and circular arcs



# Polyhedral Surface Meshes

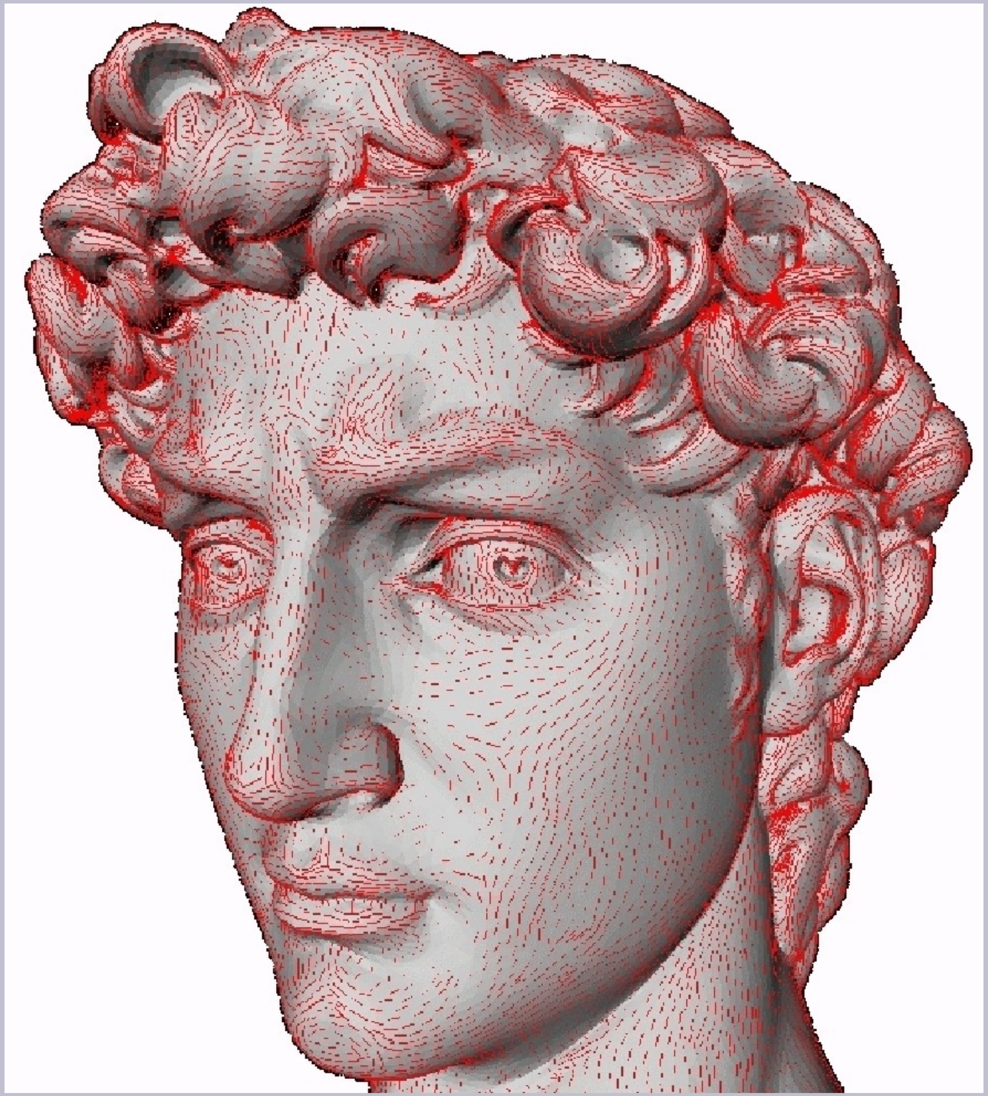


Parametrization

Simplification

# Differential Quantities

- Estimation of Differential Quantities
- Extraction of Ridges and Crest Lines



# Distinguishing Features of CGAL

- Focus on geometry
- Interfaces with de facto standards/leaders:
  STL, boost, GMP, Qt, blas
- Ease of integration through generic programming
- Ease of customization
- Main stream language C++
- Huge collection with uniform APIs
- Modular and not monolithic
- Epsilon-free

# CGAL Users

# Main Motivation for Using CGAL

“ I recommended to the senior management that we start a policy of buying-in as much functionality as possible to reduce the quantity of code that our development team would have to maintain.

This means that we can concentrate on the application layer and concentrate on our own problem domain.”

Senior Development Engineer
& Structural Geologist

Midland Valley Exploration

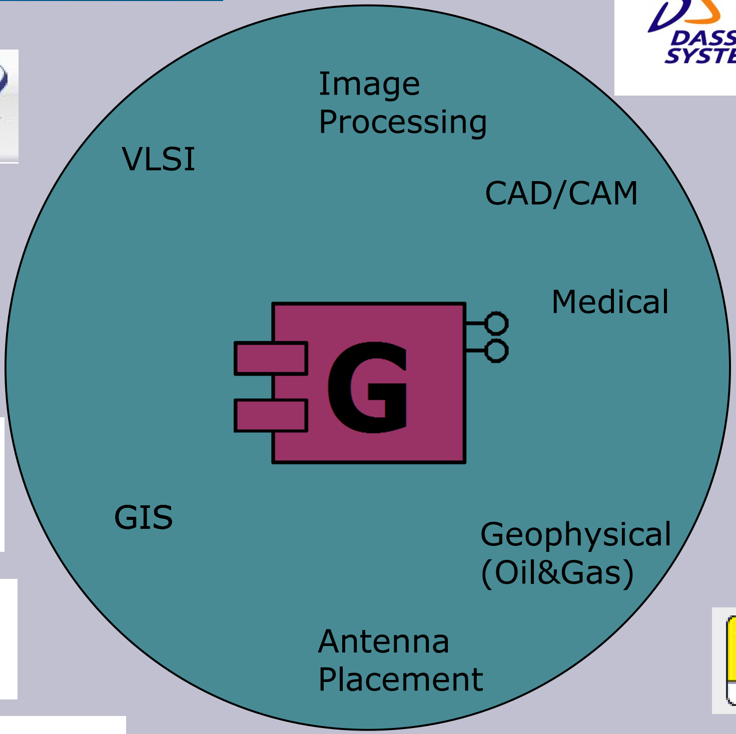
# Main Motivation for Using CGAL

" My research group JYAMITI at the Ohio State University uses CGAL because it provides an efficient and robust code for Delaunay triangulations and other primitive geometric predicates. Delaunay triangulation is the building block for many of the shape related computations that we do. [...]

Without the robust and efficient codes of CGAL, these codes could not have been developed. "

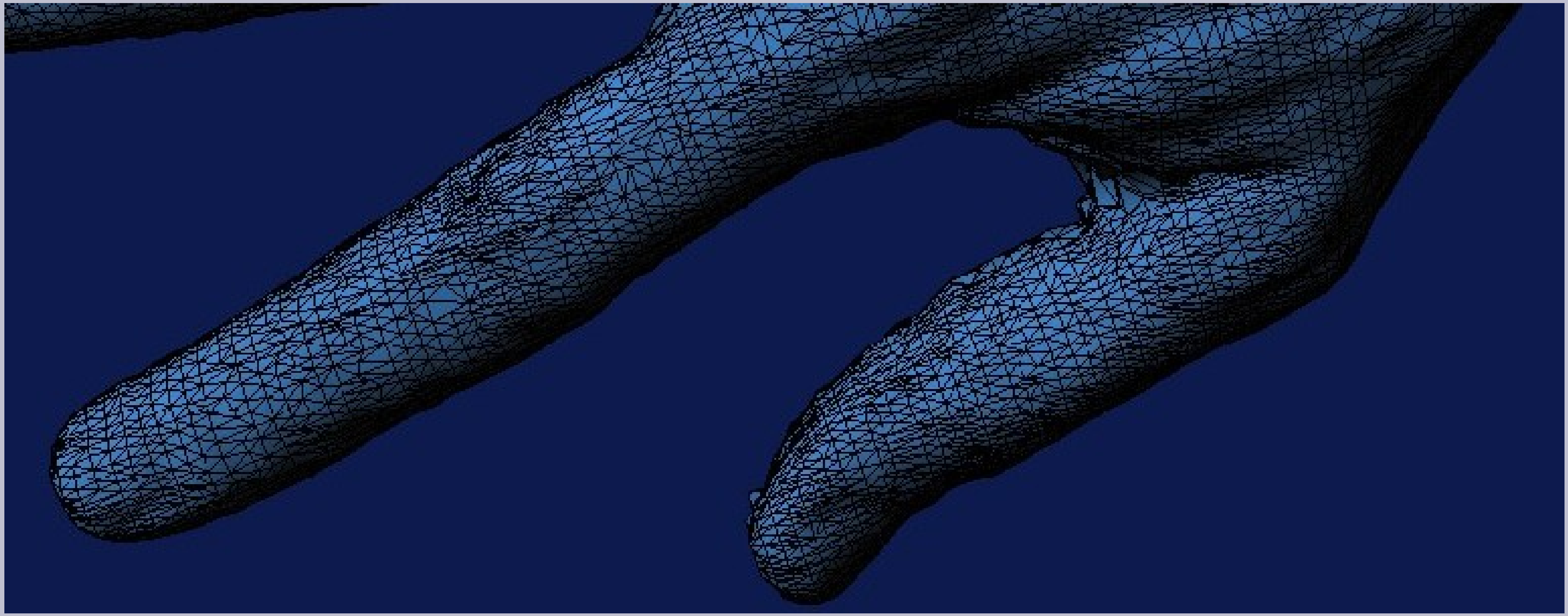
Tamal Dey
Professor, Ohio State University

# Some Companies using CGAL



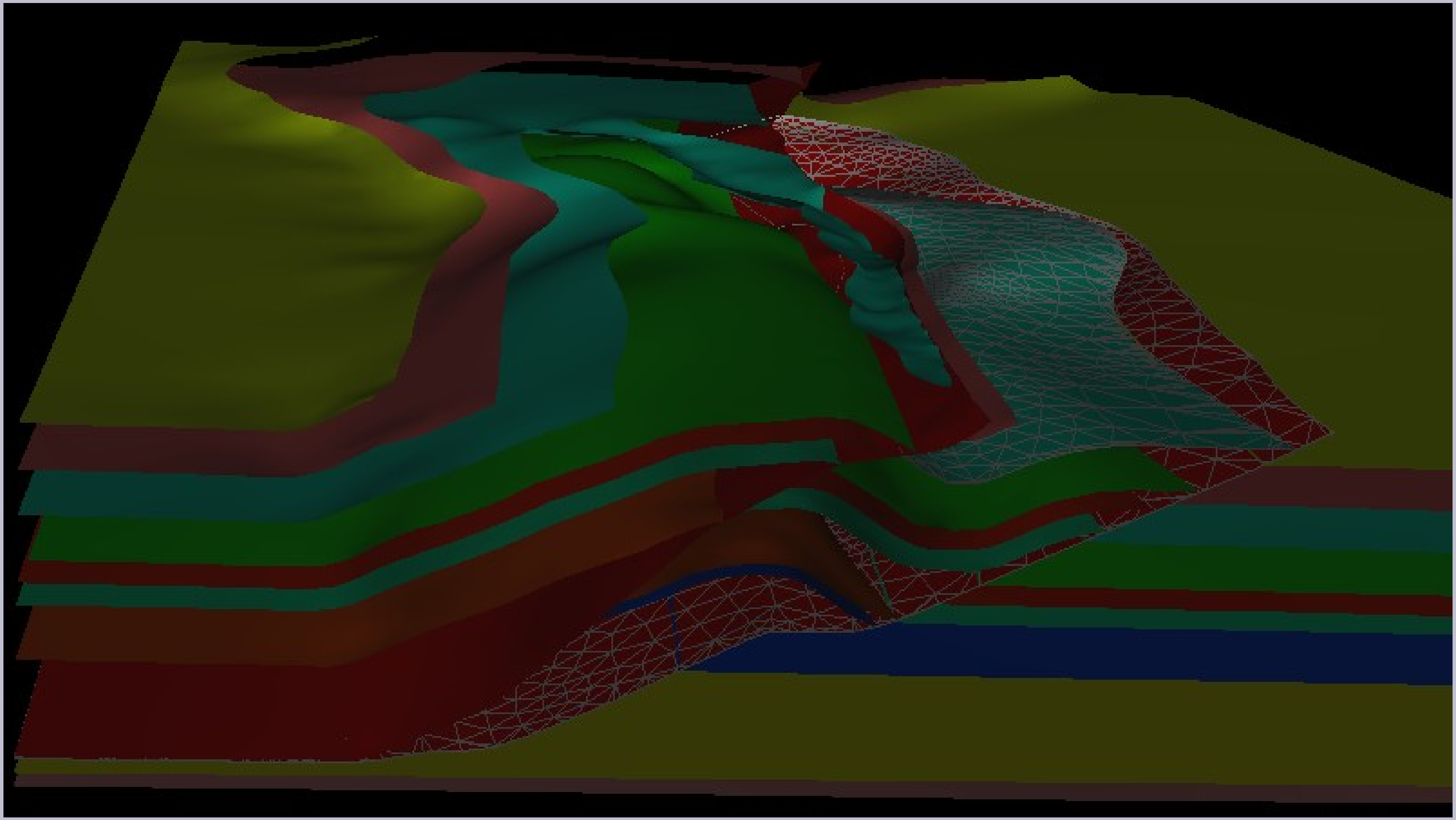
# Surface Reconstruction

- for point clouds
- for parallel slices with polygonal contours



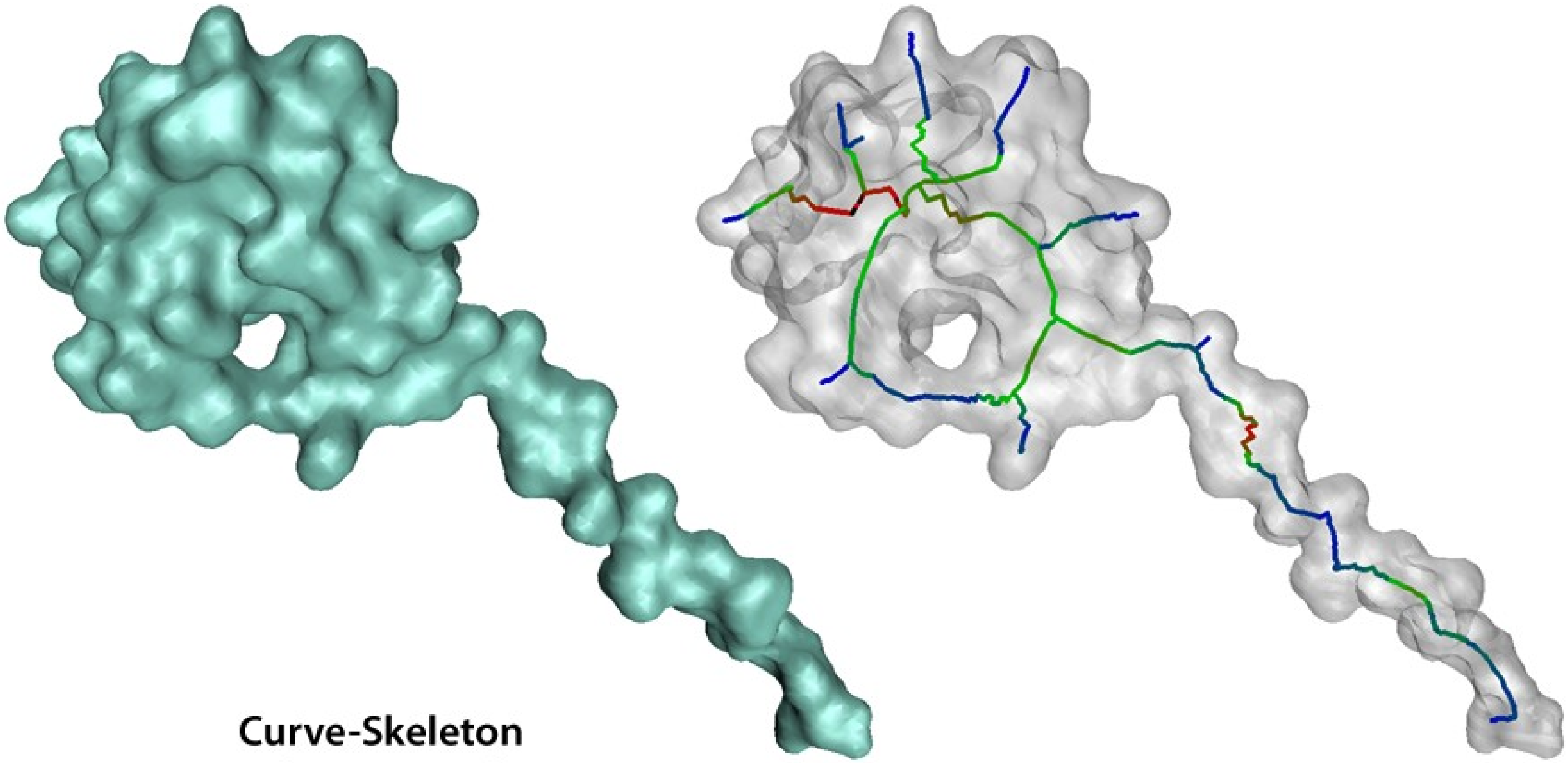
Courtesy: Device Works

# Modeling of Geological Structures



Courtesy: Midland Valley Exloration

# Curve Skeleton



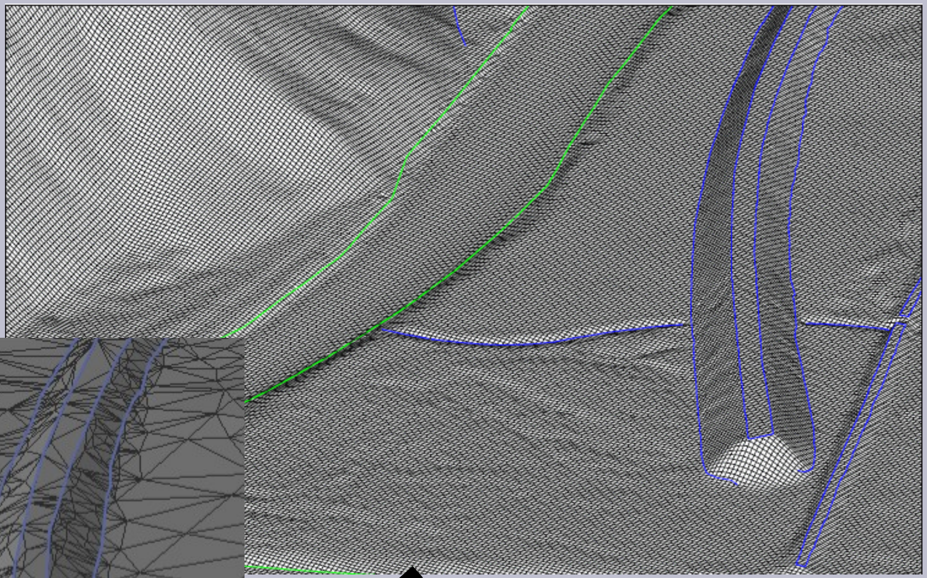
**Curve-Skeleton**
Left: Input surface.
Right: Output Curve-Skeleton, colored by eccentricity values.

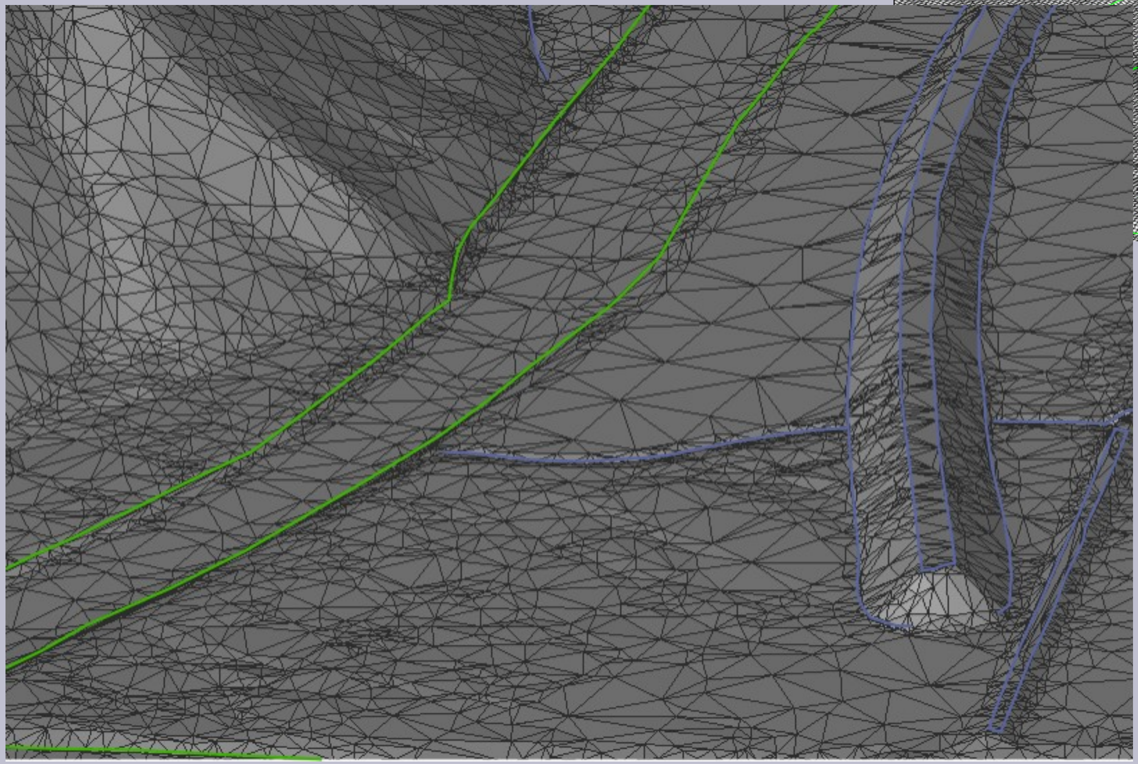
Courtesy: Ohio State University

# Remeshing of Terrains

Photogrammetry software

- Grid + breaklines -> TIN

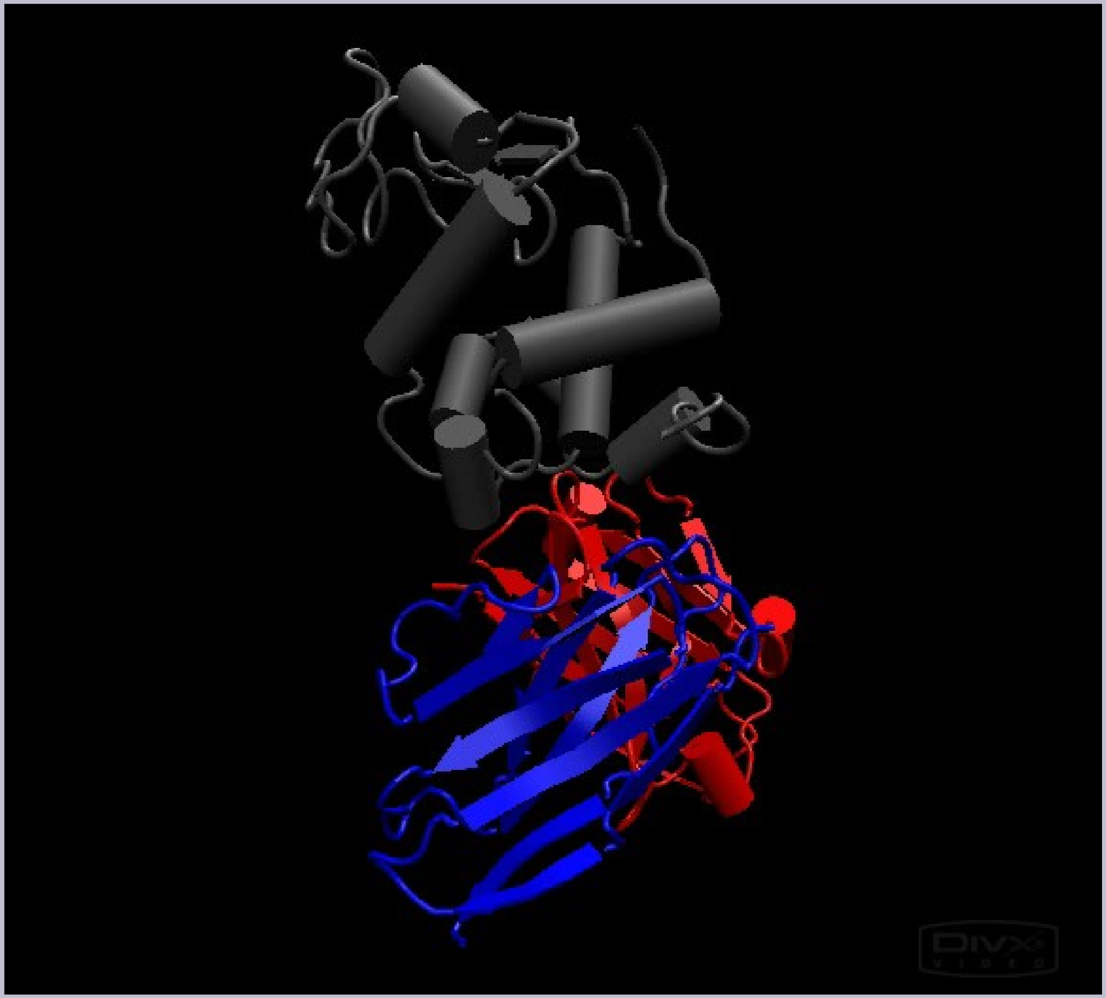




Courtesy: Inpho/TU Vienna

# Exploring Protein-Protein Interfaces

Arrangements of circles on sphere



Courtesy: Cazals,Loriot/ INRIA