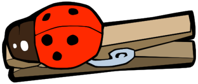
Tools for tiling surfaces

Jean-Marie Favreau

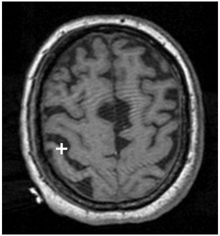
Univ. Blaise Pascal (Clermont-Ferrand II), LIMOS UMR CNRS 6158
Clermont-Ferrand (France)
PhD supervisor: Vincent Barra

2006-2009



Motivations

Amount of **available data**

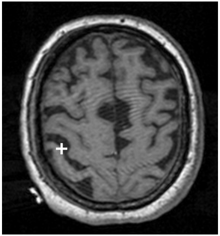



- Various acquisitions (medical imaging, 3D scanners)
- Modeling (computer graphics, Computer-aided design)

Sources: AIM@Shape, Stanford 3D Scanning Repository

Motivations

Amount of **available data**



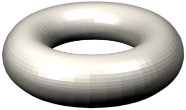
Various problems:

- Analysis and measurement
- Transformations, distortions

Sources: AIM@Shape, Stanford 3D Scanning Repository

Geometry

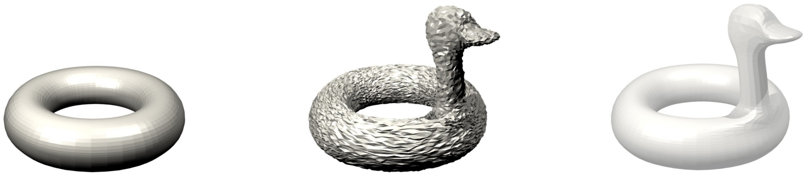
Various scale levels of **local** information



Quantification: local curvatures, distances, patterns, ...

Geometry

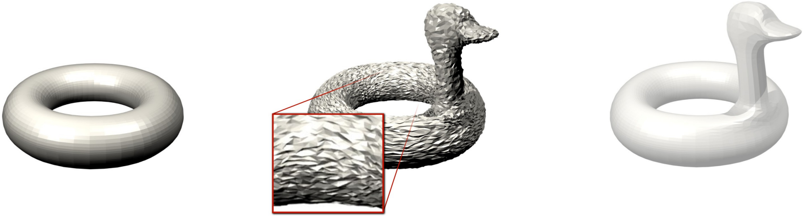
Various scale levels of **local** information



Quantification: local curvatures, distances, patterns, ...

Geometry

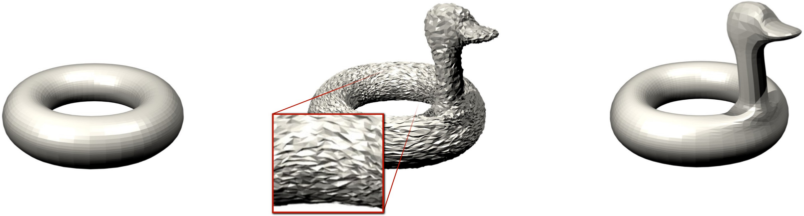
Various scale levels of **local** information



Quantification: local curvatures, distances, patterns, ...

Geometry

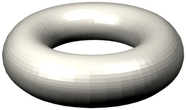
Various scale levels of **local** information

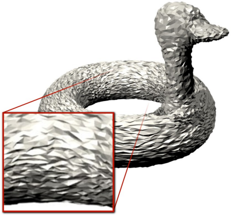


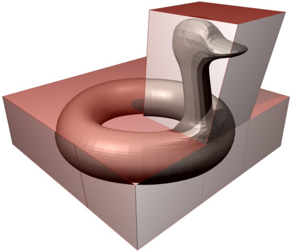
Quantification: local curvatures, distances, patterns, ...

Geometry

Various scale levels of **local** information



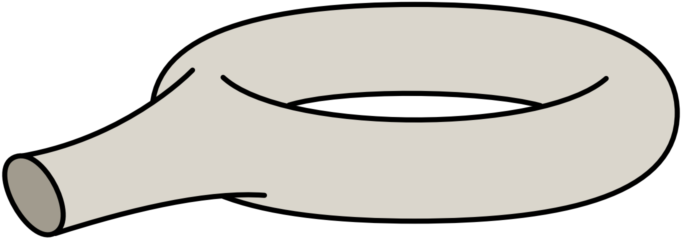




Quantification: local curvatures, distances, patterns, ...

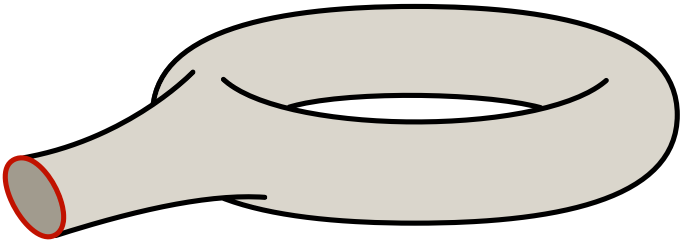
Topology

Global information



Topology

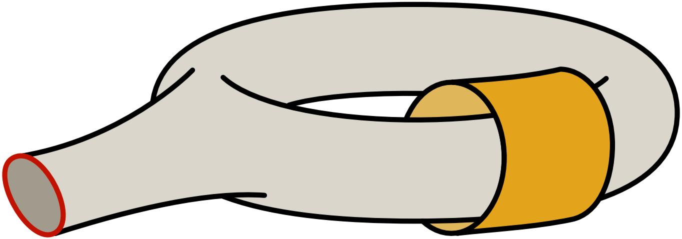
Global information



- Number of **boundaries**

Topology

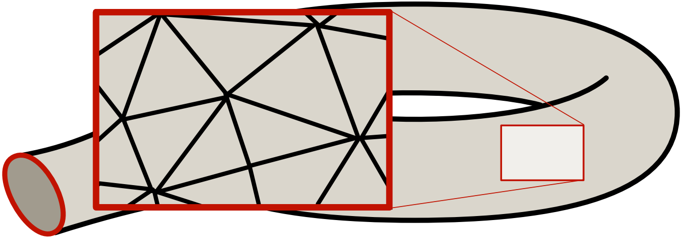
Global information



- Number of **boundaries**
- Number of handles: **genus**

Simplicial complexes

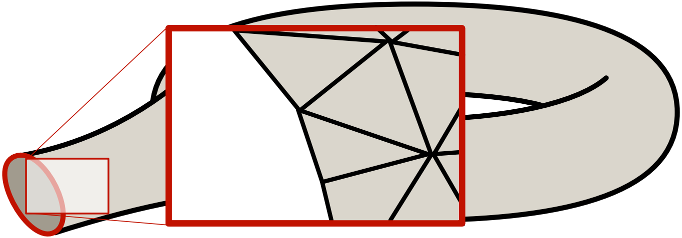
Discrete representation by 2-meshes



- **Disc**: non-boundary region

Simplicial complexes

Discrete representation by 2-meshes



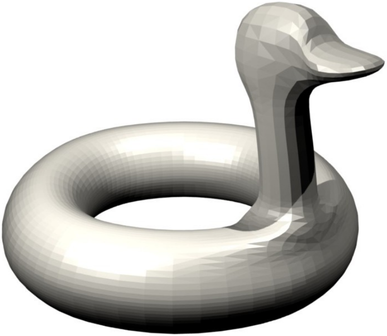
- **Disc**: non-boundary region
- **Half-disc**: boundary region

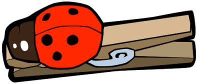
Manipulating surfaces

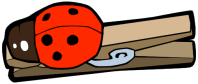
Implies using both **geometry** and **topology**

- Topological tools: **global**
- Geometrical components: **local**

Towards a global using of the geometry

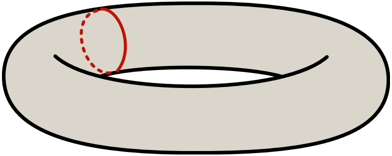






Cutting reasons

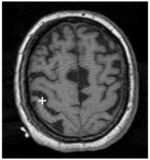
- Topological changes

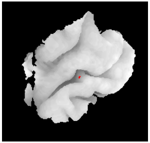


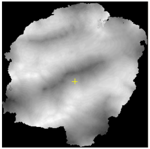
Cutting reasons

- Topological changes
- Segmentation

Cutting reasons





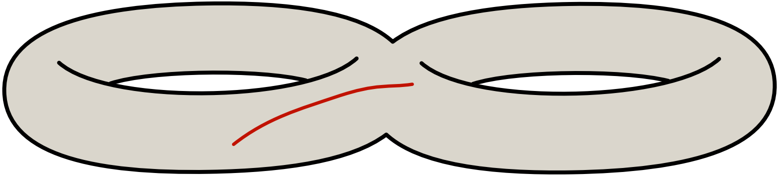


- Topological changes
- Segmentation
- Facilitate following manipulations

Paths, loops

Cutting according to:

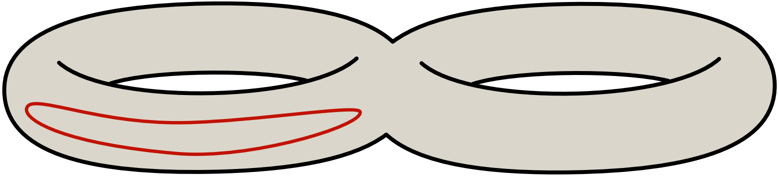
- a path



Paths, loops

Cutting according to:

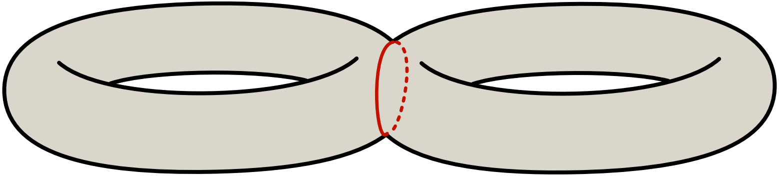
- a path
- a loop



Paths, loops

Cutting according to:

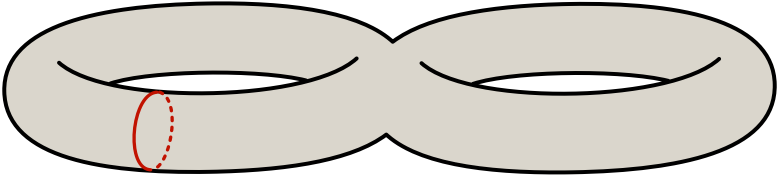
- a path
- a loop



Paths, loops

Cutting according to:

- a path
- a non-separating loop

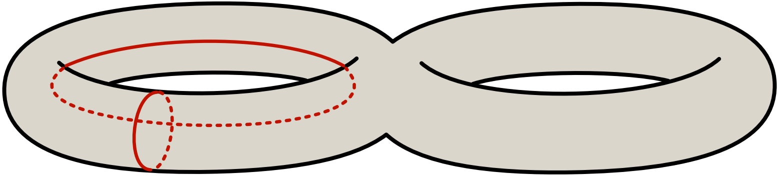


⇒ **Reducing** the genus

Paths, loops

Definition

A **cutting** on a 2-mesh $\mathcal{M}$ is defined as a set of paths and loops $(C_i)_{1 \leq i \leq n}$



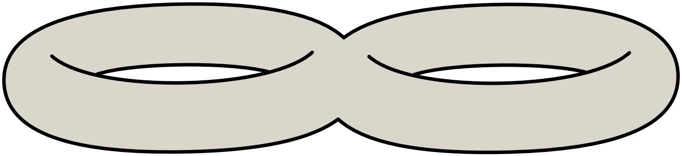
$\Rightarrow$ **Reducing** the genus and the number of boundaries

Topological cutting

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

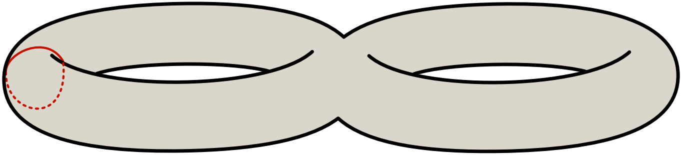
J. Erickson and S. Har-Peled, *Optimally Cutting a Surface into a Disk*, ACM Symposium on Computational Geometry, 2002

Topological cutting

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: 1 — boundaries: 2

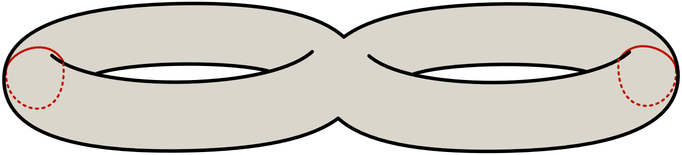
J. Erickson and S. Har-Peled, *Optimally Cutting a Surface into a Disk*, ACM Symposium on Computational Geometry, 2002

Topological cutting

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

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

Topological cutting

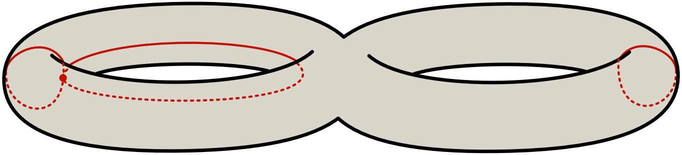
Optimal cutting — variation

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

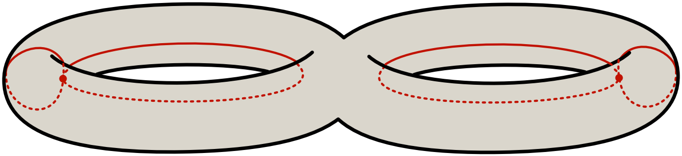
J. Erickson and S. Har-Peled, *Optimally Cutting a Surface into a Disk*, ACM Symposium on Computational Geometry, 2002

Topological cutting

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: 2

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

Topological cutting

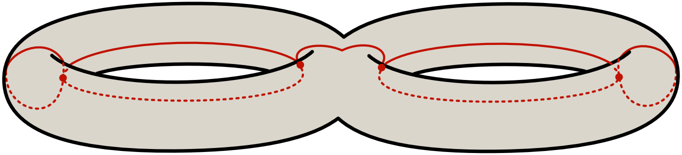
Optimal cutting — variation

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 — boundares: 1

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

Computation time, topological cutting

Surface		Algorithme exact		Algorithme approché	
Genus	# vertices	Time (s)	Length	Time (s)	Length
1	3,072	0.51	3.139,33	0.04	3.245,53
1	12,288	5.09	3.139,33	0.19	3.180,83
2	3,326	1.05	7.070,2	0.08	7.676,53
2	13,310	9.72	7.070,2	0.42	7.392,49
3	6,908	3.45	34.360,9	0.34	55.308,7
3	27,644	32.49	34.360,9	1.84	37.619
4	8,826	5.48	41.770,3	0.68	83.485,4
4	35,322	58.37	41.770,3	3.69	66.257,5
104	543,652	–	–	48,mn	–

Comparison of an exact method and an approximative cutting method using shortest non separating loops.

Tiling

Figure: M. C. Escher

Classical definition:

- On the plane
- **Repeated** patterns (tiles)
- Topologically trivial

Tiling

Figure: tiling by discs and cylinders

In our context:

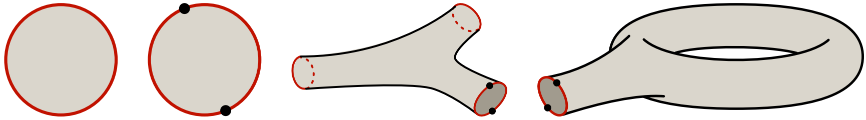
- On surfaces
- Topology of the tiles not fixed
- Need for a **precise description**

m-cellular complex

Definition

A *m-cellular complex* $\mathcal{C}$ is a set of m-cells with :

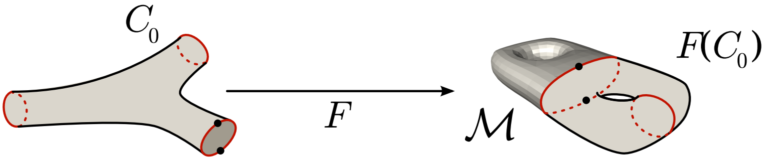
- 2-*m-cells*: **compact connex** 2-**manifold** with boundary
- 1-*m-cells*: **simple curves** (closed or opened)
- 0-*m-cells*: **points**
- **boundary** of any *i*-m-cell: $(i-1)$-sub-complex
- The **intersection** of two m-cells is an m-cell of $\mathcal{C}$



Tile

$F : C \longrightarrow \mathcal{M}$ embedding of a 2-m-cell C in a 2-mesh $\mathcal{M}$, with:

- F continuous.
- v **vertex** of $C \Rightarrow F(v)$ 0-**simplex** of $\mathcal{M}$.
- e **edge** of $C \Rightarrow F(e)$ **set of** 1-**simplexes** de $\mathcal{M}$.
- b **body** of $C \Rightarrow F(b)$ **set of** 2-**simplexes** de $\mathcal{M}$.
- F **injective** on the interior of C. Images of two edges or two vertices should be equal.

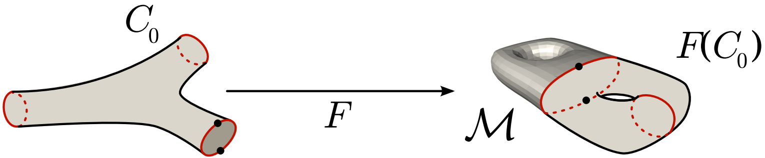


Tile

Equivalence relation

$F \sim F'$ iff for all cell, edge or vertex X of C, $F(X) = F'(X)$.

M-tiles = equivalence classes

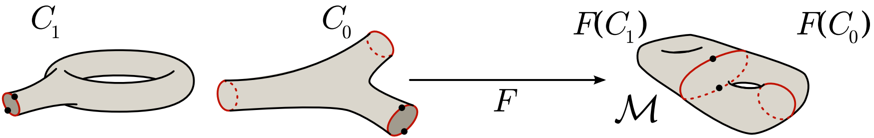


Tiling

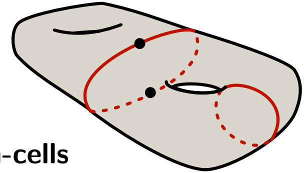
M-tiling defined by equivalence classes of embeddings $F : \mathcal{C} \longrightarrow \mathcal{M}$ where:

- $\forall$ 2-m-cell C of $\mathcal{C}$, $F_{|C}$ is an **M-tile**
- F **injective** except on the boundaries
- F is **surjective**

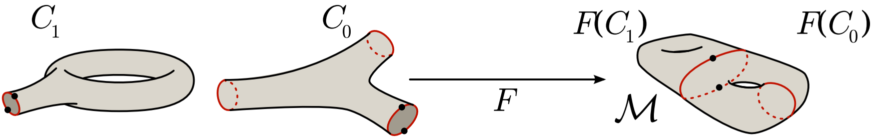
Two embeddings F and F' are equivalents $\Leftrightarrow \forall X \in \mathcal{C}$, $F(X) = F'(X)$.

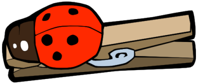


Cutting/tiling duality



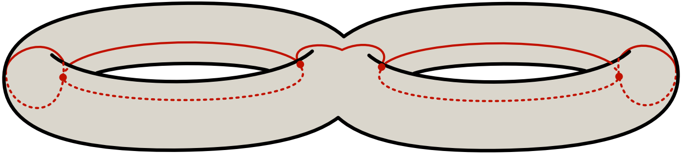
- **Paths and loops**: images of the 1-**m-cells**
- **Extremities** of the paths: images of the 0-**m-cells**
- **Complement**: images of the 2-**m-cells**





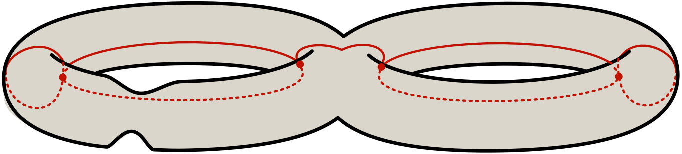
Geometry as an component in the cutting process

General structure of the algorithms: **topology**



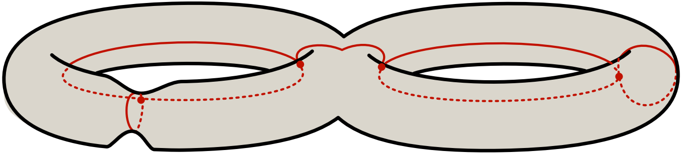
Geometry as an component in the cutting process

General structure of the algorithms: **topology**



Geometry as an component in the cutting process

General structure of the algorithms: **topology**

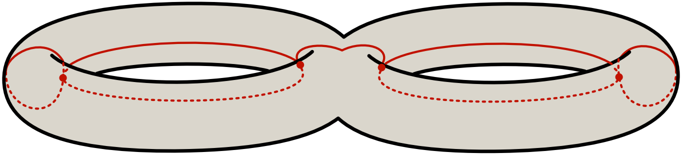


Precise driving by **geometry**

Distance

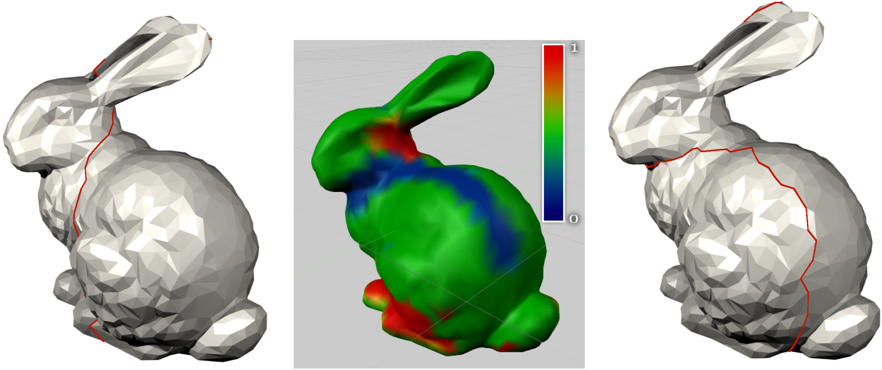
Topological and geometrical **cutting**:

- Length of the **loops** and **paths**
- Depends on **edges' length** (Dijkstra)
- Generally speaking, Euclidean distance: $\mathbb{R}^3$: $l_3(e)$



Other choices available

Cutting driven by user map and/or application

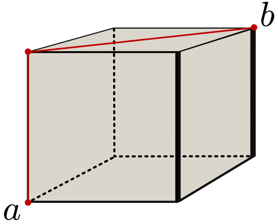


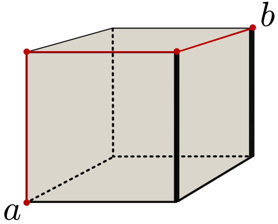
$$m_u : V \rightarrow [0,1]$$

$$l\left((v_1, v_2)\right) = l_3\left((v_1, v_2)\right) \times m_u(v_1) \times m_u(v_2)$$

Using local curvature

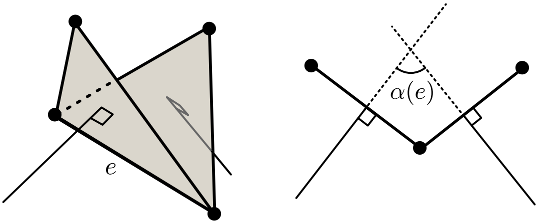
Constrain the cutting using local **curvature**





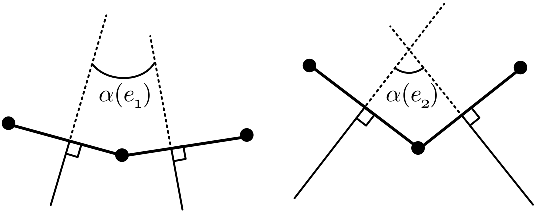
Using local curvature

Constrain the cutting using local **curvature**



Using local curvature

Constrain the cutting using local **curvature**

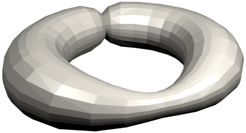


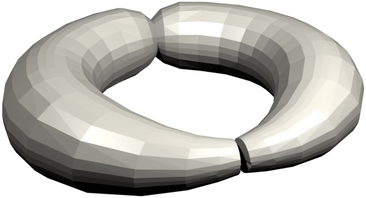
$$l(e) = l_3(e) \times (1 - c\frac{|\alpha(e)|}{\pi})$$

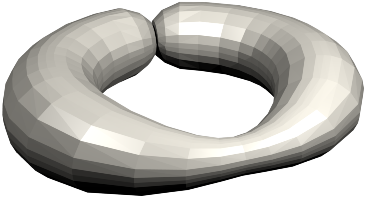
with c constant

Using local curvature

Constrain the cutting using local **curvature**

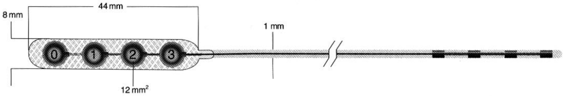






Medical context

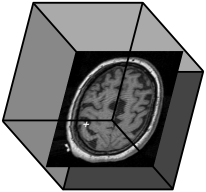
Cortical stimulation

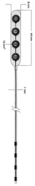


J.-M. Favreau and V. Barra, *Conformal Flattening of the Cortical Surface using a Topological and Geometrical Cutting*, SEECCM'09

Medical context

Cortical stimulation





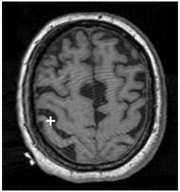
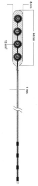
Available data:

- Anatomical MRI

J.-M. Favreau and V. Barra, *Conformal Flattening of the Cortical Surface using a Topological and Geometrical Cutting*, SEECCM'09

Medical context

Cortical stimulation, electrodes' localization

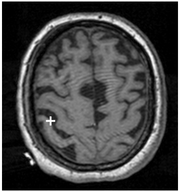
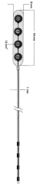
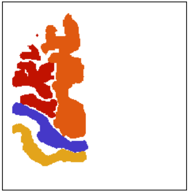

Available data:

- Anatomical MRI
- 3D location of the electrodes

J.-M. Favreau and V. Barra, *Conformal Flattening of the Cortical Surface using a Topological and Geometrical Cutting*, SEECCM'09

Medical context

Cortical stimulation, electrodes' localization

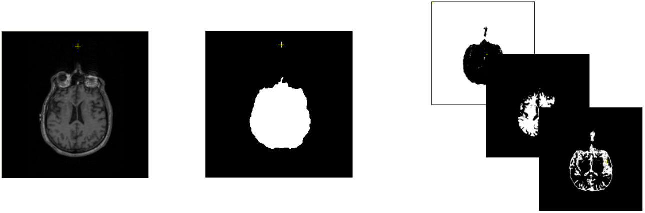
Available data:

- Anatomical MRI
- 3D location of the electrodes
- Various specific information

J.-M. Favreau and V. Barra, *Conformal Flattening of the Cortical Surface using a Topological and Geometrical Cutting*, SEECCM'09

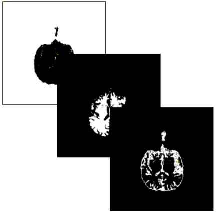
Preprocessing: from the MRI to the surface

- **Segmentation**



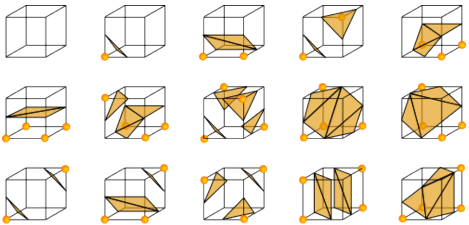
Preprocessing: from the MRI to the surface

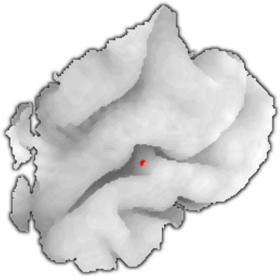
- **Segmentation**
- Computation of a brain **mask**



Preprocessing: from the MRI to the surface

- **Segmentation**
- Computation of a brain **mask**
- Computation of the **surface**

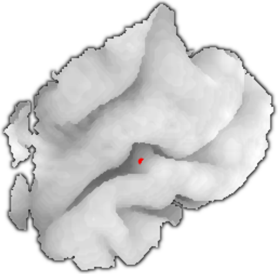


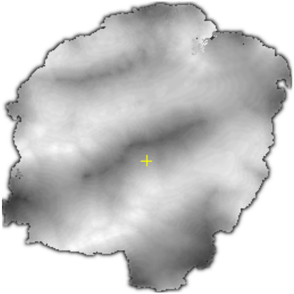


Flat map

Visulization tool: **flat map**

- Topological cutting (genus: 0, boundary: 1)
- Unfolding (conformal or quasi-conformal approaches)





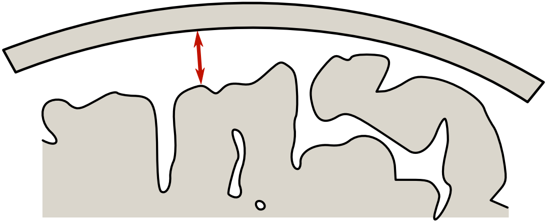
C. R. Collins and K. Stephenson, *A Circle Packing Algorithm*, Computational Geometry: Theory and Applications, 2003

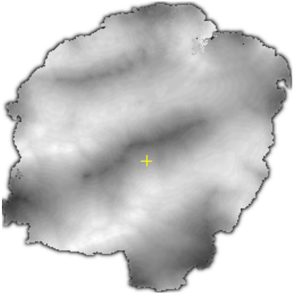
R. Zayer, B. Lévy and H.-P. Seidel, *Linear Angle Based Parameterization*, ACM/EG Symposium on Geometry Processing, 2007

Flat map

Visulization tool: **flat map**

- Topological cutting (genus: 0, boundary: 1)
- Unfolding (conformal or quasi-conformal approaches)
- **Data projection**

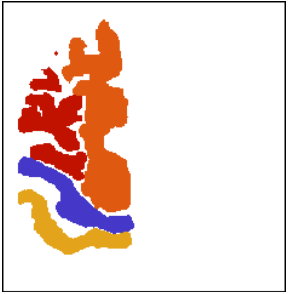


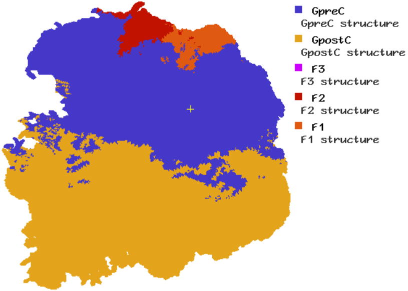


Flat map

Visulization tool: **flat map**

- Topological cutting (genus: 0, boundary: 1)
- Unfolding (conformal or quasi-conformal approaches)
- **Data projection**

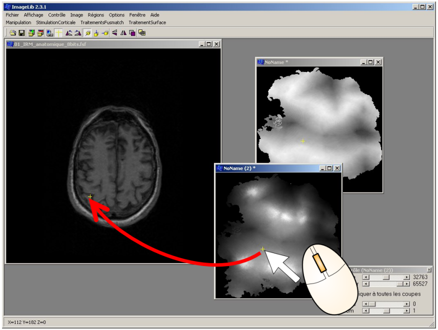




Flat map

Visulization tool: **flat map**

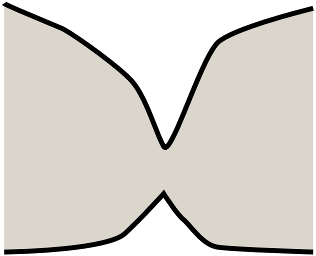
- Topological cutting (genus: 0, boundary: 1)
- Unfolding (conformal or quasi-conformal approaches)
- **Data projection**



Limitations

Topological errors:

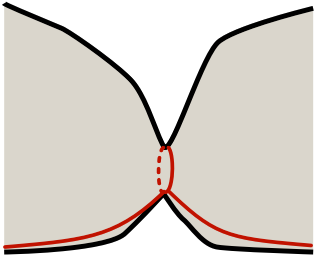
- **Partial volume** effects
- Imprecise **segmentation**



Limitations

Topological errors:

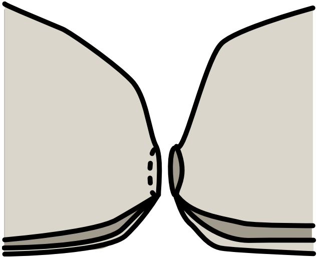
- **Partial volume** effects
- Imprecise **segmentation**



Limitations

Topological errors:

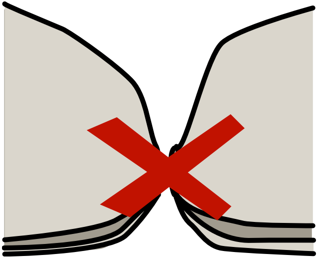
- **Partial volume** effects
- Imprecise **segmentation**



Limitations

Topological errors:

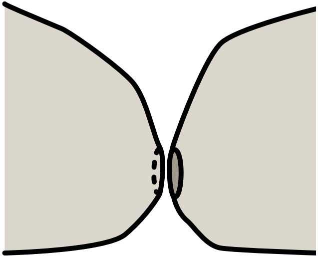
- **Partial volume** effects
- Imprecise **segmentation**



Correcting the α-junctions

After computation of the shortest loop

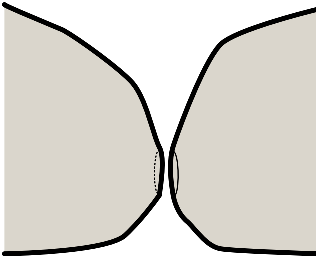
- **Cut the** 2-**mesh** according to the loop l



Correcting the α-junctions

After computation of the shortest loop

- **Cut the 2-mesh** according to the loop l
- If $length(l) < \alpha$: α-junction detection
  - Add 2 **small discs** to close the cutting



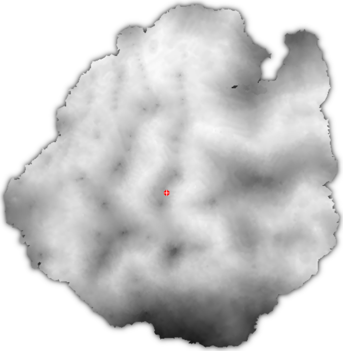
Correcting the α-junctions

Imprecise segmentation $\Rightarrow$ broken maps

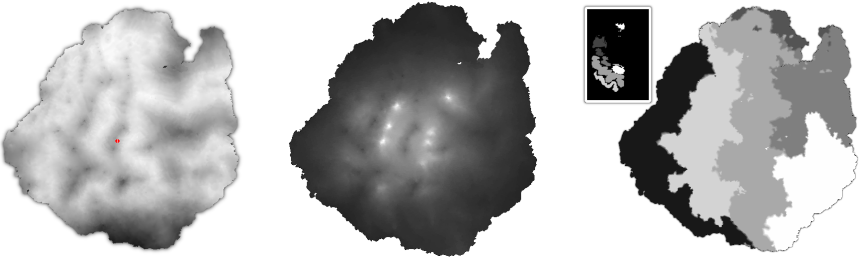
Correcting the α-junctions

Imprecise segmentation $\Rightarrow$ broken maps

Correcting the α-junctions

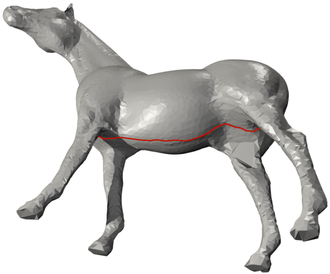


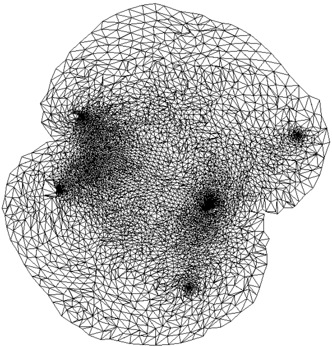
Results



Distortion as geometrical information

Great distortion of triangles:

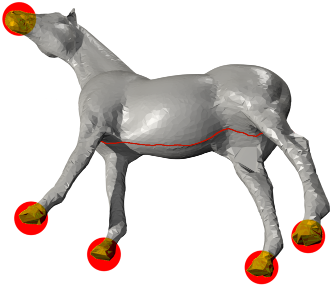


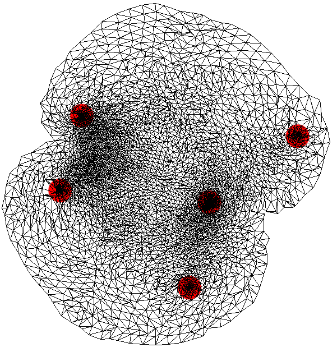


J.-M. Favreau and V. Barra, *Cutting an Organic Surface*, EuroCG, 2009

Distortion as geometrical information

Great distortion of triangles:





$\Rightarrow$ **Detect compressed areas** (protrusions)

J.-M. Favreau and V. Barra, *Cutting an Organic Surface*, EuroCG, 2009

Local distortion

- $\mathcal{M}$: 2-mesh
- Edge length in $\mathbb{R}^3$: $l_3 : E(\mathcal{M}) \rightarrow \mathbb{R}^+$
- Edge length in $\mathbb{R}^2$: $l_2 : E(\mathcal{M}) \rightarrow \mathbb{R}^+$

Local distortion

- $\mathcal{M}$: 2-mesh
- Edge length in $\mathbb{R}^3$: $l_3 : E(\mathcal{M}) \to \mathbb{R}^+$
- Edge length in $\mathbb{R}^2$: $l_2 : E(\mathcal{M}) \to \mathbb{R}^+$

Definition

The *distortion* of an edge (v_1, v_2) is defined as

$$D(v_1, v_2) = l_3(v_1, v_2)/l_2(v_1, v_2)$$

Local distortion

- $\mathcal{M}$: 2-mesh
- Edge length in $\mathbb{R}^3$: $l_3 : E(\mathcal{M}) \rightarrow \mathbb{R}^+$
- Edge length in $\mathbb{R}^2$: $l_2 : E(\mathcal{M}) \rightarrow \mathbb{R}^+$

Definition

The *distortion* of an edge (v_1, v_2) is defined as

$$D(v_1, v_2) = l_3(v_1, v_2)/l_2(v_1, v_2)$$

Definition

The *distortion* of an vertex v is defined as

$$D(v) = \frac{1}{|(v, v_i) \in E(\mathcal{M})|} \sum_{(v,v_i) \in E(\mathcal{M})} D(v, v_i)$$

Extremum definition

- $\mathcal{M}$: 2-mesh and B its boundary
- Using $D(\cdot,\cdot)$ in **Dijkstra algorithm** from B

$\Rightarrow$ defining a distance function: L_b

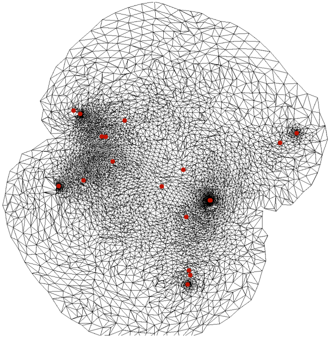
Extremum definition

- $\mathcal{M}$: 2-mesh and B its boundary
- Using $D(\cdot, \cdot)$ in **Dijkstra algorithm** from B

$\Rightarrow$ defining a distance function: L_b

Definition

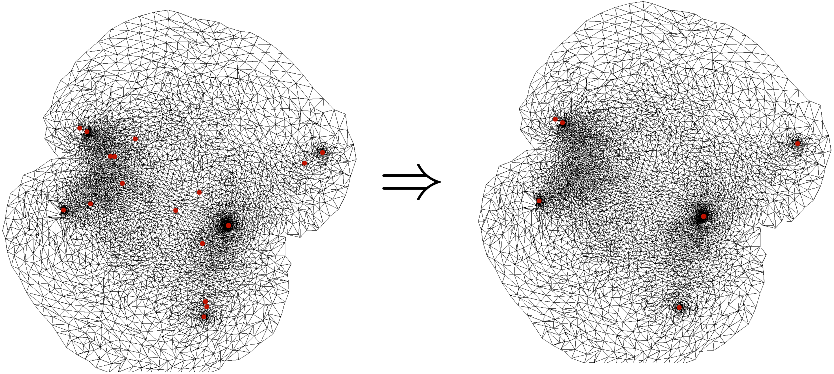
A *extremum* vertex v is a local maximum of L_b



Extremum filtering

For graphical applications: **too many extrema**

- Ordering extrema using $D(\cdot)$
- Keep **only high distortion vertices** (mean, median, manual, etc.)



Geometrical cutting

Data:

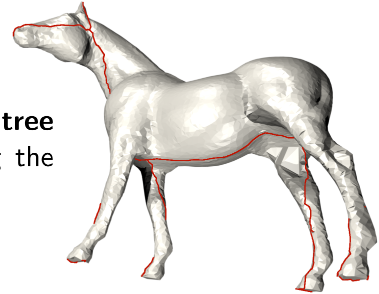
- A 2-mesh $\mathcal{M}$ homeomorphic to a disc
- A set of extrema $(m_i)_{0 \leq i < n}$

Geometrical cutting

Data:

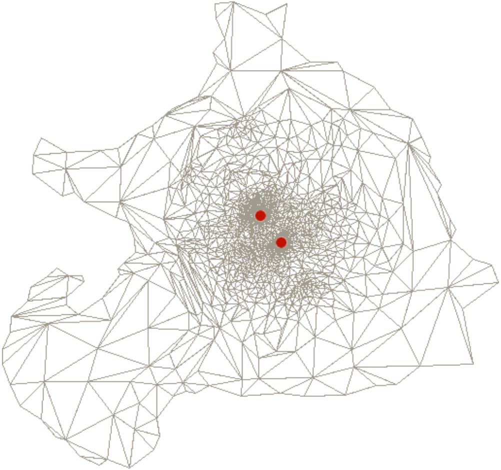
- A 2-mesh $\mathcal{M}$ homeomorphic to a disc
- A set of extrema $(m_i)_{0 \leq i < n}$

Cut accorting to the **shortest spanning tree** joining $(m_i)_{0 \leq i < n}$ and the boundary using the **geodesic distance**



Step by step example

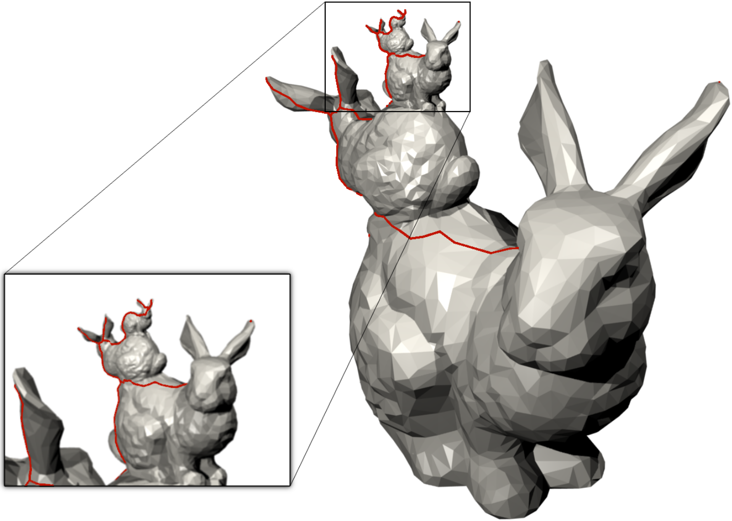
Step by step example



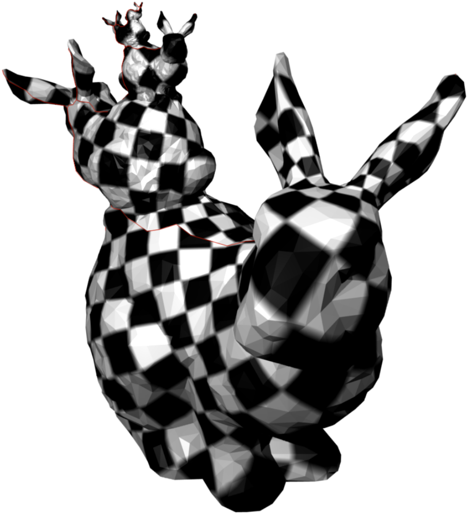
Number of extrema: 19

Step by step example

Step by step example

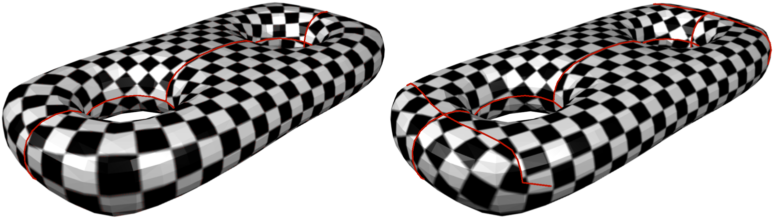


Step by step example



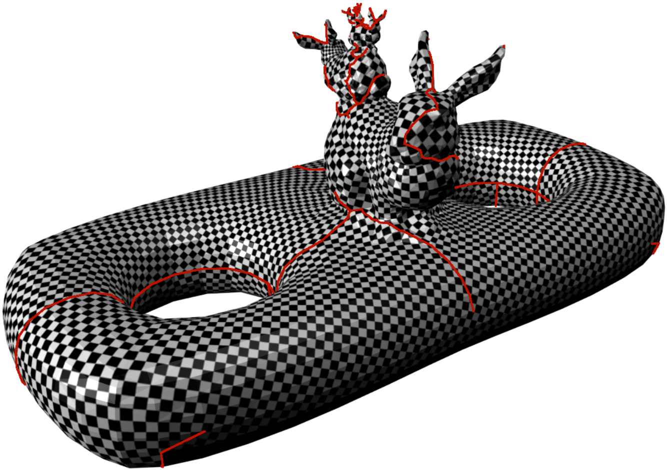
Application: texture mapping

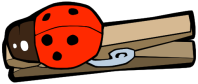
Texture regularity



Application: texture mapping

The multiscale rabbit on a double torus

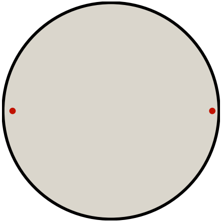




Cylinders

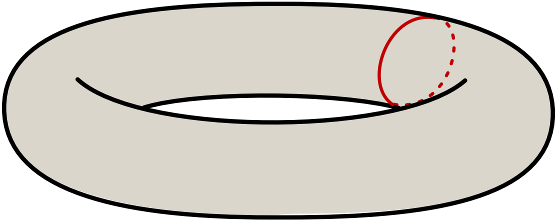
Genus: 0, boundaries: 2

Cylinders

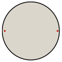


Genus: 1, boundaries: 0

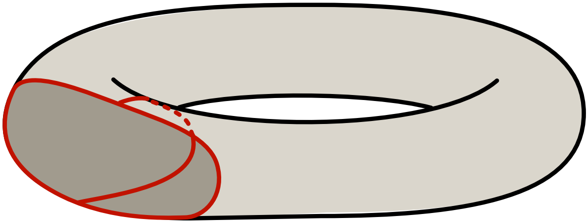
Cylinders



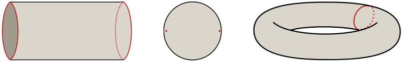
Genus: 1, boundaries: 0



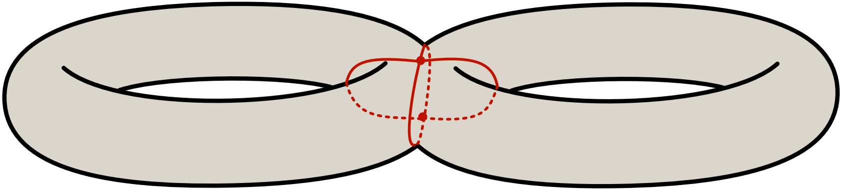
Cylinders



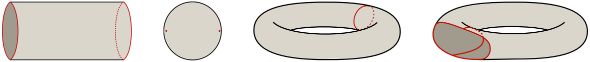
Genus: 1, boundaries: 1



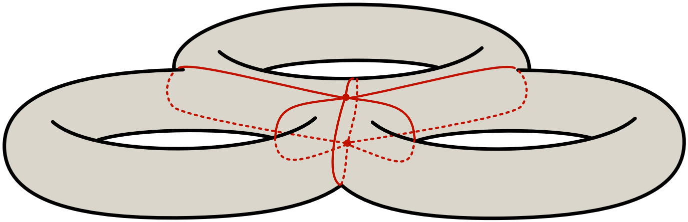
Cylinders



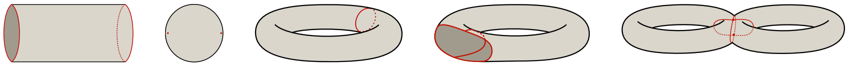
Genus: 2, boundaries: 0



Cylinders



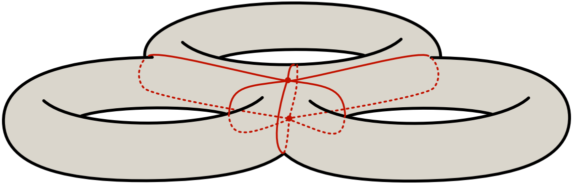
Genus: 3, boundaries: 0



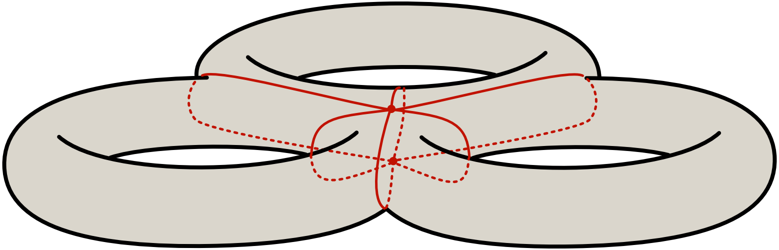
n-loops: definition

Definition (*n*-loop on a 2-mesh $\mathcal{M}$)

- Two **base points** b_0 and b_1
- n non-homotopic **paths** joining b_0 and b_1

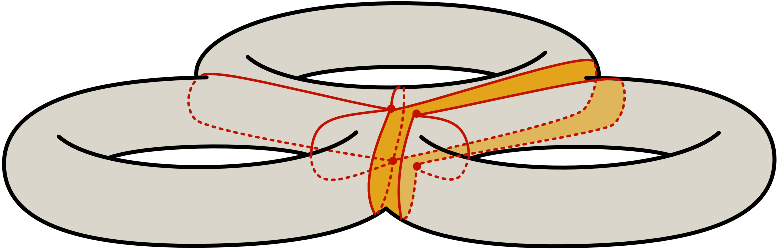


From *n*-loop to 3-loops



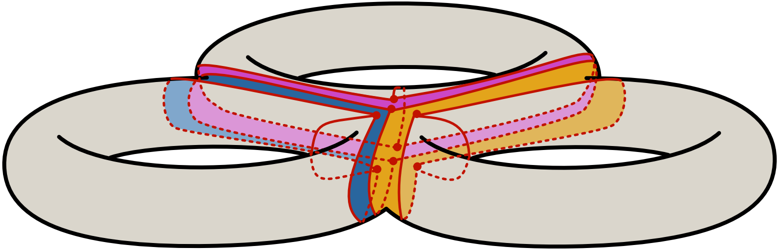
three cylinders, one 6-**loop**

From *n*-loop to 3-loops



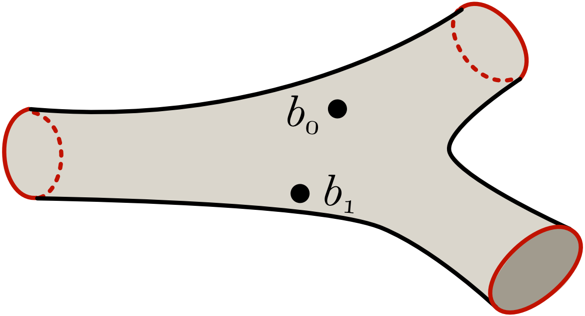
four cylinders, one 5-**loop**, one 3-**loop**

From *n*-loop to 3-loops

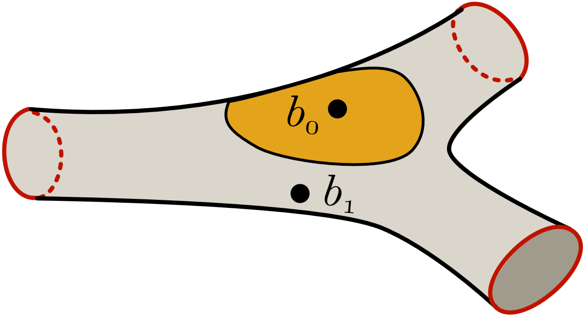


six cylinders, four 3-**loop**

Approximation of minimal 3-loop

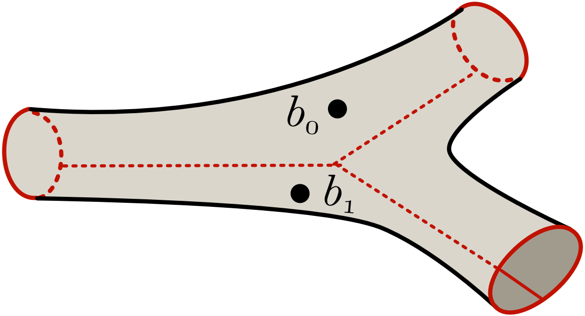


Approximation of minimal 3-loop



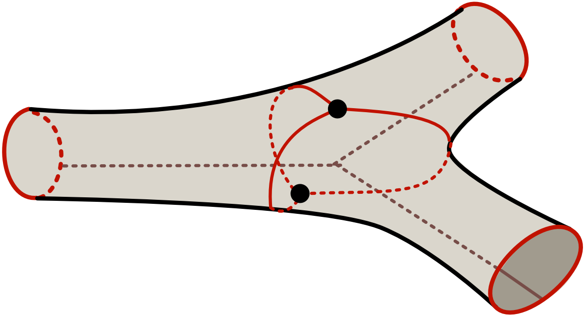
Approximation of minimal 3-loop

Computation using the reduced **cut locus** (Voronoï cell)



Approximation of minimal 3-loop

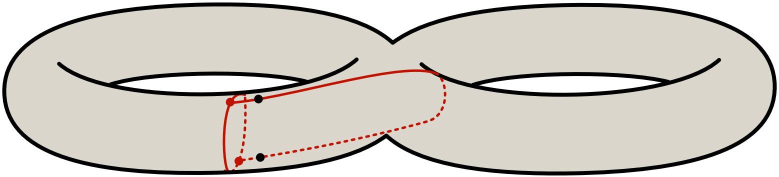
Computation using the reduced **cut locus** (Voronoï cell)



Adjustment by optimization

Goal: segmentation by cylinders using 3-loops

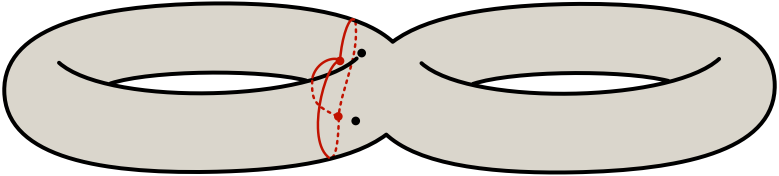
- Principle: **optimize** each 3-loop
- Energy function: **length** of the paths describing the 3-loop



Adjustment by optimization

Goal: segmentation by cylinders using 3-loops

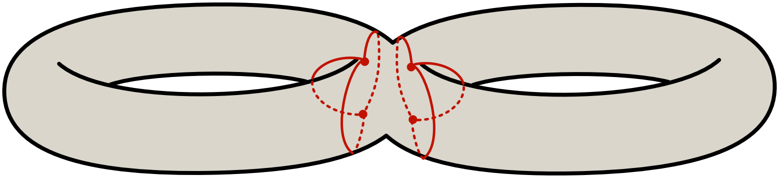
- Principle: **optimize** each 3-loop
- Energy function: **length** of the paths describing the 3-loop



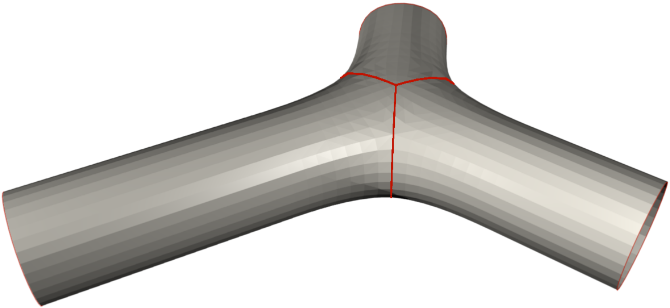
Adjustment by optimization

Goal: segmentation by cylinders using 3-loops

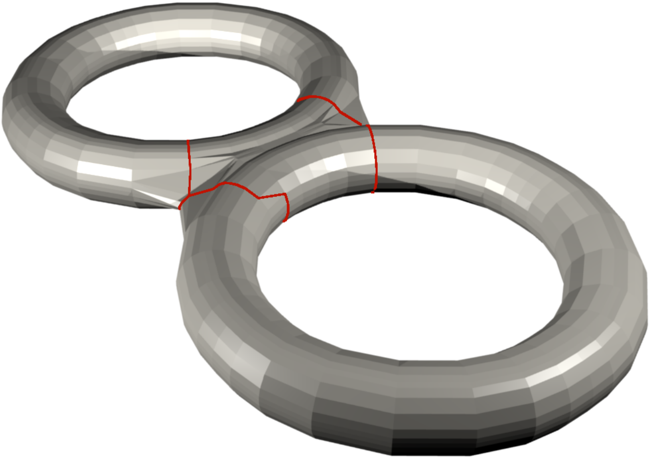
- Principle: **optimize** each 3-loop
- Energy function: **length** of the paths describing the 3-loop



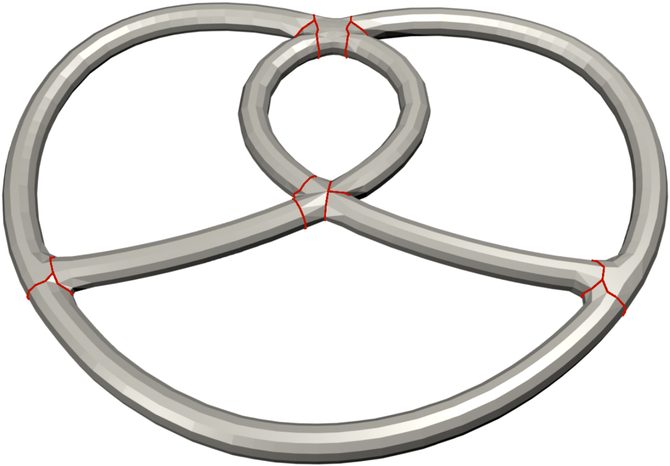
Cylinders from 3-loops: results



Cylinders from 3-loops: results

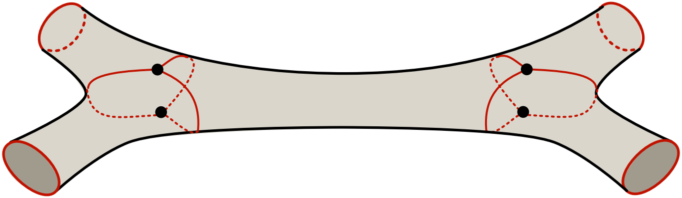


Cylinders from 3-loops: results



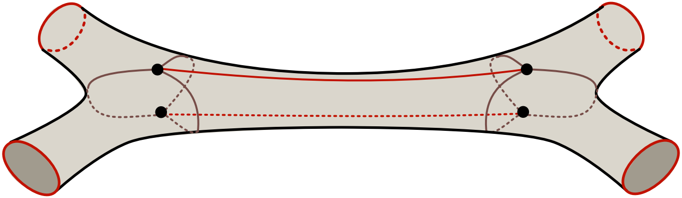
Quadrangulation

Cutting into cylinders by n-loops: tiles with **2 vertices on each boundary**



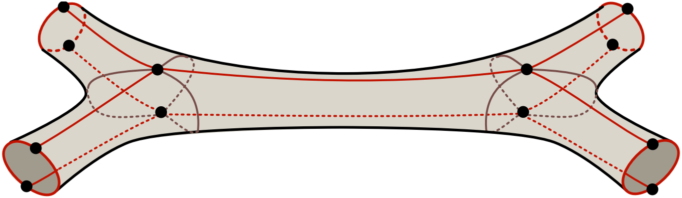
Quadrangulation

Cutting into cylinders by *n*-loops: tiles with **2 vertices on each boundary**



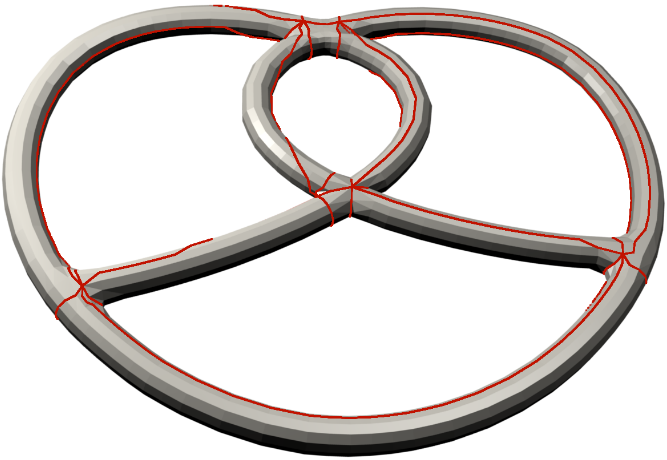
Quadrangulation

Cutting into cylinders by *n*-loops: tiles with **2 vertices on each boundary**



- Description using **Bézier patches**
- Easy **texture** mapping

Quadrangulation: results



Semantic segmentation

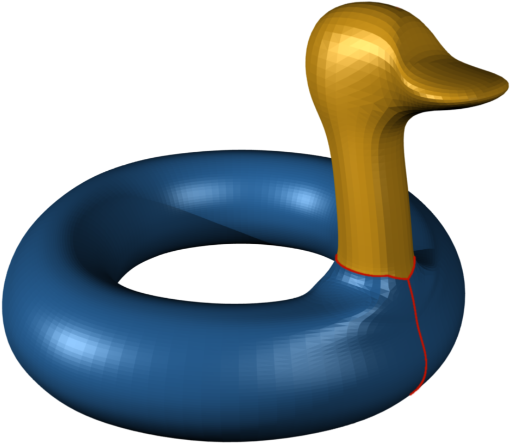
- Removing the detected **extrema**

Semantic segmentation

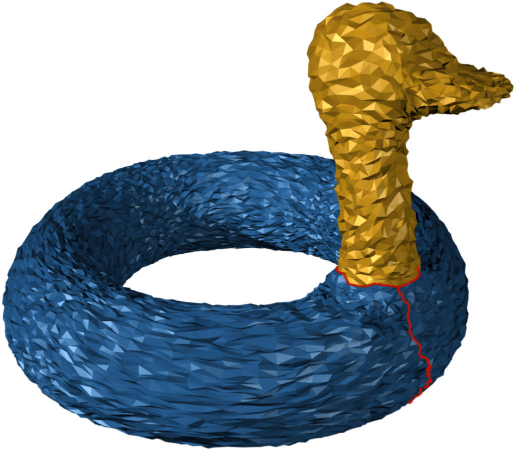
- Removing the detected **extrema**
- Perform a **cutting into cylinders**

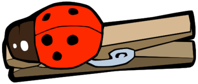
Semantic segmentation: results

Semantic segmentation: results



Semantic segmentation: results

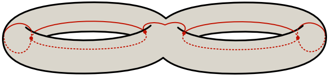
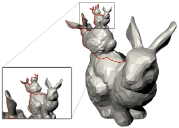
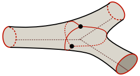




Conclusion

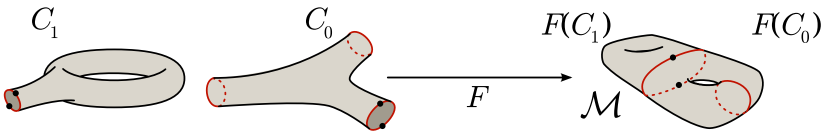
Cutting / tiling:

- Exploit the **topology**
- Integrate the **geometry**
- Supervise the **topology of the tiles**

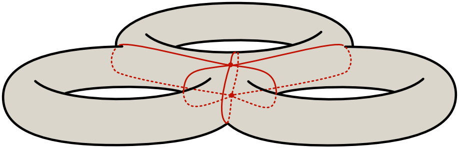

Future

- Expand the using of **M-tilings**



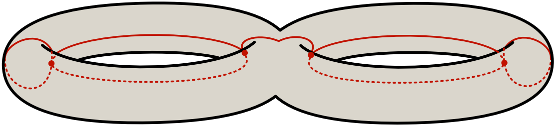
Future

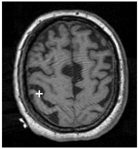
- Expand the using of **M-tilings**
- Explore possibilities of the *n*-**loops**

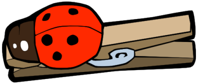


Future

- Expand the using of **M-tilings**
- Explore possibilities of the *n*-**loops**
- Work on the **applications** allowed by **computational topology**







6 Découpage topologique: détails
- Plus court lacet non-séparant
- Améliorations

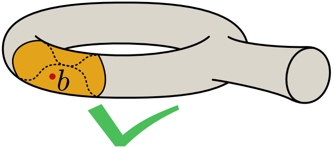
7 Implementation
- TAGLUT

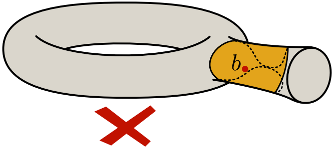
8 Point clouds
- Geometry

Shortest non-separating loop (1)

Given a **basepoint** b

- Using a **"geodesic" distance** (e.g. Dijkstra[1])
- Circular **wavefront propagation**
- Catching the **junctions** (and their nature)





Complexité: $O(n \log n)$

Dijkstra, A note on two problems in connexion with graphs. *Numerische Mathematik*, 1959.

[1] 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)

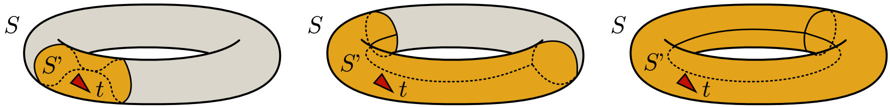
Principle: a set **crossed by all the non-separating loops**

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 $\mathcal{M}'$

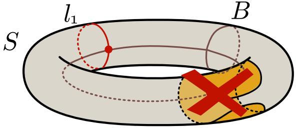
end

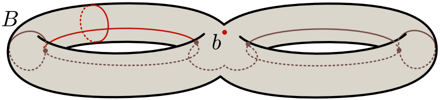


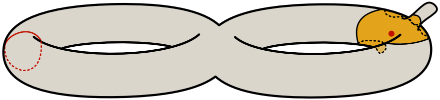
Improvements of the Cutting Algorithm

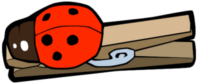
Improvements of the **shortest loop computation**:

- Bounded Dijkstra (using previous cuts)
- Ordering basepoints
- Use minimal length









6 Découpage topologique: détails
- Plus court lacet non-séparant
- Améliorations

7 Implementation
- TAGLUT

8 Point clouds
- Geometry

TAGLUT

Topological And Geometrical Library : a Useful Toolkit

Informations:

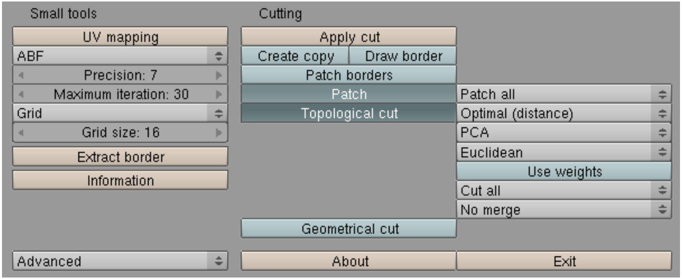
- License: GPL v2 (`http://jmfavreau.info/?q=en/taglut`)
- Languages: `C++`, `python` (et `xml`)
- Supported formats: `vrml`, `ply`, `obj`, etc.

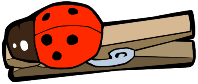
Functionalities:

- Cutting **algorithms** (topology & geometry)
- **Display** and **input/output** functionalities
- Binding with **blender**
- **Scriptable** with python

TAGLUT

blender interface





6 Découpage topologique: détails
- Plus court lacet non-séparant
- Améliorations

7 Implementation
- TAGLUT

8 Point clouds
- Geometry

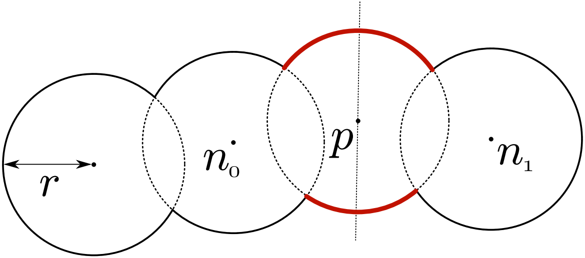
Normals' computation

Initial data:

- A sequence of **2D point clouds**
- **Rigid reconstruction** on subsequences
- A sequence of **3D point clouds**

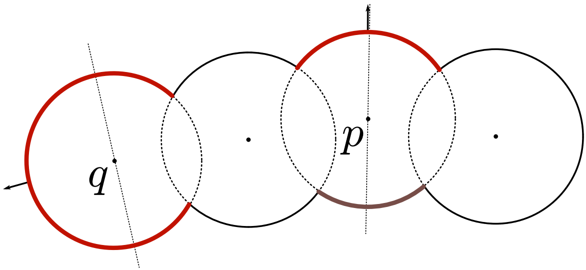
Normals' estimation

- **offset** approach



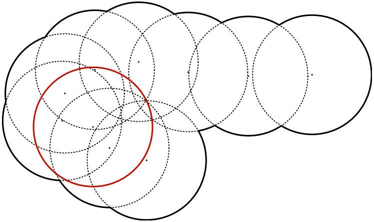
Normals' estimation

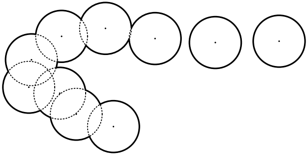
- **offset** approach
- Detect the **boundary property**



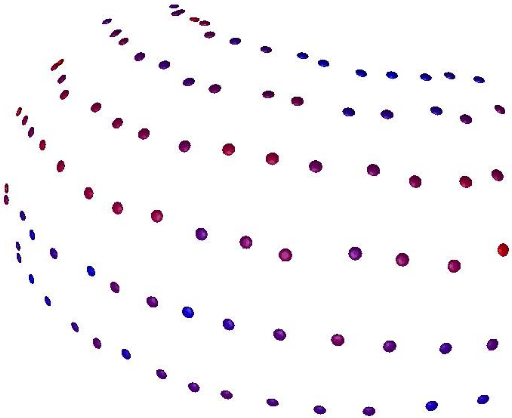
Normals' estimation

- **offset** approach
- Detect the **boundary property**
- **Local adjustment** of the radii

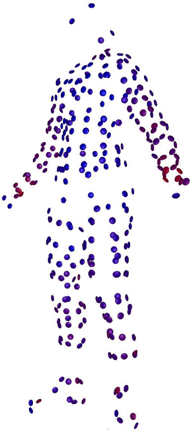




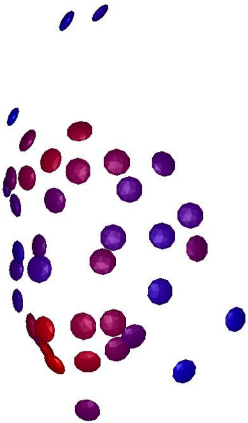
Normals' estimation: results



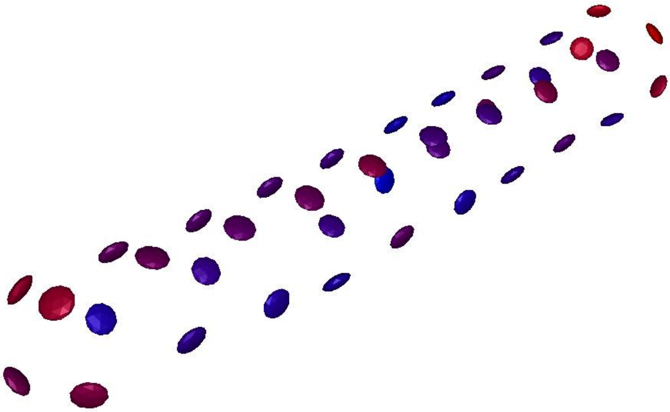
Normals' estimation: results



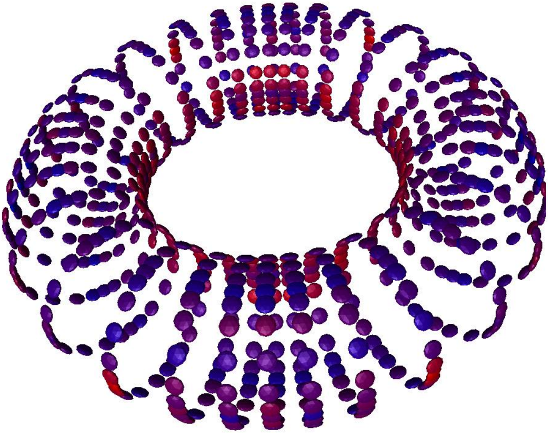
Normals' estimation: results



Normals' estimation: results



Normals' estimation: results



Normals' estimation: results