Mesh Processing: Theory and Applications

Jean-Marie Favreau

UdA, LIMOS (UMR 6158)

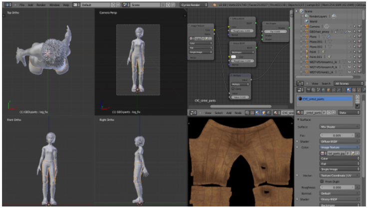
May–June 2016

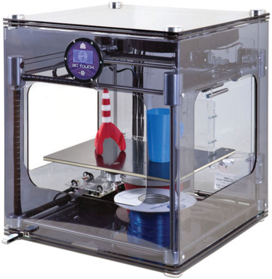
Slides available online: `http://jmfavreau.info/spi6/`

Introduction

Motivations

Available data

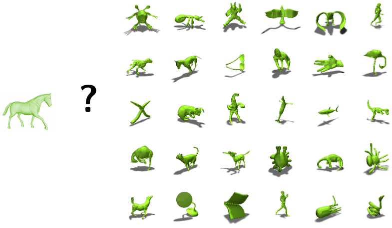


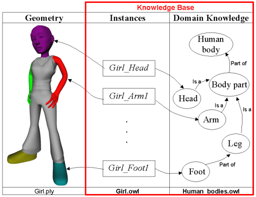


Motivations

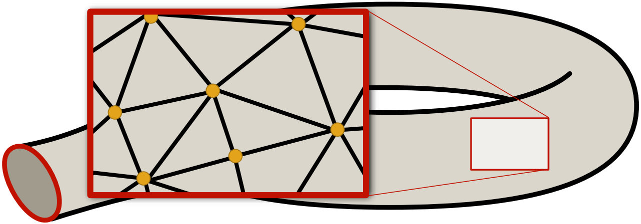
Required tasks

- Retrieval in databases
- Shape recognition
- Semantic extraction
- Shape deformation
- Augmented reality
- ...





Triangular Mesh



Vertices, edges, triangles.

Data Structures

A mesh in a file

```
OFF
#   cube.off
#   A cube

8 6 12
 1.0    0.0    1.0
 0.0    1.0    1.0
-1.0    0.0    1.0
 0.0   -1.0    1.0
 1.0    0.0   -1.0
 0.0    1.0   -1.0
-1.0    0.0   -1.0
 0.0   -1.0   -1.0
 4   0 1 2 3
 4   7 4 0 3
 4   4 5 1 0
 4   5 6 2 1
 4   3 2 6 7
 4   6 5 4 7
```

File formats

ascii (off, obj, ply, ...), binary (wrl, ...), application specific (blend)

Available data

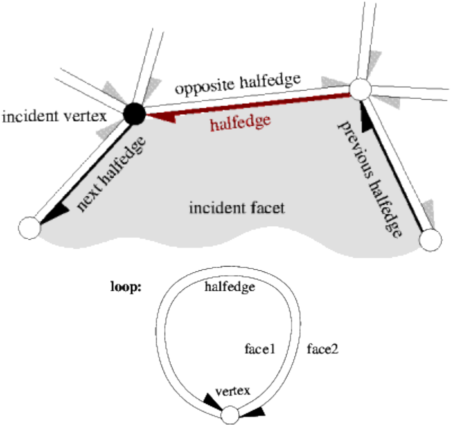
- A list of vertices (3d coordinates)
- A list of facets (IDs of the vertices)
- Supplementary information for each vertex/facet (color, texture, etc.)
- Possibly edges, textures, etc.

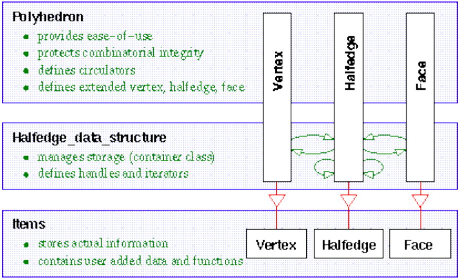
Naive (and matlab) data structures

The list of 3D coordinates, and the list of facets.

Data Structures

Halfedge based approach (CGAL, OpenMesh)

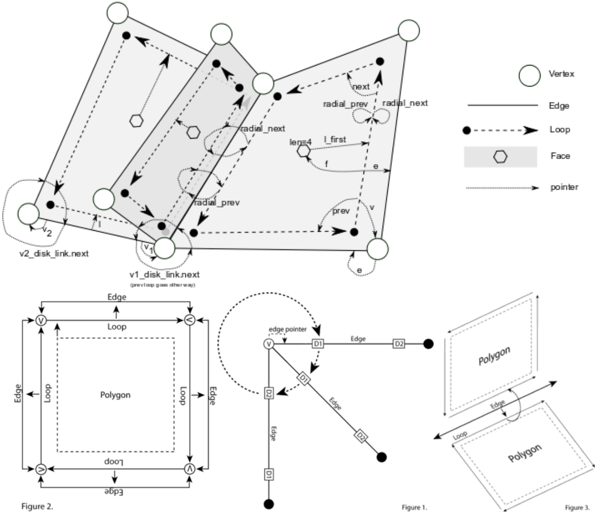




Circulators, iterators

Data Structures

Blender data structure

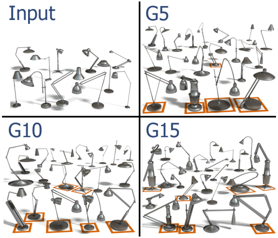


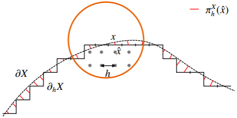
Go back to the manifold definition

Many scientific communities

Scientific Fields

- Computer vision
- Geometry modelling
- Computer graphics
- Computational geometry
- Computational topology
- Discrete geometry
- Digital geometry
- Medical imaging





Lemma 7 *The singular normal surface in a tube specified by the gluing is immersed if and only if both variables of the tube are equal.*

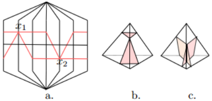
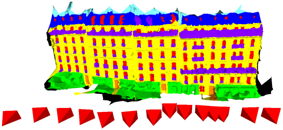
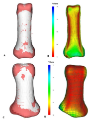


Figure 5: a. A tube gadget. b.c. The constant gadgets CG_0 and CG_1.

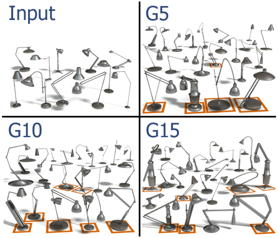


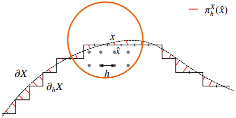


Many scientific communities

Sc. Fields: "End users"

- **Computer vision**
- Geometry modelling
- Computer graphics
- Computational geometry
- Computational topology
- Discrete geometry
- Digital geometry
- **Medical imaging**





Lemma 7 *The singular normal surface in a tube specified by the gluing is immersed if and only if both variables of the tube are equal.*

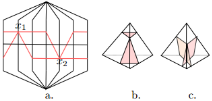
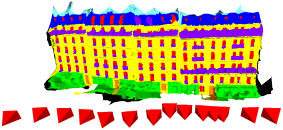
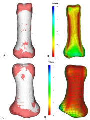


Figure 5: a. A tube gadget. b.c. The constant gadgets CG_0 and CG_1.

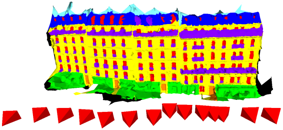


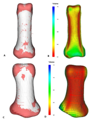


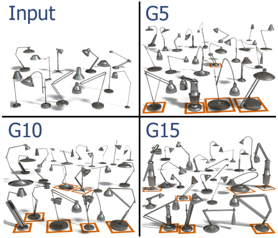
Many scientific communities

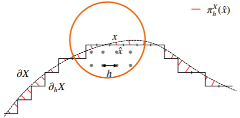
Sc. Fields: Model fitting

- **Computer vision**
- Geometry modelling
- Computer graphics
- Computational geometry
- Computational topology
- Discrete geometry
- Digital geometry
- Medical imaging









Lemma 7 *The singular normal surface in a tube specified by the gluing is immersed if and only if both variables of the tube are equal.*

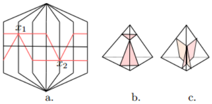
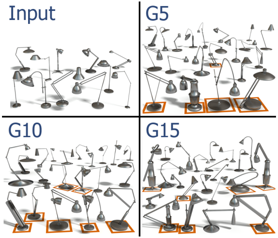


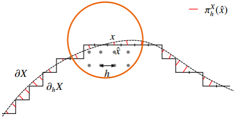
Figure 5: a. A tube gadget. b.c. The constant gadgets CG_0 and CG_1.

Many scientific communities

Sc. Fields: Fast & eye candy

- **Computer vision**
- **Geometry modelling**
- **Computer graphics**
- Computational geometry
- Computational topology
- Discrete geometry
- Digital geometry
- Medical imaging





Lemma 7 *The singular normal surface in a tube specified by the gluing is immersed if and only if both variables of the tube are equal.*

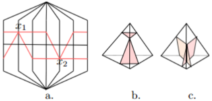
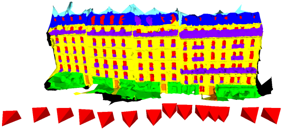
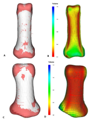


Figure 5: a. A tube gadget. b.c. The constant gadgets CG_0 and CG_1.

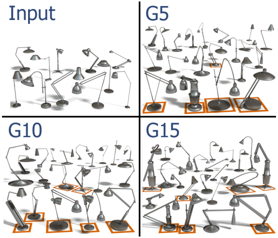


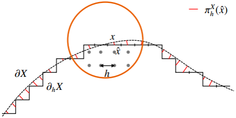


Many scientific communities

Sc. Fields: Mathematics

- Computer vision
- Geometry modelling
- Computer graphics
- **Computational geometry**
- **Computational topology**
- **Discrete geometry**
- **Digital geometry**
- Medical imaging





Lemma 7 *The singular normal surface in a tube specified by the gluing is immersed if and only if both variables of the tube are equal.*

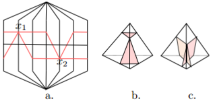
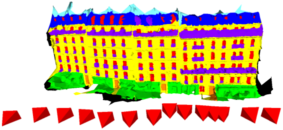
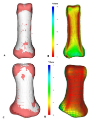


Figure 5: a. A tube gadget. b.c. The constant gadgets CG_0 and CG_1.

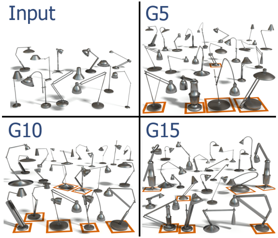


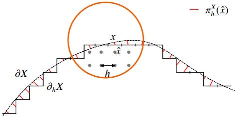


Many scientific communities

Sc. Fields: Algebra

- Computer vision
- Geometry modelling
- Computer graphics
- **Computational geometry**
- **Computational topology**
- Discrete geometry
- **Digital geometry**
- Medical imaging





Lemma 7 *The singular normal surface in a tube specified by the gluing is immersed if and only if both variables of the tube are equal.*

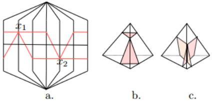
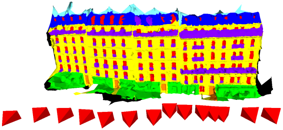
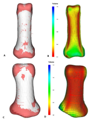


Figure 5: a. A tube gadget. b.c. The constant gadgets CG_0 and CG_1.

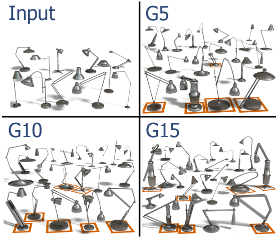


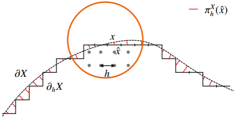


Many scientific communities

Sc. Fields: Algorithmics

- Computer vision
- Geometry modelling
- Computer graphics
- **Computational geometry**
- **Computational topology**
- **Discrete geometry**
- **Digital geometry**
- Medical imaging





Lemma 7 *The singular normal surface in a tube specified by the gluing is immersed if and only if both variables of the tube are equal.*

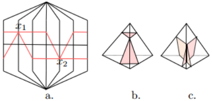
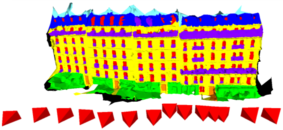
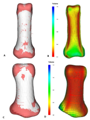


Figure 5: a. A tube gadget. b.c. The constant gadgets CG_0 and CG_1.

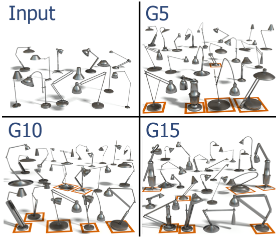


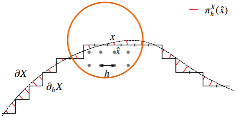


Many scientific communities

Sc. Focus of this lecture

- Computer vision
- Geometry modelling
- **Computer graphics**
- **Computational geometry**
- **Computational topology**
- **Discrete geometry**
- **Digital geometry**
- *Medical imaging*





Lemma 7 *The singular normal surface in a tube specified by the gluing is immersed if and only if both variables of the tube are equal.*

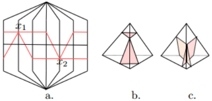
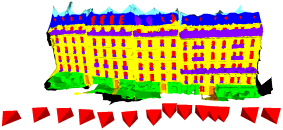
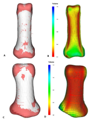


Figure 5: a. A tube gadget. b.c. The constant gadgets CG_0 and CG_1.





Mesh Analysis

Mesh Analysis – ingredients

Measure

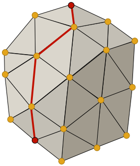
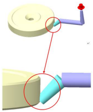
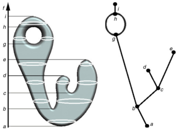
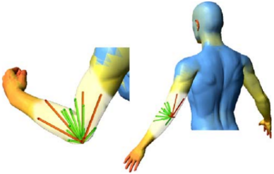
Distances, areas, diameters, etc.

Understand

Relative position, extremities, excentricity, shape, primitives, etc.

Extract abstract description

Global structure, extrema, squeleton, etc.

Mesh Analysis

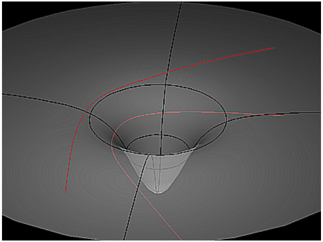
Measure on Surfaces

Geodesic distance

Definition

Geodesic path

A **geodesic** is a shortest path between two points of the surface.



Alternative definition: on a Riemannian space, the parallel transport along the curve preserves the tangent vector to the curve.

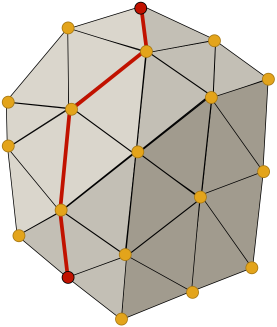
Geodesic distance

The **geodesic distance** between two points is *the length of a corresponding geodesic*.

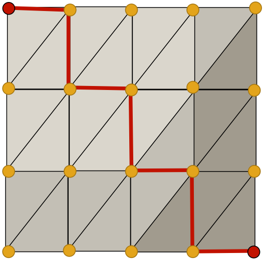
Geodesic distance

First approach using Dijkstra

First approximation: shortest path along edges.



estimation ≈ geodesic distance

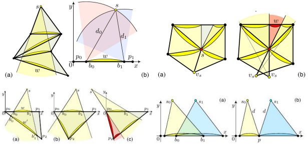


estimation $\approx \sqrt{2} \times$ geodesic distance
Manhattan distance

Dijkstra algorithm

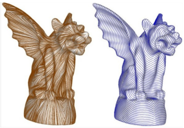
Geodesic distance

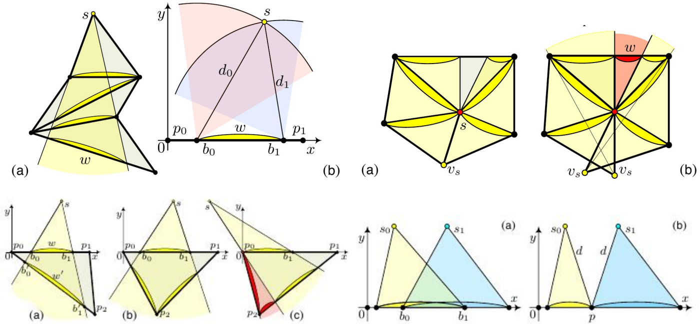
Exact and approximate computation (Surazhsky 2005)



Geodesic distance

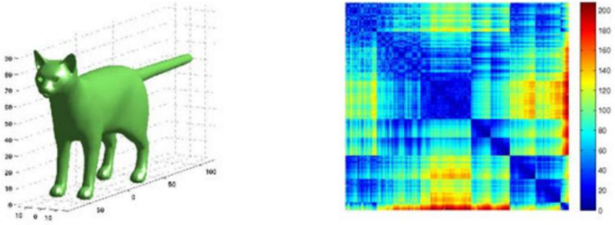
Exact and approximate computation (Surazhsky 2005)





Not easy to implement. Memory complexity of the exact algorithm: $O(n^3)$.

Application: Mesh Signature[1]



Geodesic distance matrix

[1]Smeets, D., Fabry, T., Hermans, J., Vandermeulen, D., & Suetens, P. (2010). Inelastic deformation invariant modal representation for non-rigid 3D object recognition. In Articulated Motion and Deformable Objects (pp. 162-171). Springer Berlin Heidelberg.

Mesh Analysis

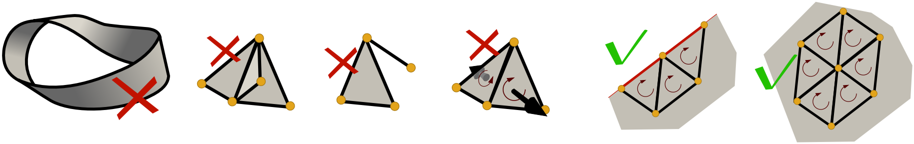
Topology

2-manifold (with boundary)

Definition, intuition

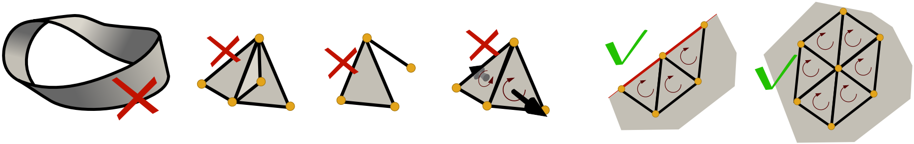
Surfacic simplicial complex (2-mesh)

A simplicial complex is said to be *an oriented surface* if it's of dimension 2, if the neigborhood of each 0-simplex is equivalent to a disc or an half-disc, and if the orientation of the faces are coherent.



2-manifold (with boundary)

Inspection, implementation



Inspect the mesh

- **Vertices:** check the coherency of the neigborhood
- **Edges:** check the number of adjacent faces
- **Faces:** check the orientation of the adjacent faces
- **Mesh:** check if the manifold is orientable

See data structures

Connected components

Definition, intuition

Connected components

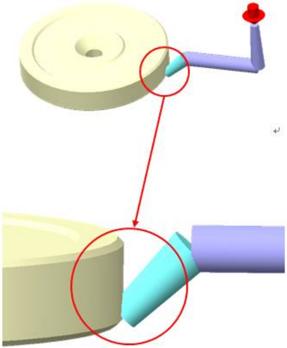
A **connected component** in a graph is a subgraph in which two vertices are connected to each other by paths.

Not to be confused with...

Intersections, self-intersections, embedding...

Number of connected components

▸ Implementation?

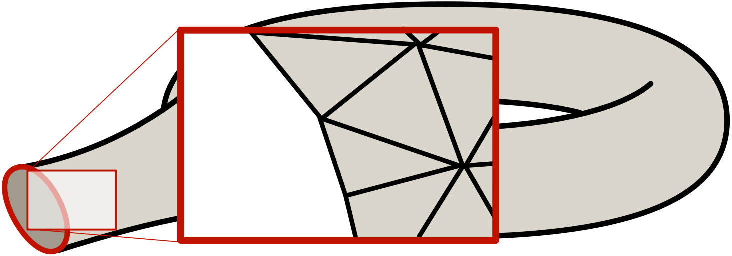


Borders

Definition, intuition

Border

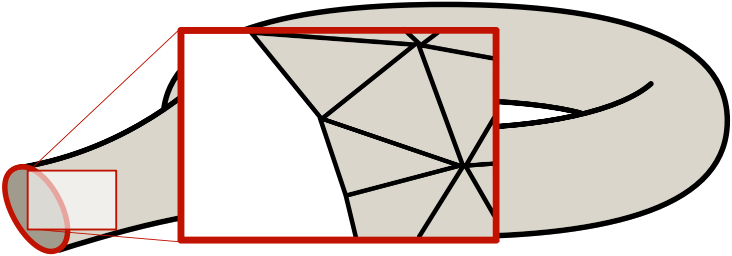
A border of an oriented surface is defined by a loop of boundary edges.



Number of borders

Borders

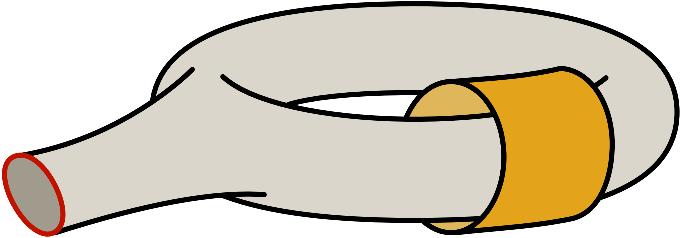
Inspection, implementation



Question: How to count the number of boundaries?

Genus

Definition, intuition



Genus: number of “handles”

Genus

Effective computation

Euler caracteristic

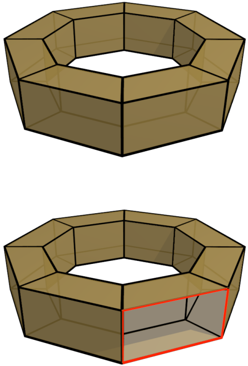
$$\chi = \#\textit{vertices} - \#\textit{edges} + \#\textit{faces}$$

Genus (for surfaces without boundary)

$$g = \frac{2-\chi}{2}$$

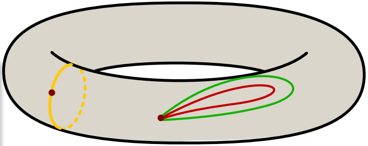
Genus (for surfaces with boundary)

?



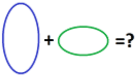
Fundamental group (first homotopy group)

Definition, intuition



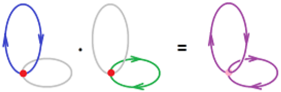
Loop, base point

Let $p \in \mathcal{M}$ be a base point. A loop $\gamma \in \mathcal{L}(\mathcal{M}, p)$ on $\mathcal{M}$ is a continuous application $\gamma : [0;1] \rightarrow \mathcal{M}$ such that $\gamma(0) = \gamma(1) = p$.



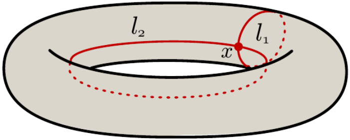
Homotopy

Two loops based on p γ_0, γ_1 are homotopic ($\gamma_0 \sim \gamma_1$) iif $\exists \delta : [0;1] \times [0;1] \rightarrow E$ such that: $\delta(\cdot, 0) = \gamma_0(\cdot)$, $\delta(\cdot, 1) = \gamma_1(\cdot)$, $\delta(0, \cdot) = \delta(1, \cdot) = p$.



Fundamental group

$\mathcal{L}(\mathcal{M}, p)$ quotient by $\sim$ is the fundamental group.

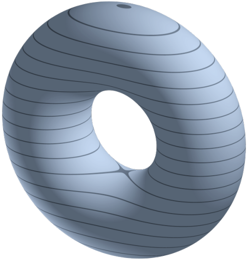


Other ingredients: contractile loop, separating loop...
Questions: dimension of the group?

Reeb Graph

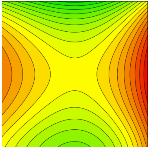
Ingredients

Requirement: $\mathcal{M}$ is a differentiable manifold



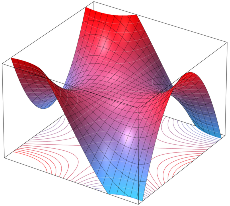
Smooth scalar function, contour lines

Let $f : \mathcal{M} \to \mathbb{R}$ be a *smoothed scalar function* on the mesh. A *contour line* (or *level set*) is the inverse image of a point in $\mathbb{R}$.



Critical points (maxima, saddle points)

p critical point $\Leftrightarrow$ gradient of f in p is 0.



Non degenerated (non singular) critical point

Critical point p not generated $\Leftrightarrow$ number of decreasing directions ≤ 2.

Reeb Graph

Morse function

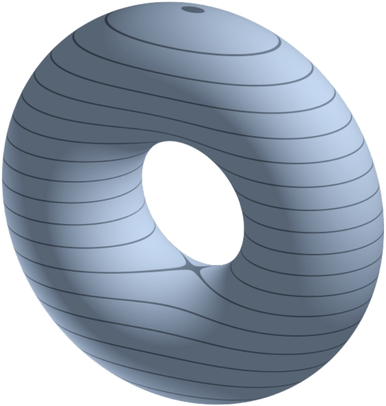
Morse function

$f : \mathcal{M} \rightarrow \mathbb{R}$ is a Morse function $\Leftrightarrow$ smooth function with no degenerated critical points.

We define $\mathcal{M}^a = f^{-1}(]-\infty; a])$.

Topological changes

Let $a < b$. If $f^{-1}([a; b])$ compact with no critical point between a and b, then $\mathcal{M}^b$ deformation retracts onto $\mathcal{M}^a$.



Reeb Graph

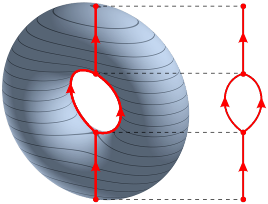
Definition, intuition

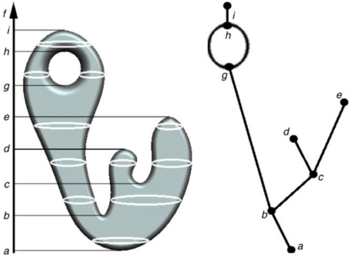
Equivalent relation

We define an equivalent relation $\sim$ such that $p \sim p \Leftrightarrow p$ and q belong to the same connected component for a single level set $f^{-1}(c)$ for some real c.

Reeb graph

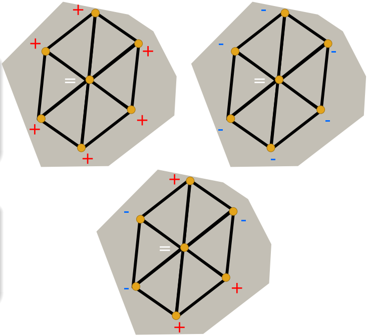
The *Reeb graph* is the quotient space $\mathcal{M}/\sim$.





Reeb Graph on meshes

Some ideas on the computation



Function definition

Scalar function only defined in vertices (+linear interpolation)

Avoid constant functions

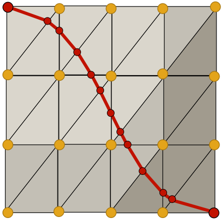
Add a different epsilon on each $f(p)$ to insure the total ordrer

Identify critical points

Compare local values in the 1-ring

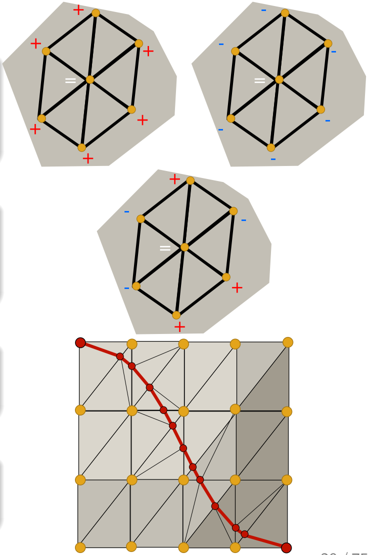
Level sets

Compute piecwise-linear paths throw triangles



Reeb Graph on meshes

Some ideas on the computation



Function definition

Scalar function only defined in vertices (+linear interpolation)

Avoid constant functions

Add a different epsilon on each $f(p)$ to insure the total ordrer

Identify critical points

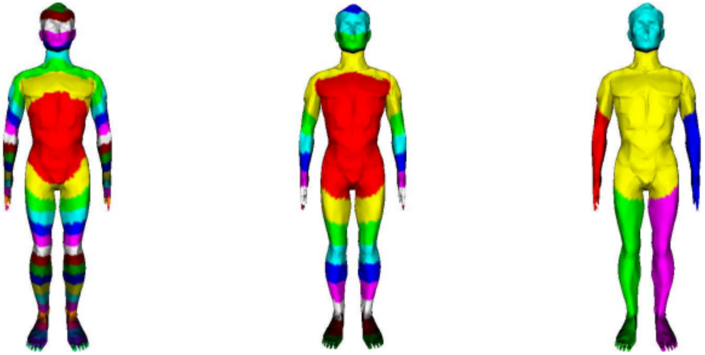
Compare local values in the 1-ring

Level sets

Compute piecwise-linear paths throw triangles

Reeb Graph

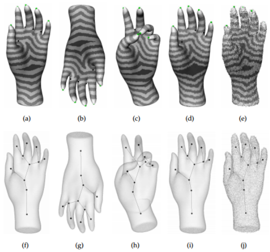
First application: mesh segmentation (Berretti *et al*, 2009)

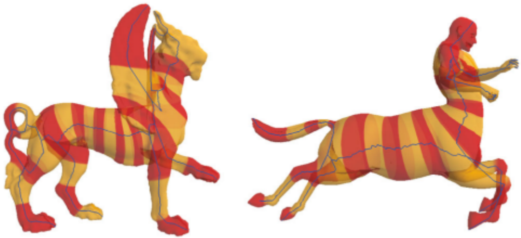


(a) (b) (c)

Reeb Graph

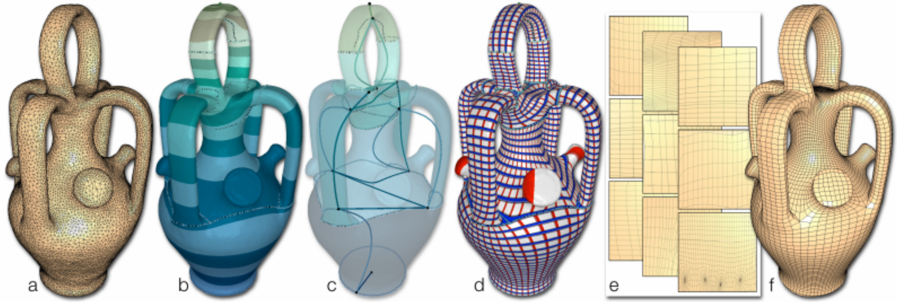
Second application: squeleton (PhD Tierny, 2008; Reuter 2010)





Reeb Graph

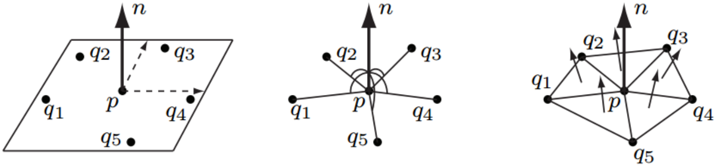
Third application: parameterization (Thierny 2011)



Mesh Analysis

Local Quantification

Normal computation
Intuition and practical questions



Strongly dependant of the scientific community

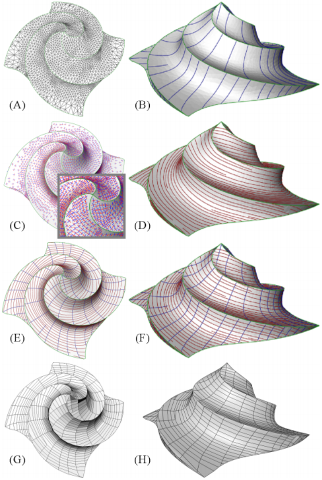
Mesh processing community

- Use the mesh and **only the mesh**
- Compute a **mean of the normals** in a neigborhood
- Supplementary ingredients:
  - **Radius** of the neigborhood
  - Area of the triangles

Principal curvatures

Illustrations (Cohen-Steiner and Morvan 2003, Alliez et al, 2003)

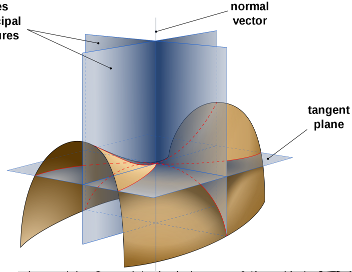
Minimum curvature



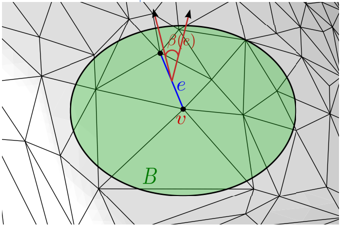
blue: minimum, red: maximum

Principal curvatures computation

Intuition and practical questions (Alliez et al, 2003)



$$\tau(v) = \frac{1}{|B|} \sum_{edges\ e} \beta(e)|e \cap B|\bar{e}\bar{e}^t$$

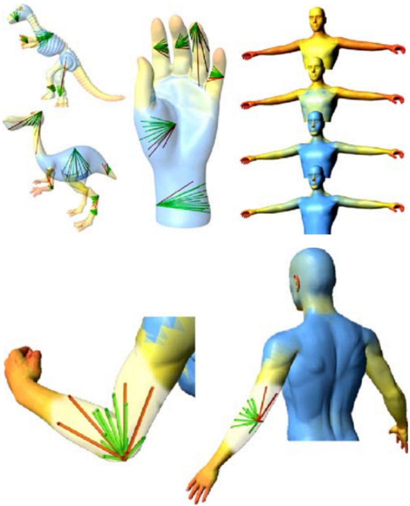


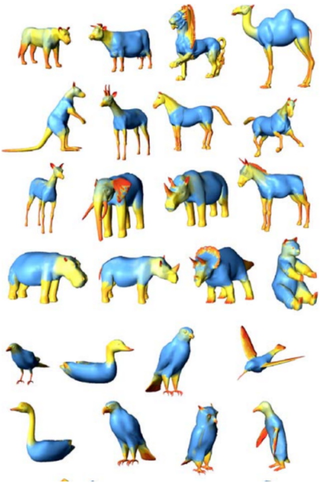
- B: geodesic disc window
- $\beta(e)$: angle betwen triangle normals
- $|e \cap B|$: length of e inside B
- $\bar{e}$: unit vector along e

Curvature values

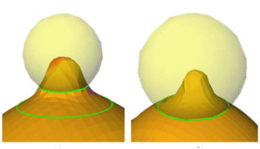
- κ_{min}, κ_{max}: **eigenvalues** of τ, min. and max. curvature **amplitudes**
- γ_{min}, γ_{max}: **eigenvectors** of τ, min. and max. curvature **vectors**

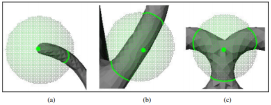
Shape Diameter Function (Shapira et al, 2008)

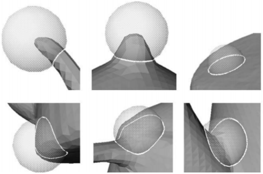


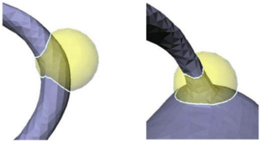


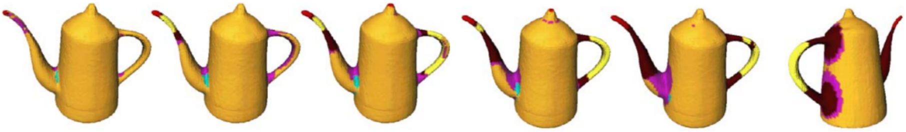
Local topology
Taylor (Mortara 2004)











Mesh Processing

Mesh Processing – ingredients

Mesh segmentation

Segmentation/labelling/cutting driven by geometry, topology, semantic

Local modifications

Mesh simplification, remeshing, smoothing

Global modifications

Parameterization, mesh deformation.

Applications

Computer graphics, medical imaging

imagse manquantes

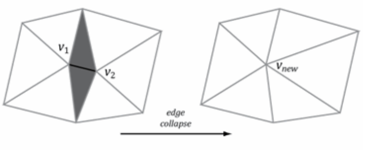
Mesh Processing

Local Modification

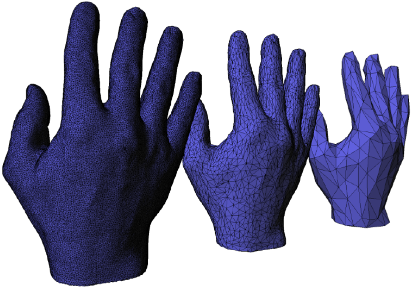
Mesh simplification

Edge Collapsing approaches (Hoppe 1993, Daniels 2008)

Hoppe 1993

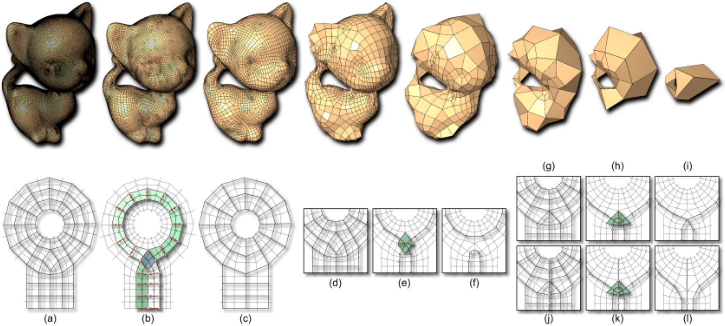


- Local cost defined by geometry
- Iterative optimization



Mesh simplification

Edge Collapsing approaches (Hoppe 1993, Daniels 2008)



Mesh simplification

Octree-based and local adjustment (Boubekeur 2007)

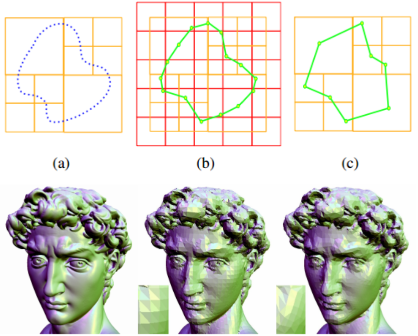
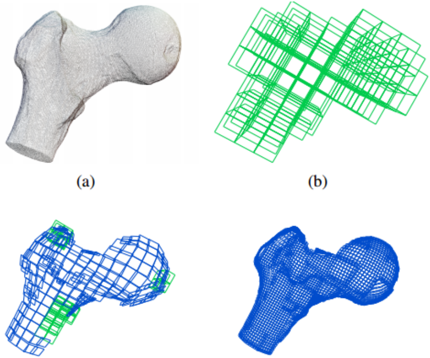
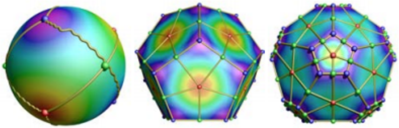


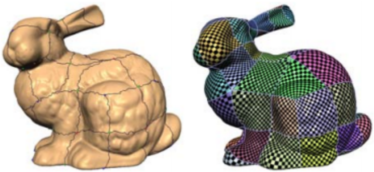
Figure 4: *Hierarchical mesh simplification with L^2 error bounded at 2.10^{-3}. Left: Original object (7M triangles). Middle: Octree simplification (1.75 sec. - 62856 triangles). Right: VS-Tree simplification (1.20 sec. - 52024 triangles).*



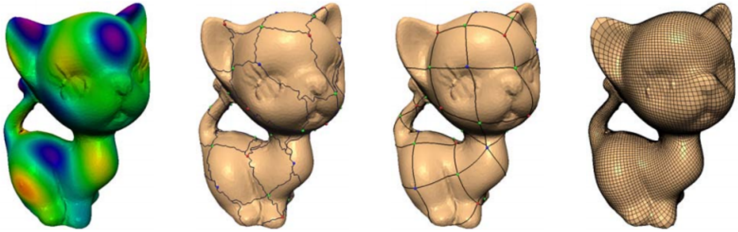
Mesh remeshing

Spectral approach (Dong 2006)



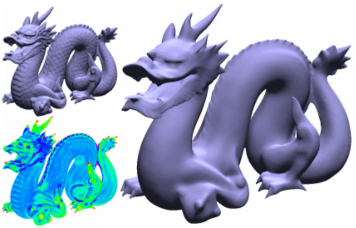


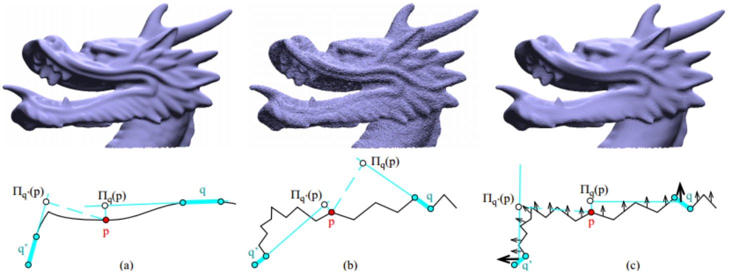
(a) Initial complex (b) Final complex



(a) Laplacian eigenfunction (b) Morse-Smale complex (c) Optimized complex (d) Semi-regular remeshing

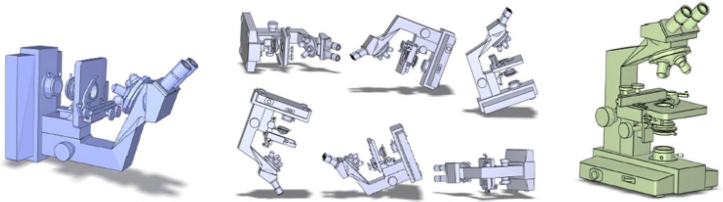
Mesh smoothing (Jones 2003)

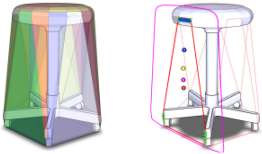




Prediction on neighbors Possible problems Correction using mollification

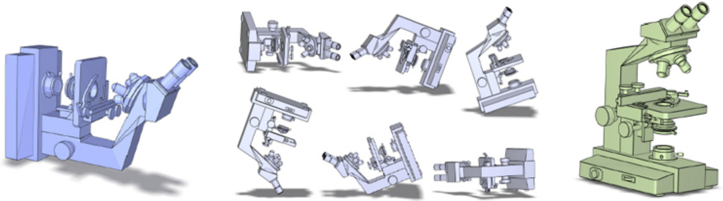
Mesh orientation

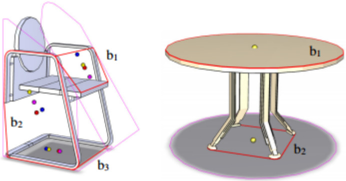




Facets of the convex hull

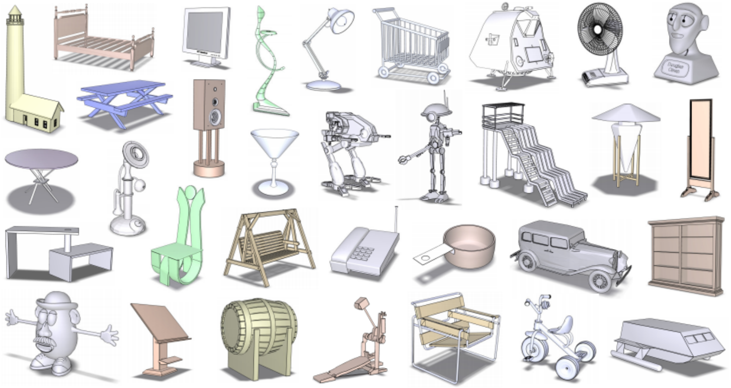
Mesh orientation





- projected center of mass (yellow)
- barycenter of supporting polygon (red)
- barycenter of convex hull proj. on sup. plane (pink)
- barycenter of actual base (blue)

Mesh orientation



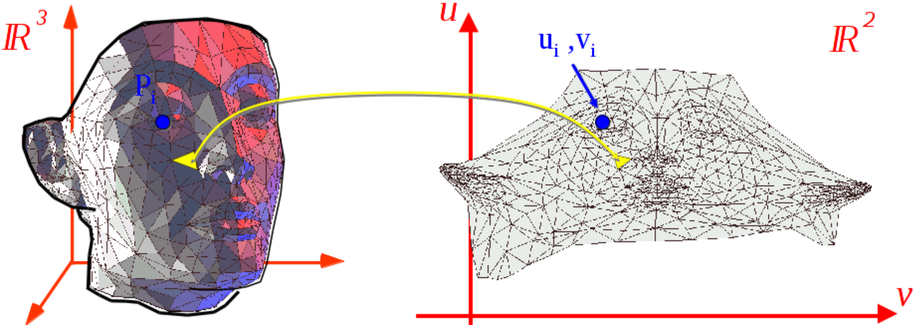
Ingredients: {static stability, symmetry, parallelism} + learning

Mesh Processing

Global Modifications

Mesh unfolding

Explicit parameterization (Scheffer 2011)



(u_i, v_i) values are the 2D coordinates of P_i

Mesh unfolding

Explicit parameterization (Scheffer 2011)

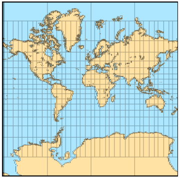
To evaluate the quality of a parameterization:

- *Conformal* parameterization: **preserve the angles**
- *Equivalent* parameterization: **preserve the areas**

Impossible to conciliate both, classical problem in cartography

orthographic
∼ 500 B.C.

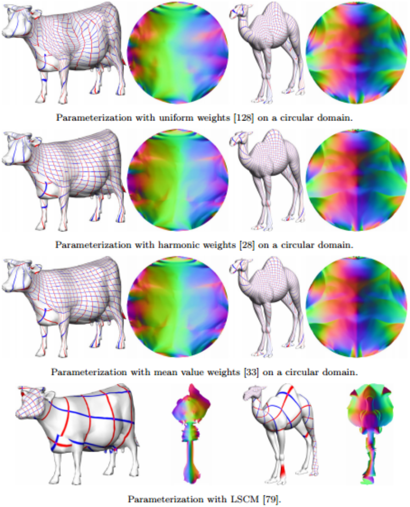
stereographic
∼ 150 B.C.
conforme



Mercator
1569
conform

Lambert
1772
equivalent

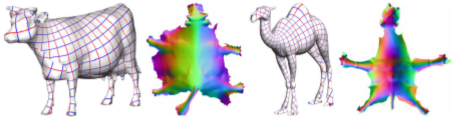
Mesh unfolding
Explicit parameterization (Scheffer 2011)



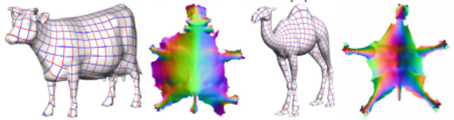
Fixed borders

Mesh unfolding

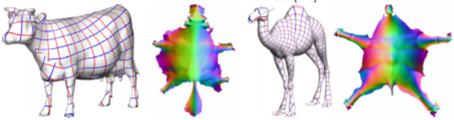
Explicit parameterization (Scheffer 2011)



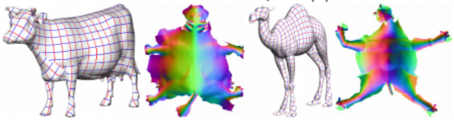
Parameterization with MIPS [54].



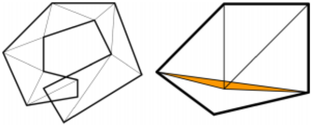
Parameterization with ABF++ [118].



Parameterization with circle patterns [62].



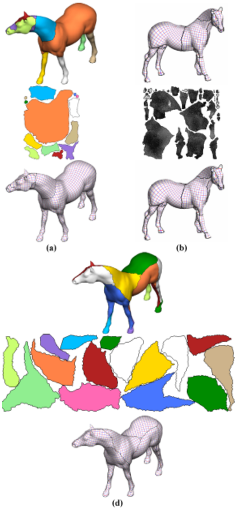
Stretch minimizing parameterization [107].



Non fixed borders

Mesh unfolding

Explicit parameterization (Scheffer 2011)

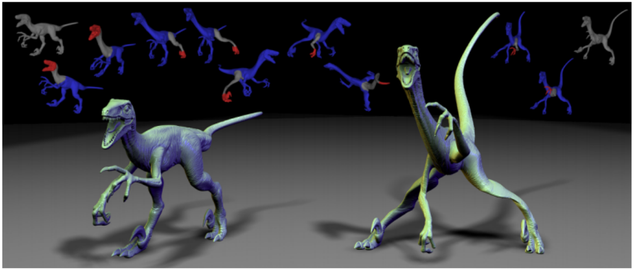


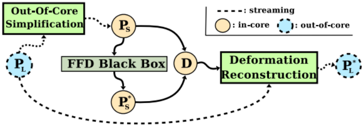
Mesh deformation

Skeleton-based approach (de Aguiar 2008)

Mesh deformation

Using proximity (Boubekeur 2008)





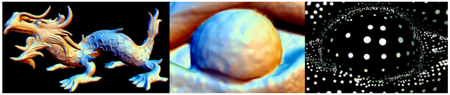
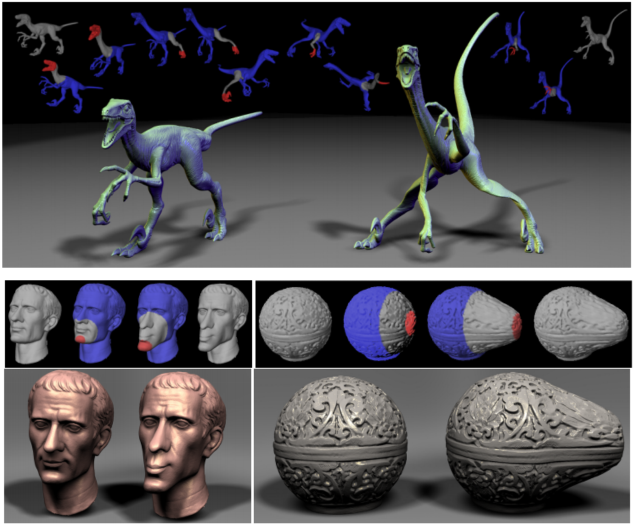


Figure 3: ***Left:*** *XYZRGB Dragon (7.2M triangles).* ***Middle:*** *Close-up of the eyeball on the original geometry.* ***Right:*** *Adaptive downsampling by out-of-core VS-Tree clustering.*

Mesh deformation

Using proximity (Boubekeur 2008)



Mesh Processing

Mesh Segmentation

Mesh Segmentation

Definition, motivations

Mesh segmentation

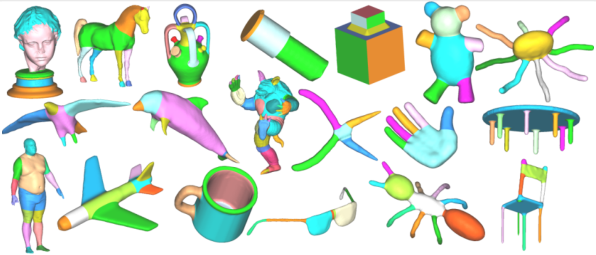
Split the mesh into regions with coherent properties

Properties

Geometry, topology, additionnal information

Motivations

Shape analysis, understanding, matching, partial recognition, substitution, ...

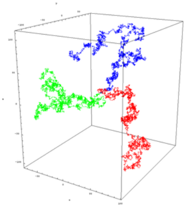


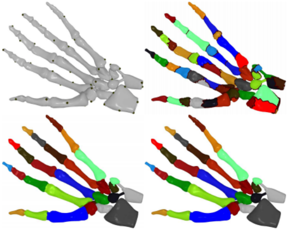
Mesh Segmentation

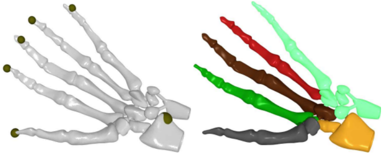
Random walk approaches (Lai 2008)

Random walk

- Graph-based approach
- Nodes and probability of walking from one node to another
- Description by a sparse linear system
- **Result:** probability to reach the point from a seed using a random walk







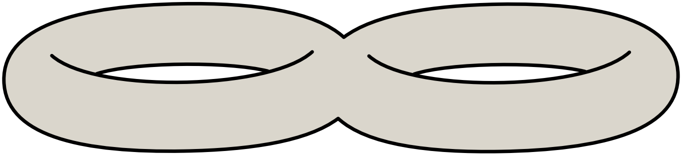
Question: what value for the edge probability?

Mesh cutting driven by topology

Optimal cutting — variation
Input: a surface $\mathcal{M}$ (non homeomorphic to a disc)
Begin

- **While** $\mathrm{genus}(\mathcal{M}) \neq 0$ **do**
  - Find the **shortest** non separating **loop** *l*
  - Cut $\mathcal{M}$ according to *l*
- Cut according to the **shortest spanning tree** joining the boundaries of $\mathcal{M}$

End



Genus: 2 — boundaries: 0

▸ Implementation details

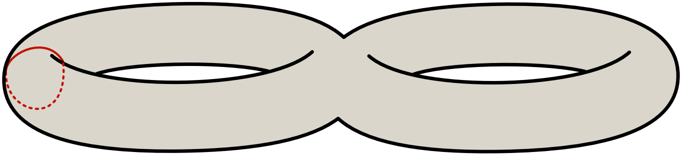
Erickson and S. Har-Peled, *Optimally Cutting a Surface into a Disk*, ACM Symposium on Computational Geometry, 2002

Mesh cutting driven by topology

Optimal cutting — variation
Input: a surface $\mathcal{M}$ (non homeomorphic to a disc)
Begin

- **While** $\text{genus}(\mathcal{M}) \neq 0$ **do**
  - Find the **shortest** non separating **loop** *l*
  - Cut $\mathcal{M}$ according to *l*
- Cut according to the **shortest spanning tree** joining the boundaries of $\mathcal{M}$

End



Genus: 1 — boundaries: 2

▸ Implementation details

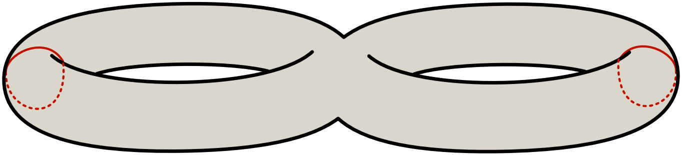
Erickson and S. Har-Peled, *Optimally Cutting a Surface into a Disk*, ACM Symposium on Computational Geometry, 2002

Mesh cutting driven by topology

Optimal cutting — variation
Input: a surface $\mathcal{M}$ (non homeomorphic to a disc)
Begin

- **While** $\mathrm{genus}(\mathcal{M}) \neq 0$ **do**
  - Find the **shortest** non separating **loop** *l*
  - Cut $\mathcal{M}$ according to *l*
- Cut according to the **shortest spanning tree** joining the boundaries of $\mathcal{M}$

End



Genus: 0 — boundaries: 4

Implementation details

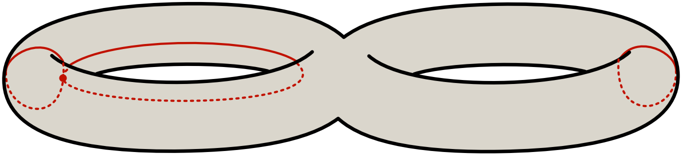
Erickson and S. Har-Peled, *Optimally Cutting a Surface into a Disk*, ACM Symposium on Computational Geometry, 2002

Mesh cutting driven by topology

Optimal cutting — variation
Input: a surface $\mathcal{M}$ (non homeomorphic to a disc)
Begin

- **While** $\text{genus}(\mathcal{M}) \neq 0$ **do**
  - Find the **shortest** non separating **loop** *l*
  - Cut $\mathcal{M}$ according to *l*
- Cut according to the **shortest spanning tree** joining the boundaries of $\mathcal{M}$

End



Genus: 0 — boundaries: 3

▸ Implementation details

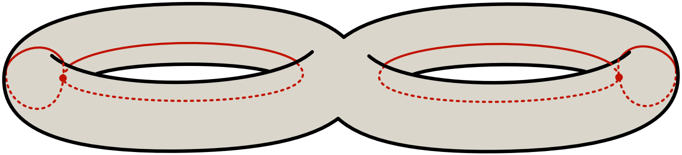
Erickson and S. Har-Peled, *Optimally Cutting a Surface into a Disk*, ACM Symposium on Computational Geometry, 2002

Mesh cutting driven by topology

Optimal cutting — variation
Input: a surface $\mathcal{M}$ (non homeomorphic to a disc)
Begin

- **While** $\mathrm{genus}(\mathcal{M}) \neq 0$ **do**
  - Find the **shortest** non separating **loop** *l*
  - Cut $\mathcal{M}$ according to *l*
- Cut according to the **shortest spanning tree** joining the boundaries of $\mathcal{M}$

End



Genus: 0 — boundaries: 2

Implementation details

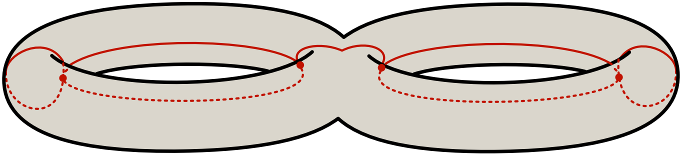
Erickson and S. Har-Peled, *Optimally Cutting a Surface into a Disk*, ACM Symposium on Computational Geometry, 2002

Mesh cutting driven by topology

Optimal cutting — variation
Input: a surface $\mathcal{M}$ (non homeomorphic to a disc)
Begin

- **While** $\mathrm{genus}(\mathcal{M}) \neq 0$ **do**
  - Find the **shortest** non separating **loop** *l*
  - Cut $\mathcal{M}$ according to *l*
- Cut according to the **shortest spanning tree** joining the boundaries of $\mathcal{M}$

End



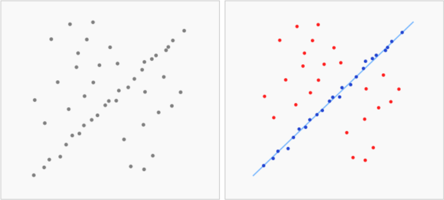
Genus: 0 — boundaries: 1

▸ Implementation details

Erickson and S. Har-Peled, *Optimally Cutting a Surface into a Disk*, ACM Symposium on Computational Geometry, 2002

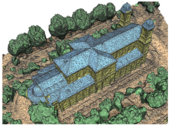
Primitive fitting
Intuition

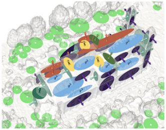
On point clouds (RANdom SAmple Consensus)



On 3D meshes

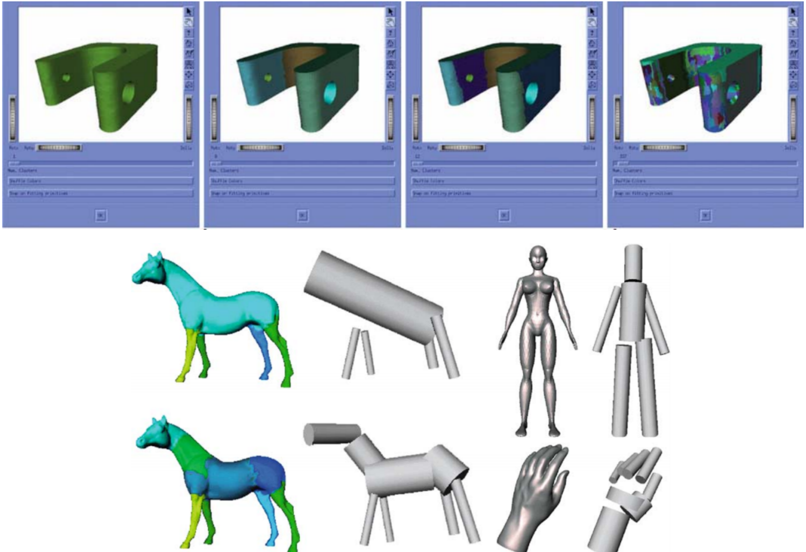
- Sphere, plane, cylinder, cone, etc.
- Sparse data with structure (triangles)





Primitive fitting

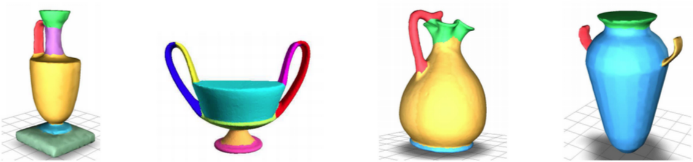
Hierarchical approach (Attene 2006)



Semantic segmentation

Functionality (Laga 2015)

- Segmentation
  - Decompose a 3D model into nearly-convex parts

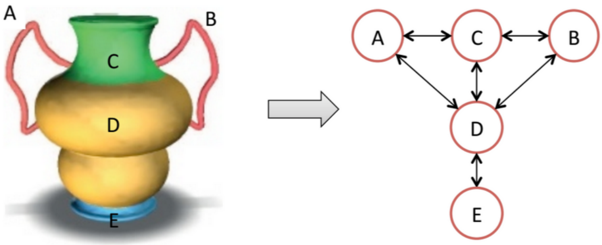


Segmentation with Randomized Cuts (Golovinskiy et al. 2009)

Semantic segmentation

Functionality (Laga 2015)

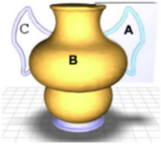
- Initial structural graph
  - Each shape part forms a node
  - Connect adjacent parts with an undirected edge



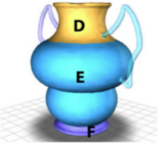
Semantic segmentation

Functionality (Laga 2015)

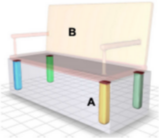
- Enrich the graph with more structural relations
  - Side contact, co-centricity, symmetry, containment



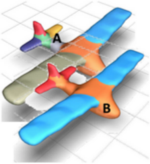
A and C are in side contact with B



D, E, and F are co-centric



Parts in A are symmetric

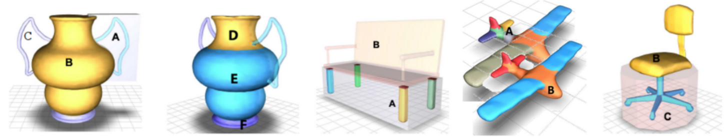


A is contained in B

Semantic segmentation

Functionality (Laga 2015)

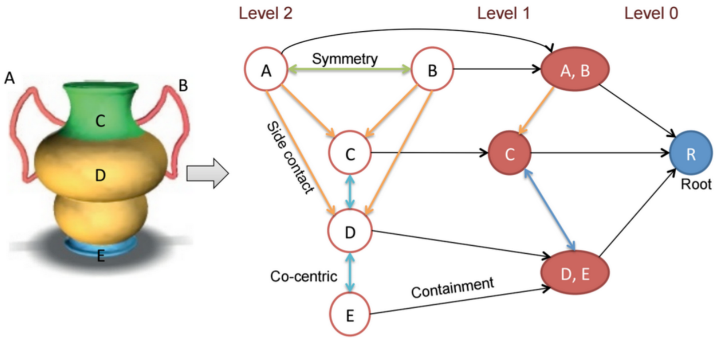
- Enrich the graph with more structural relations
  - Side contact, co-centricity, symmetry, containment
  - Horizontal support (in case we know the upright orientation)



A and C are in side contact with B | D, E, and F are co-centric | Parts in A are symmetric | A is contained in B | C is a horizontal support of B

Semantic segmentation

Functionality (Laga 2015)



Semantic segmentation

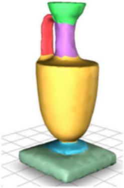
Functionality (Laga 2015)

- Two nodes P_A and P_B are similar if
  - Their geometries are similar and their contexts are similar
- Compare the geometry of the nodes and the local structure of the graph around those nodes
  - Node geometry is captured with shape descriptors
  - Local structure captured with paths on the structural graph

Semantic segmentation

Functionality (Laga 2015)

- Represent each node with geometric descriptors

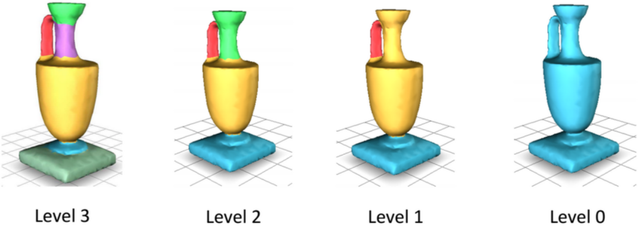


Level 3

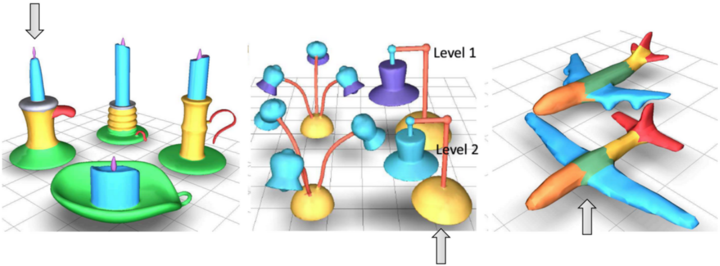
- D2 shape distribution [Osada et al. 2002]
- Component's size
  (radius of its bounding sphere)
- The three eigenvalues of the component
  (shapes are normalized for scale).

Semantic segmentation
Functionality (Laga 2015)

- Represent each node with geometric descriptors



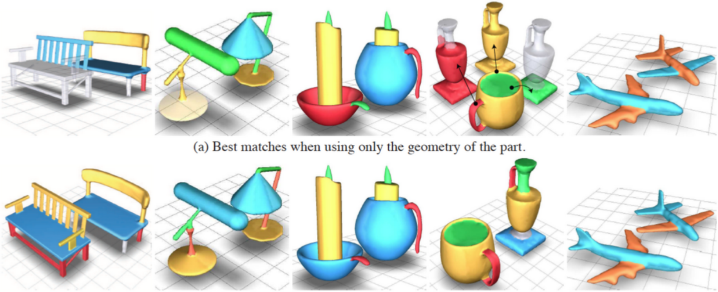
Semantic segmentation
Functionality (Laga 2015)



Semantic segmentation

Functionality (Laga 2015)

Geometry vs. Geometry + structure



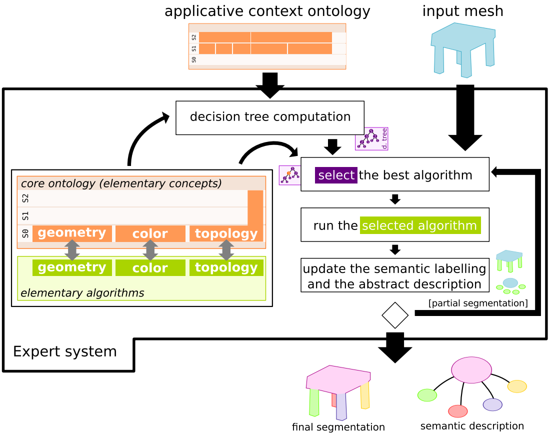
(a) Best matches when using only the geometry of the part.

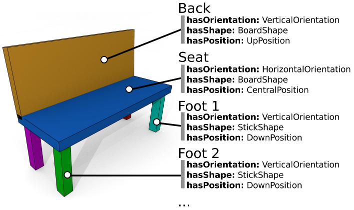
(b) Best matches when using part context.

25

Semantic segmentation

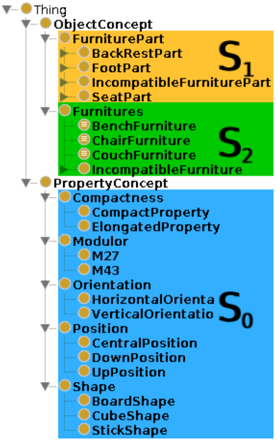
Integrate the expert knowledge (Dietenbeck 2015)

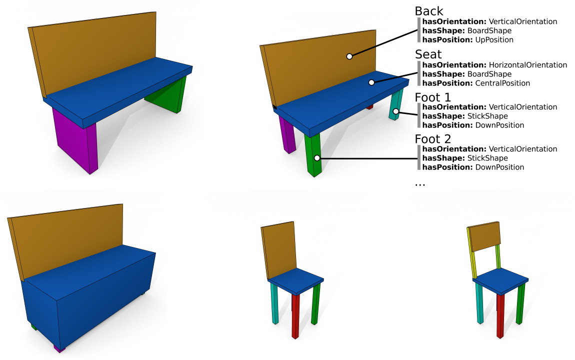




Semantic segmentation

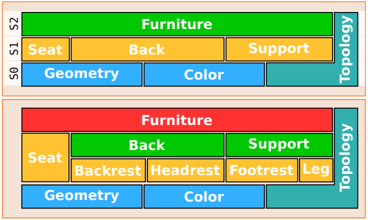
Integrate the expert knowledge (Dietenbeck 2015)

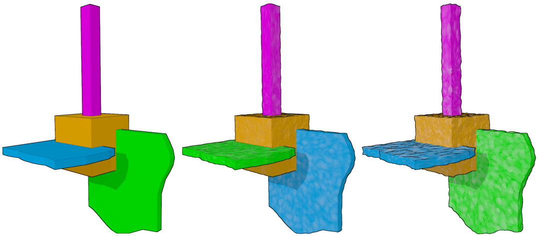




Semantic segmentation

Integrate the expert knowledge (Dietenbeck 2015)



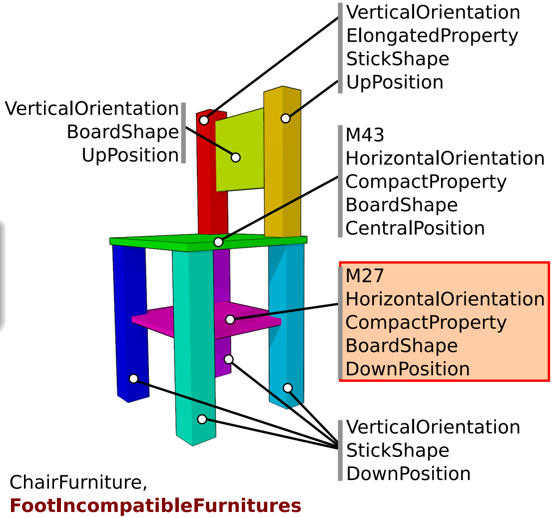


Semantic segmentation

Integrate the expert knowledge (Dietenbeck 2015)

Closed world

- Detect incoherent configurations

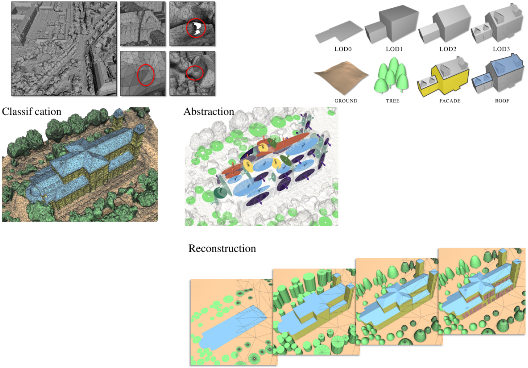


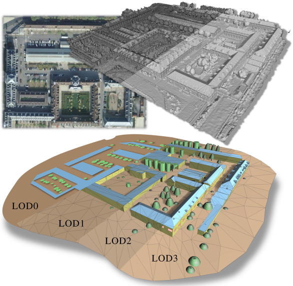
Semantic segmentation

Handle big data (Verdié 2015)

LOD Generation for Urban Scenes

- Model reconstruction
- Segmentation
- Level of details



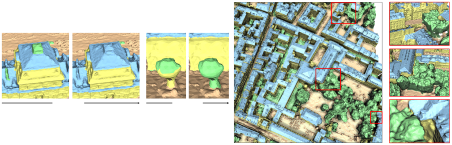


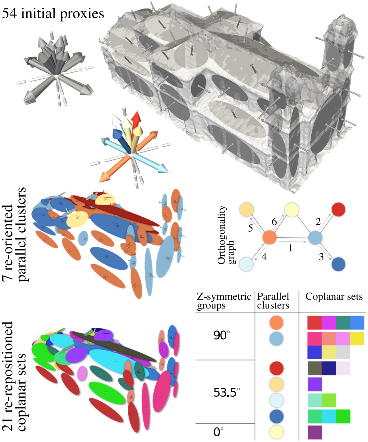
Semantic segmentation
Handle big data (Verdié 2015)

Semantic Rules

—*Rule 1*. superfacets labeled as *tree* and adjacent to only superfacets labeled as *roof* are re-labeled *roof*. This rule relies on the common assumption that large trees are not located on top of roofs.

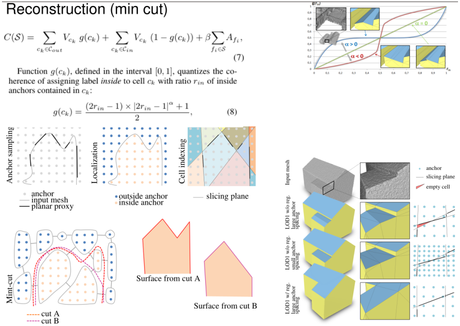
—*Rule 2*. superfacets labeled as *facade* and adjacent to superfacets labeled as *tree* and *ground* are turned to *tree*.

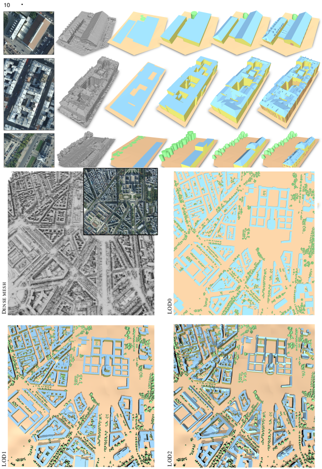




Semantic segmentation

Handle big data (Verdié 2015)





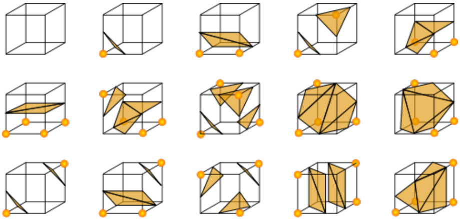
11M facets

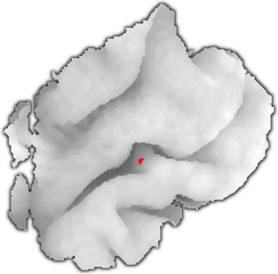
Mesh Processing

Applications

Cortical surface mapping

Marching cube (Lorensen 1987)



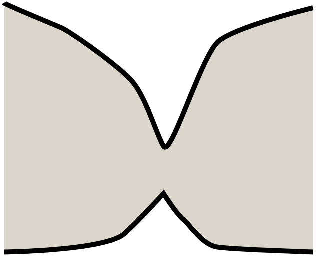


Cortical surface mapping

Topological correction

Reasons of the topological errors

- **Partial volume** effect
- Imprecision of the **segmentation**

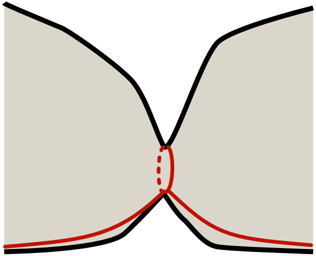


Cortical surface mapping

Topological correction

Reasons of the topological errors

- **Partial volume** effect
- Imprecision of the **segmentation**

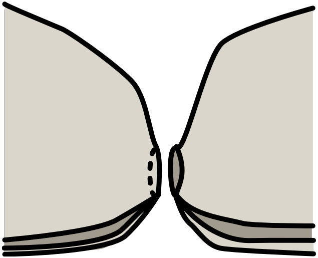


Cortical surface mapping

Topological correction

Reasons of the topological errors

- **Partial volume** effect
- Imprecision of the **segmentation**

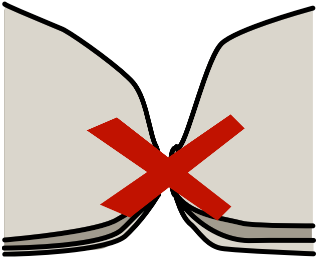


Cortical surface mapping

Topological correction

Reasons of the topological errors

- **Partial volume** effect
- Imprecision of the **segmentation**

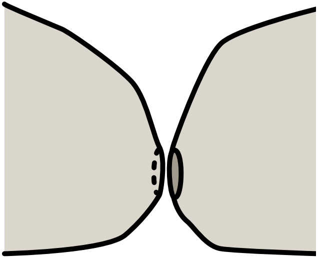


Cortical surface mapping

Topological correction

Reasons of the topological errors

- **Partial volume** effect
- Imprecision of the **segmentation**

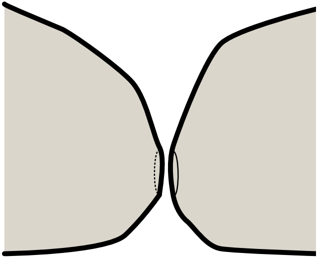


Cortical surface mapping

Topological correction

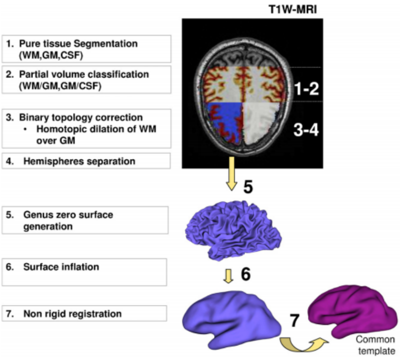
Reasons of the topological errors

- **Partial volume** effect
- Imprecision of the **segmentation**

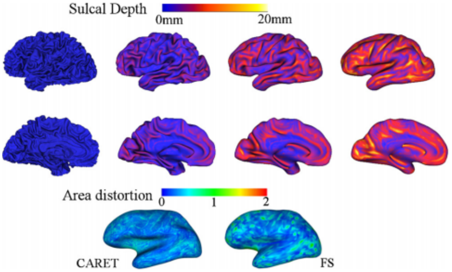


An example on medical imaging

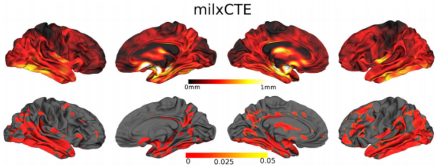
Cortical surface mapping to quantify atrophy in Alzheimer's disease (Acosta 2009)



Surface extraction and matching



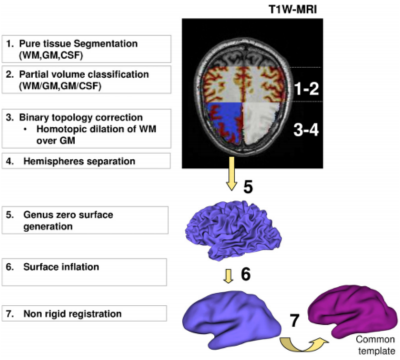
Sulcal depth



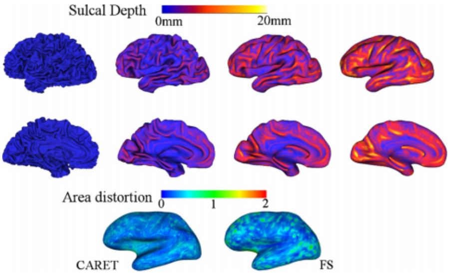
Significant locations

An example on medical imaging

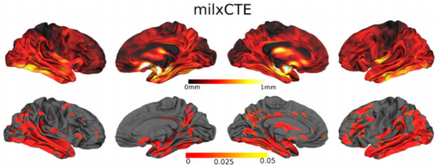
Cortical surface mapping to quantify atrophy in Alzheimer's disease (Acosta 2009)



Surface extraction and matching

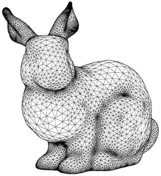


Sulcal depth



Significant locations

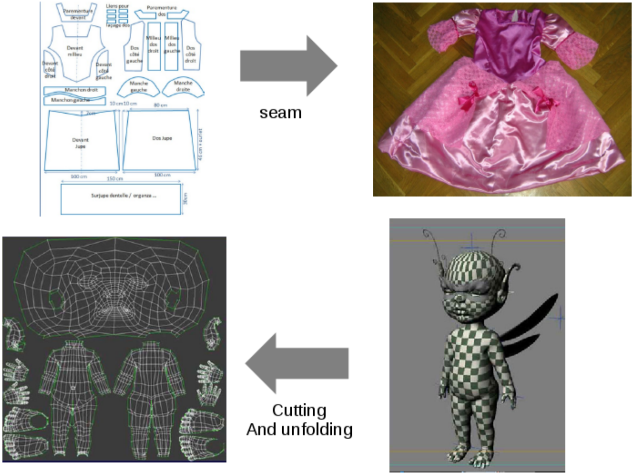
Computer graphics workflow



Classical pipeline

- Mesh modeling
- Texture mapping
- Rendering

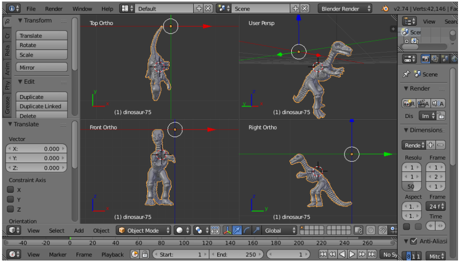
Computer graphics workflow



Computer graphics workflow

Conclusion

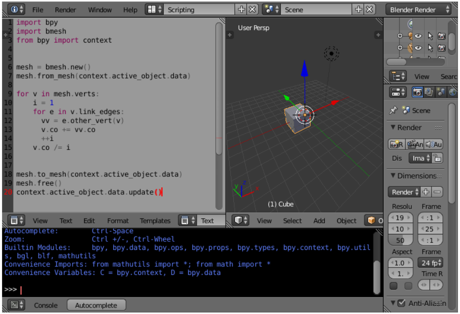
Tomorrow morning: blender (1)



Python scripting

- Introduction to the interface
- Shape modelization
- Texture and light adjustment
- Scene rendering

Friday morning: blender (2)



Python scripting

- Introduction to the API
- Tutorial on mesh processing

Resources about blender

Algorithms

Algorithms

Dijkstra

Dijkstra algorithm

```
Data: G: Graph; source: vertex
Data: dist: list of floats, prev: list of vertices
dist[source] ← 0;
prev[source] ← undefined;
forall vertex v in G do
    if v ≠ source then
        dist[v] ← infinity;
        prev[v] ← undefined ;
    end
    add v to Q;
end
while Q is not empty do
    u ← vertex in Q with min dist[u];
    remove u from Q;
    forall neighbor v of u do
        alt ← dist[u] + length(u, v);
        if alt ¡ dist[u] then
            dist[v] ← alt;
            prev[v] ← u;
        end
    end
end
```

Go back to Geodesic distance

Animation on Wikipedia

Breadth first search

▸ Go back to Connected Components

BFS algorithm (G, v)

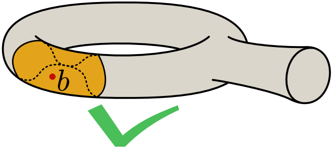
```
Data: G: Graph; v: vertex
let Q be a queue;
label v as discovered;
while Q is not empty do
    v' ← Q.dequeue();
    process(v');
    forall edges from v' to w in G.adjacentEdges(v') do
        if w is not labeled as discovered then
            Q.enqueue(w);
            label w as discovered;
        end
    end
end
```

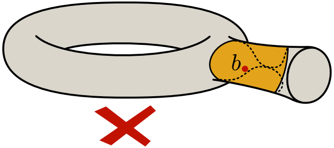
Depth-first search: replace the queue by a stack

Shortest non-separating loop (1)

Given a **basepoint** b

- Using a **“geodesic” distance** (e.g. Dijkstra[2])
- Circular **wavefront propagation**
- Catching the **junctions** (and their nature)





Complexité: $O(n \log n)$

[2] Dijkstra. E. W. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1) :269–271, 1959.

Shortest non-separating loop (2)

General method:

- Construct a **set** B potential **basepoints**
- $\forall b \in B$ compute the **shortest** non-separating **loop**
- Keep **the shortest**

Complexité: $O(|B| n \log n)$

Shortest non-separating loop (2)

General method:

- Construct a **set** B potential **basepoints**
- $\forall b \in B$ compute the **shortest** non-separating **loop**
- Keep **the shortest**

How to compute B?

Complexité: $O(|B| n \log n)$

Shortest non-separating loop (2)

General method:

- Construct a **set** B potential **basepoints**
- $\forall b \in B$ compute the **shortest** non-separating **loop**
- Keep **the shortest**

How to compute B?

Principle: a set **crossed by all the non-separating loops**

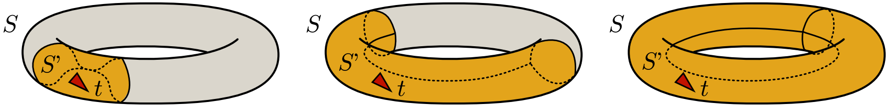
Complexité: $O(|B| n \log n)$

Shortest non-separating loop (3)

Principle: a set **crossed by all the non-separating loops**
Input: a 2-mesh $\mathcal{M}$ (non homeomorphic to a sphere)
Begin

- Select a starting triangle t
- Add t to a new 2-mesh $\mathcal{M}'$
- **While** all triangles are not visited **do**
  - Select a non visited triangle t_i adjacent to a visited triangle
  - Stick t_i to its neighbours on $\mathcal{M}'$(preserving **the genus of** $\mathcal{M}'$ **: 0**)
- B: set of the boundary points of $\mathcal{M}'$

end



Go back to the main algorithm