A Discrete Approach to Reeb Graph Computation and Surface Mesh Segmentation: Theory and Algorithm

Laura Brandolini

Department of Industrial and Information Engineering

Computer Vision & Multimedia Lab

Tutor:

Prof. Marco Piastra

Prof. Virginio Cantoni

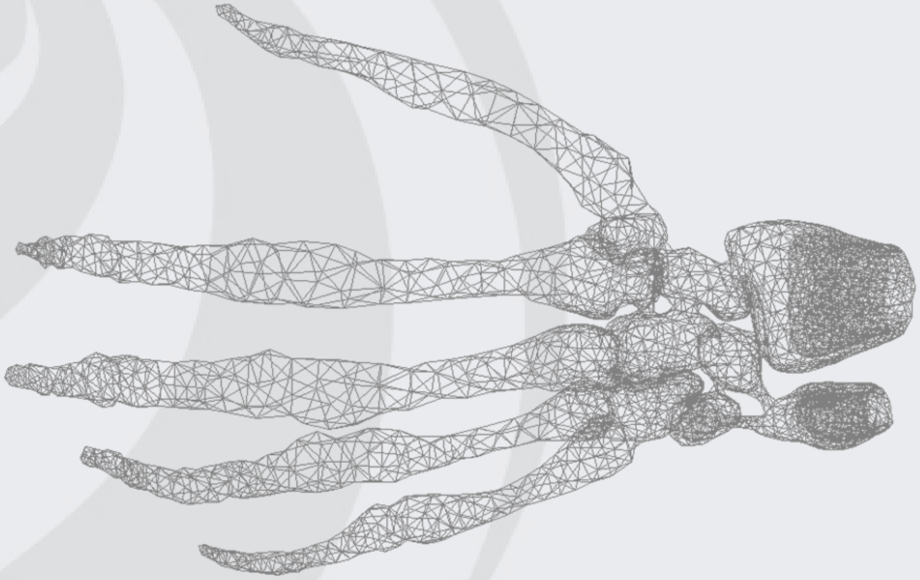
CVMLab

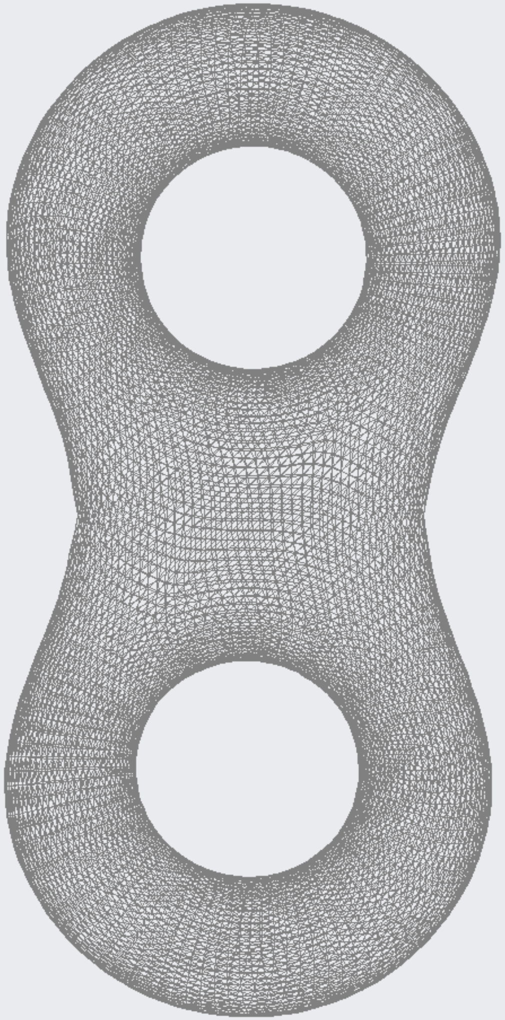
Computer Vision & Multimedia Lab

Università degli Studi di Pavia

Stating the problem

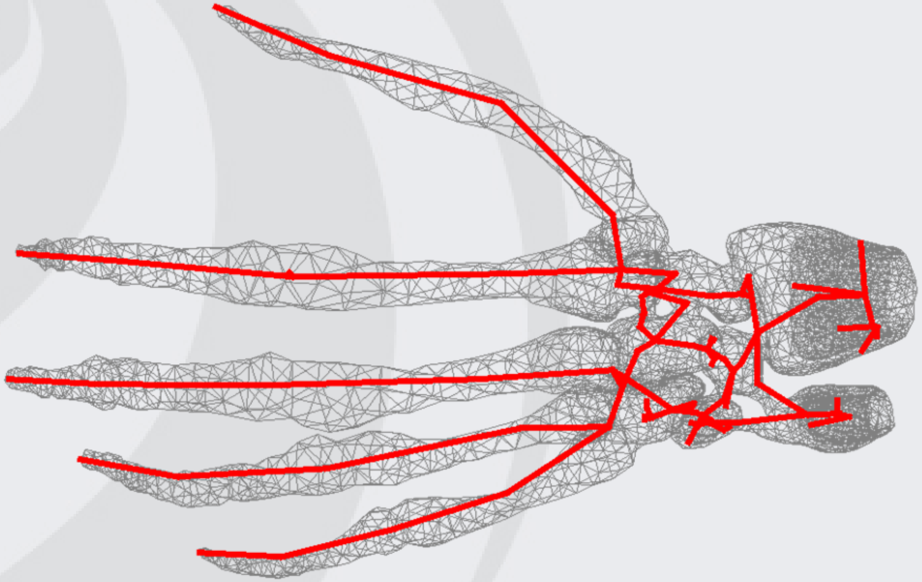
Given a closed and triangulated mesh

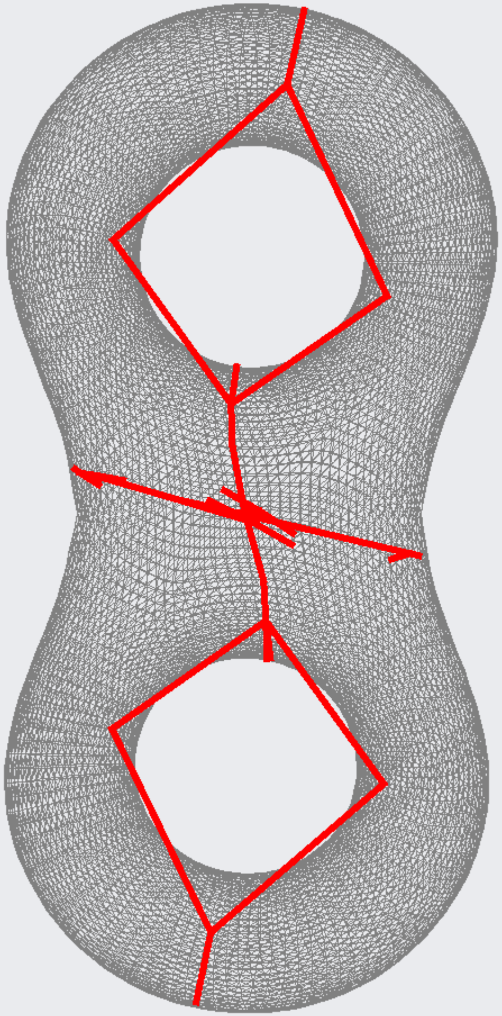




Stating the problem

Given a closed and triangulated mesh we want a compact *graph* representation of its shape





What is the purpose?

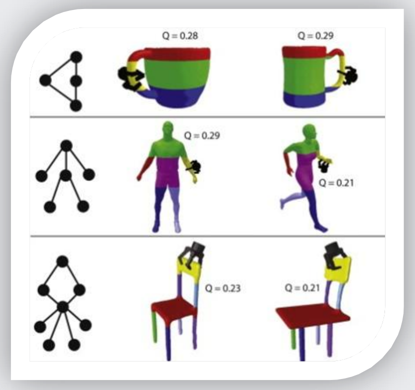
[L Antiga 2002]

- **Augmenting** shape representations

  (e.g. in order to detect *thinnings* and *thickenings*)

- As an **alternative**, compact shape representation

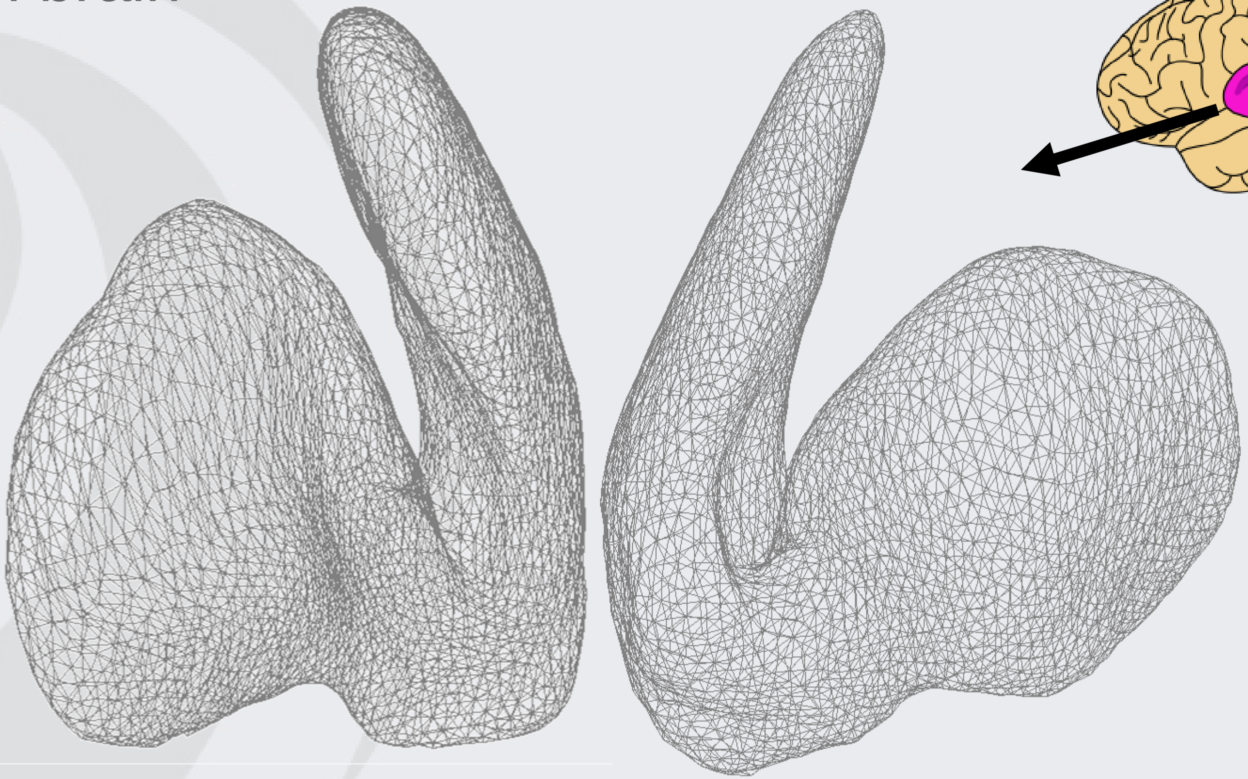
  (e.g. shape matching)

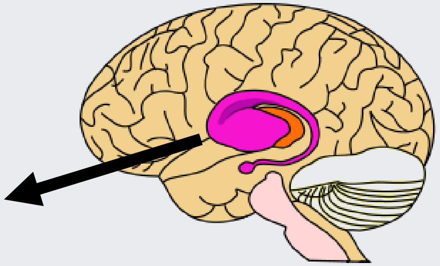


[Aleotti Caselli 2012]

An application: striatal mesh descriptor

Meshes extracted from 3D medical images (MRI) of human brain

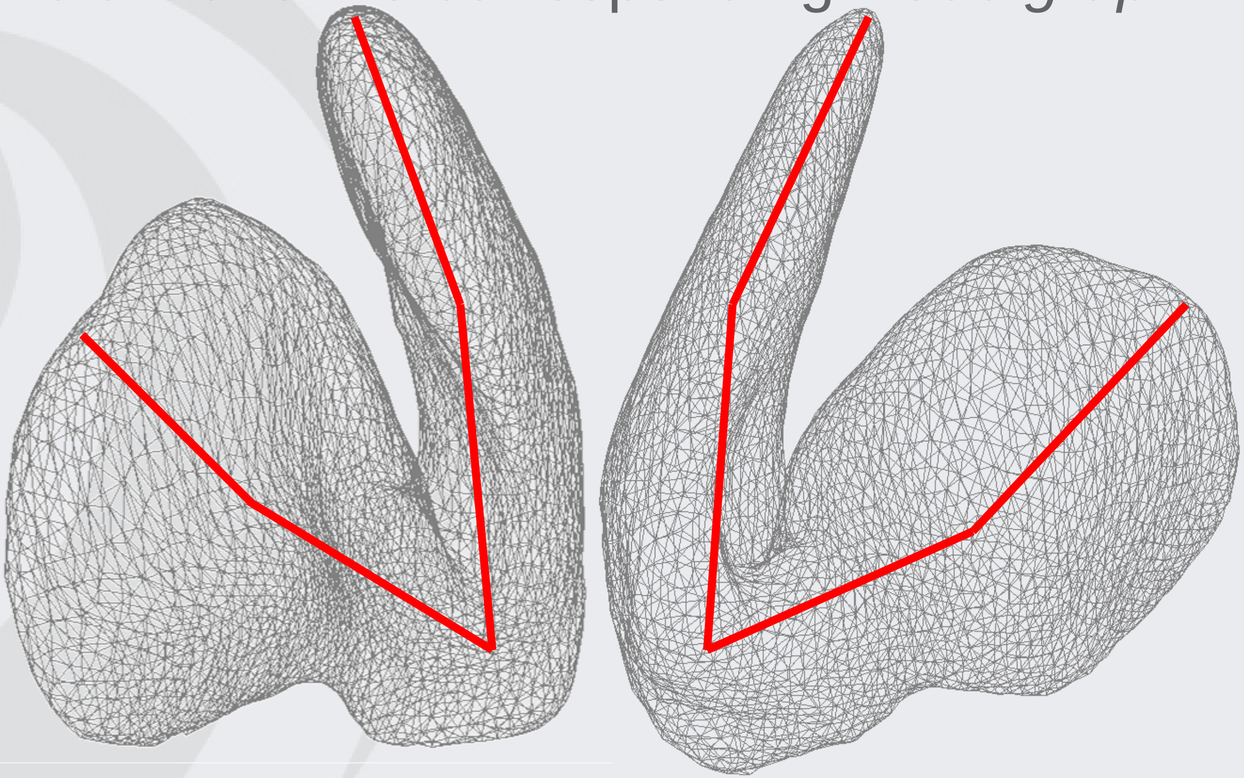




left striatum right striatum

An application: striatal mesh descriptor

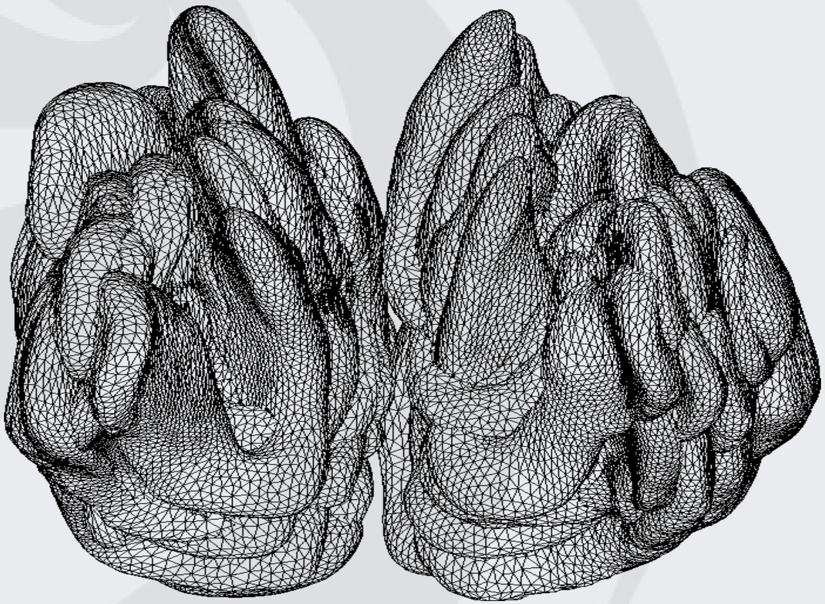
Meshes extracted from 3D medical images (MRI) of human brain and the corresponding *Reeb graph*

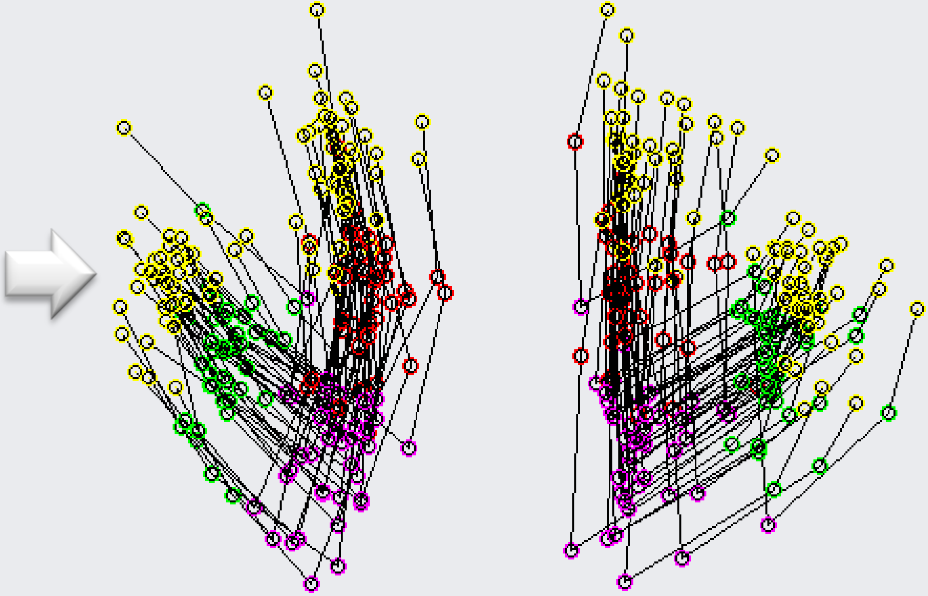


left striatum right striatum

An application: striatal mesh descriptor

40 different subjects: meshes extracted from 3D images (MRI) and the corresponding Reeb graphs



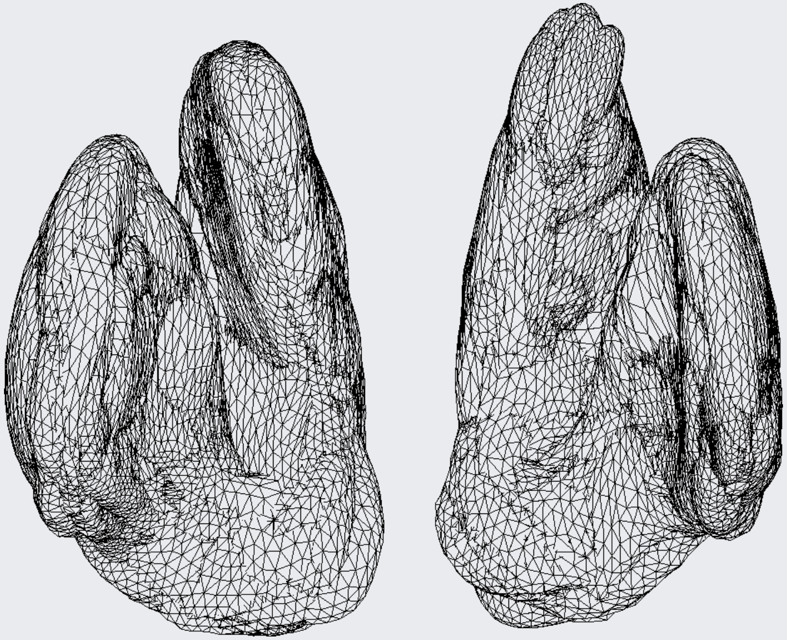


Simplified Reeb Graph as Effective Shape Descriptor for the Striatum - MICCAI 2012 MeshMed workshop, 2012, Nice, France
Tampere University of Technology, VTT Technical Research Centre of Finland, University of Turku Finland

An application: striatal mesh descriptor

Registering *graphs* instead of entire meshes

each graph: 5 vertices

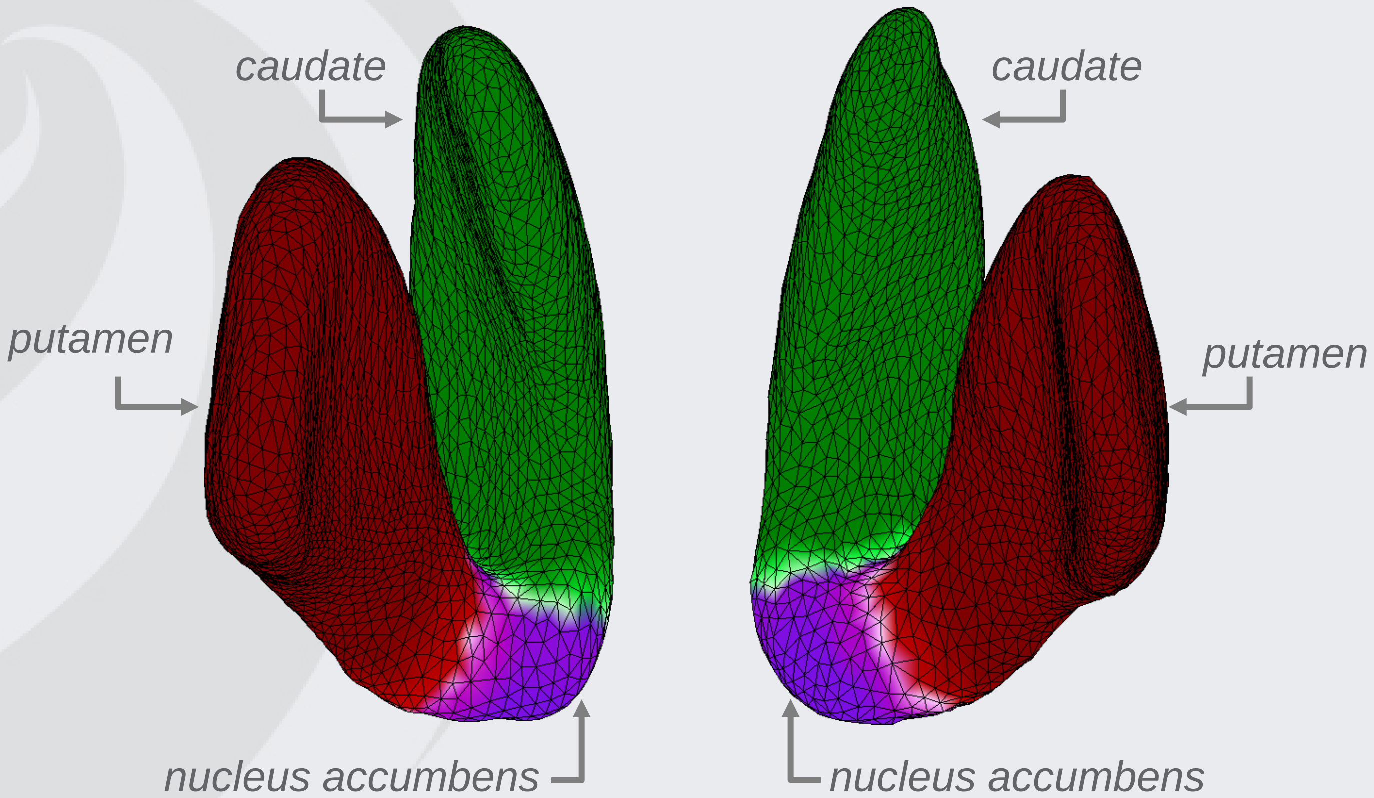


each mesh: 3335 vertices

Simplified Reeb Graph as Effective Shape Descriptor for the Striatum - MICCAI 2012 MeshMed workshop, 2012, Nice, France
Tampere University of Technology, VTT Technical Research Centre of Finland, University of Turku Finland

An application: segmentation

The human striatum has three main parts



Simplified Reeb Graph as Effective Shape Descriptor for the Striatum - MICCAI 2012 MeshMed workshop, 2012, Nice, France
Tampere University of Technology, VTT Technical Research Centre of Finland, University of Turku Finland

Reeb graphs: state of the art

Reeb graph: a *faithful* and compact representation of a shape (see after)

There are many algorithms for computing Reeb graphs:

Cole-McLaughlin, Edelsbrunner, Harer, Natarajan, Pascucci 2003:
Correct Reeb graph computation but
but no esplicit embedding or segmentation and
requires global sorting of mesh vertices and local remeshing

Tierny, Vandeborre, Daudi 2006:
Efficient but no guarantee of correctness

Pascucci, Scorzelli, Bremer, Mascarenhas 2007:
Correct and robust on-line computation but without segmentation

Patanè, Spagnuolo, Falcidieno, Biasotti 2009:
Correct Reeb graph and segmentation but requires remeshing

Reeb graphs: representative of the shape

In what sense a graph is a *faithful* representation of a shape?

Smooth surfaces

Morse theory

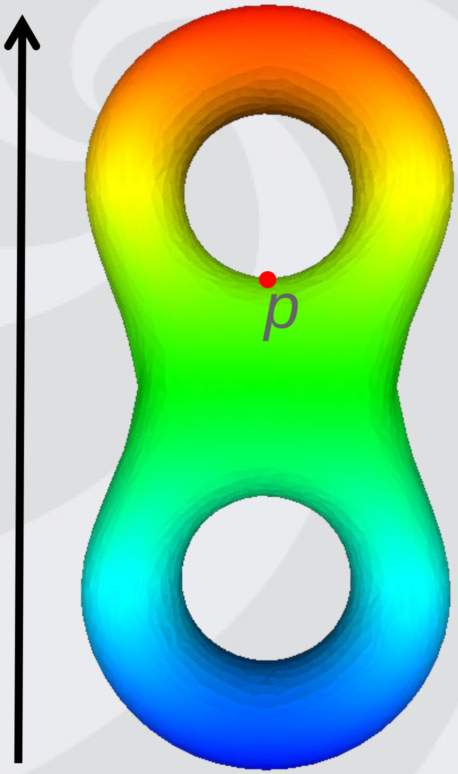
↓

Reeb graphs

Morse, 1931
Milnor, 1963

Reeb, 1946

Morse function on smooth surfaces



A **height function**
is a Morse function

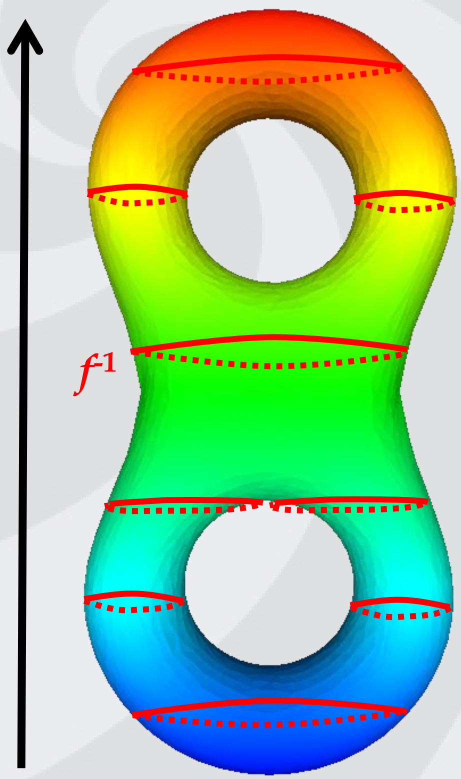
f is a ***smooth scalar*** *function defined on a* ***smooth surface*** M.

A point $p \in M$ *is* ***critical*** *for* f
if f *all first partial derivatives are zero in p.*

A critical point p is said to be ***non-degenerate***
if the Hessian matrix (matrix of second partial derivatives) of f *is non-singular at p*
(determinant not zero in p).

f *is called a* ***Morse function***
if *<u>all its critical points are non-degenerate</u>*.

Morse function on smooth surfaces



A level line $f^{-1}(\alpha)$ is the set of points of M *having the same value* α *of* f.

Morse theory:

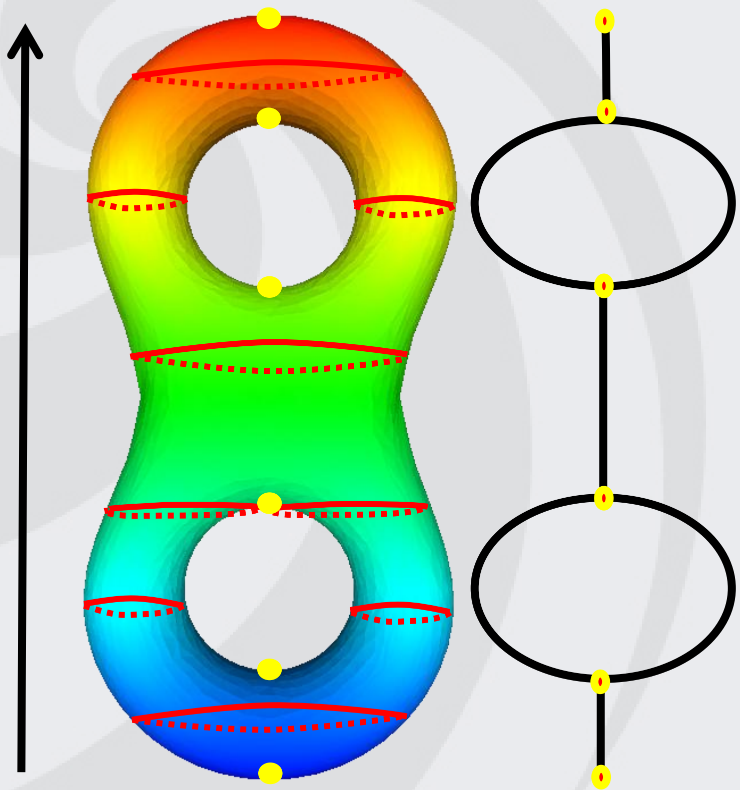
*- the critical points of a Morse function are **finite***

*- all level lines are the union of a **finite set** of connected components*

*- connected components become **points** at **maxima** and **minima***

*- connected components **intersect** at saddle points*

Reeb graphs: on smooth surfaces



***Equivalence relation**:*
two points on M are equivalent if they belong to the same connected component of a level line.

*Each point of the **Reeb graph** corresponds to a connected component of a level line:*

*- the **nodes** of the Reeb graph correspond to the **critical values** of f.*

*- **all other points** in the graph correspond to **non-critical values.***

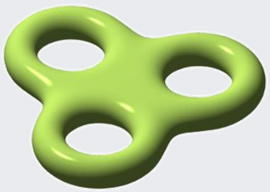
Genus and Loop lemma

The genus of a surface

*The **genus** of a closed surface equals the number of '**holes**' through it*

genus = 1

genus = 2



genus = 3

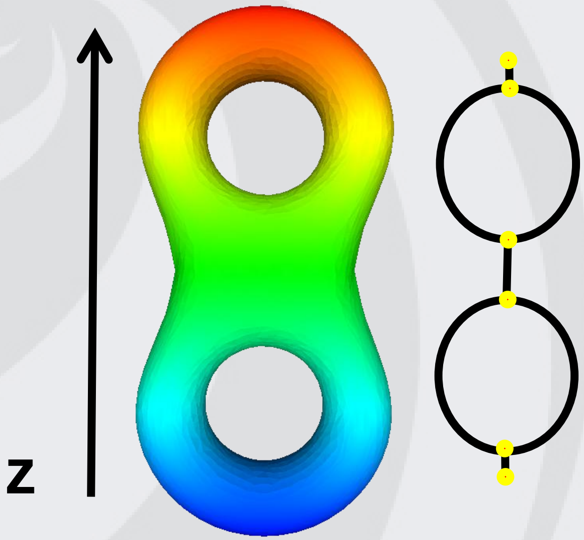
Fundamental property

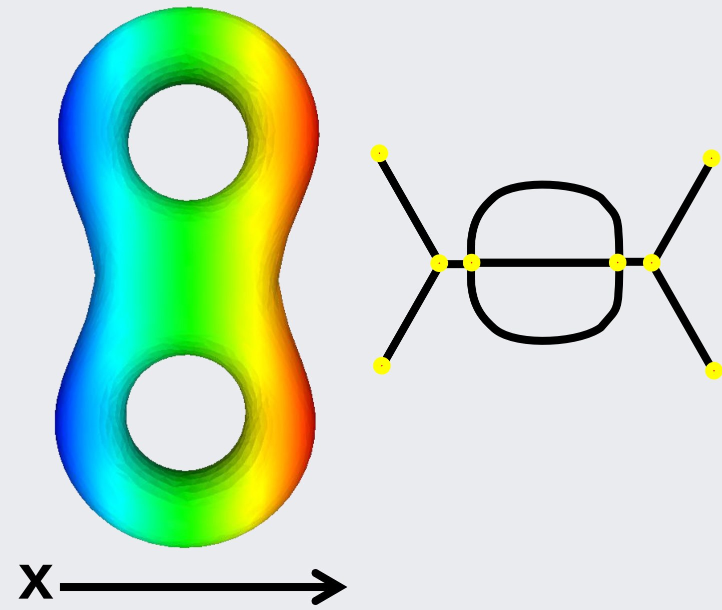
*for orientable, closed, smooth, connected **surfaces,***
***if** f is Morse, the **number of loops** in the Reeb graph corresponds to the **genus** of the surface*

[Cole-McLaughlin et al., 2003]

The Reeb graph is not unique

The actual Reeb graph depends on f



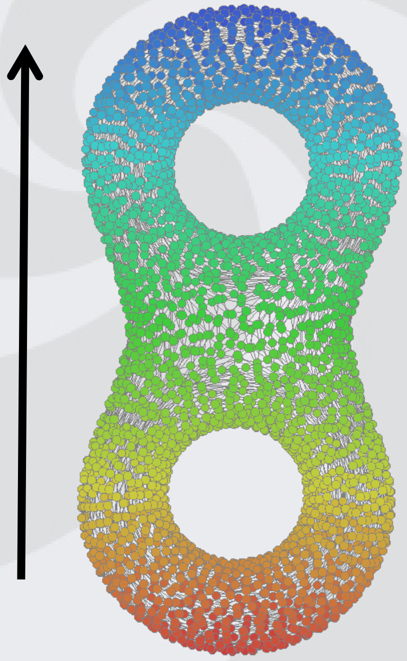


(but the number of loops in the Reeb Graph does not vary)

Reeb graphs on triangulated surfaces

Smooth surfaces	Triangulated surfaces
Morse theory	~Morse theory
Reeb graphs	~Reeb graphs

f on triangulated surfaces



Given a closed, orientable, connected, ***triangulated surface*** X.

f is a function defined ***on the vertices*** *of the triangulation* X.

f is ***general*** *if*

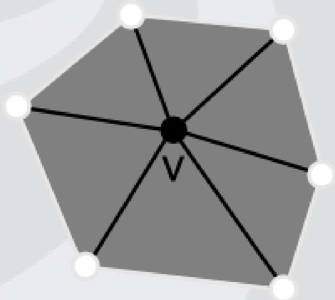
it takes ***different values*** *at* ***each vertex***.

From now on we assume f

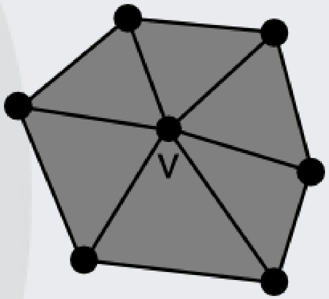
to be ***general***

Triangulated surfaces

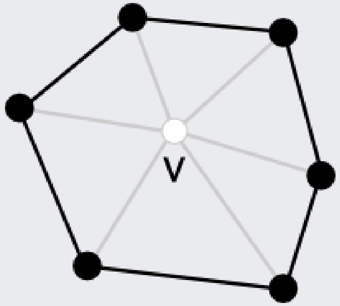
The neighborhood of a vertex in a triangulated surface



Star



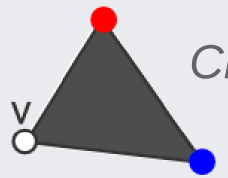
Closure of the star



Link

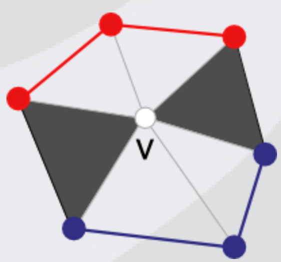
Critical points

- *higher vertex*
- *lower vertex*

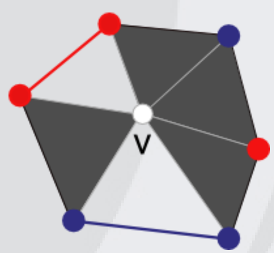


Cross-triangle for $f(v)$

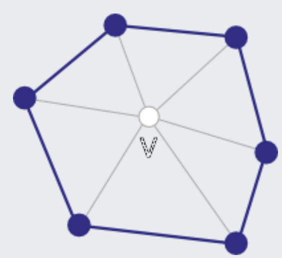
Index of a vertex: $$i(v) = 1 - \frac{1}{2} t_c$$



regular:
i=0

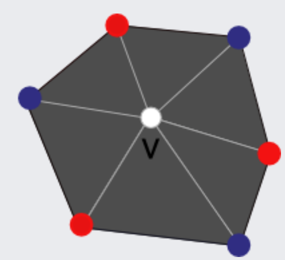


saddle:
i=-1



max
i=1

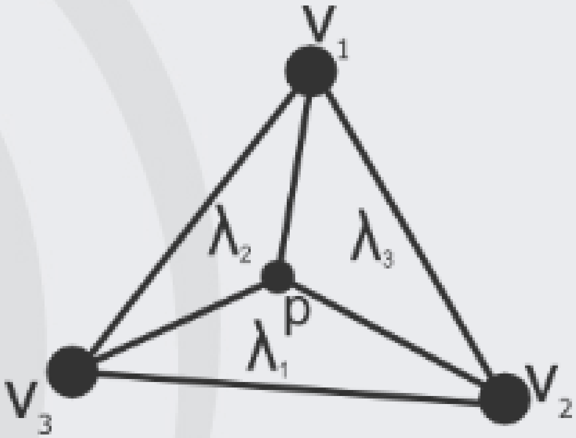
min
i=1



multiple saddle
i<=-2

Barycentric coordinates

f is defined on the **vertices**



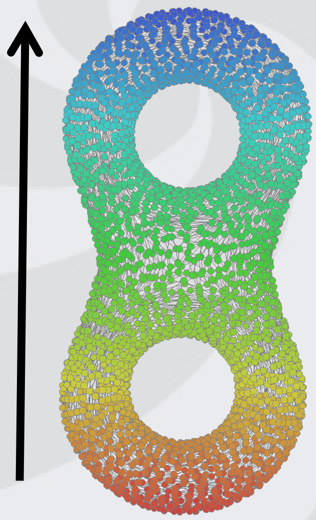
$$p = (\lambda_1, \lambda_2, \lambda_3)$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

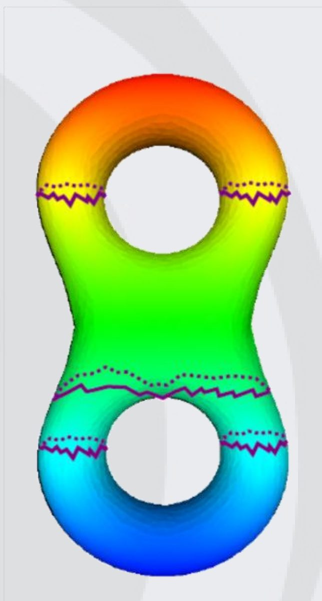
f^* is defined on the entire surface X

$$f^*(p) := \lambda_1 f(v_1) + \lambda_2 f(v_2) + \lambda_3 f(v_3)$$

f and f^* on triangulated surfaces



f



f^*

A level line of f^* at α is
the ***set of points*** *of the surface X*
having the ***same value*** α of f^*.

Each level line is the union of
a finite set
of connected components
called ***contours****.*

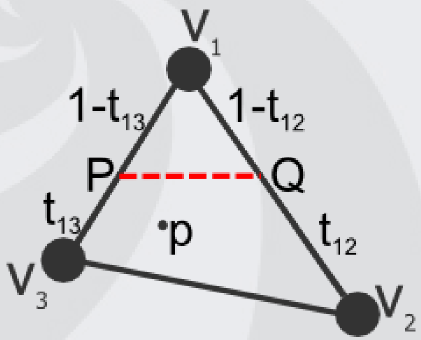
Each contour is the union
of a set of ***straight segments.***

These ***segments****, on a same face,*
are ***parallel to each other****,*
for different α values

Level lines: straight and parallel

level lines f^{*-1} are the ***union of straight segments***

level lines f^{*-1} are ***parallel***



$$P = (t_{13}, 0, 1 - t_{13})$$

$$Q = (t_{12}, 1 - t_{12}, 0)$$

$$f^*(P) = f^*(Q) = l$$

$$\begin{aligned} t_{13} f(v_1) + (1 - t_{13}) f(v_3) &= f^*(P) \\ t_{12} f(v_1) + (1 - t_{12}) f(v_2) &= f^*(Q) \end{aligned}$$

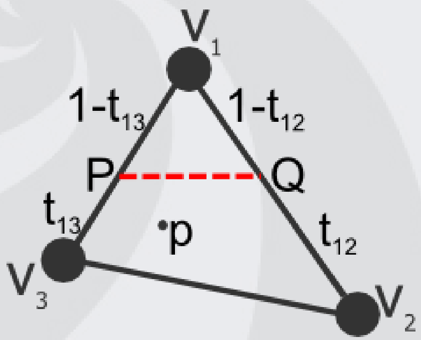
$$t_{12} = \frac{l - f(v_2)}{f(v_1) - f(v_2)}$$

$$t_{13} = \frac{l - f(v_3)}{f(v_1) - f(v_3)}$$

Level lines: straight and parallel

*level lines f^{*-1} are the **union of straight segments***

*level lines f^{*-1} are **parallel***



$$P = (t_{13}, 0, 1 - t_{13})$$
$$Q = (t_{12}, 1 - t_{12}, 0)$$

$$\begin{vmatrix} t_{12} & (1 - t_{12}) & 0 \\ t_{13} & 0 & (1 - t_{13}) \\ \lambda_1 & \lambda_2 & \lambda_3 \end{vmatrix} = 0$$

http://mathworld.wolfram.com/BarycentricCoordinates.html

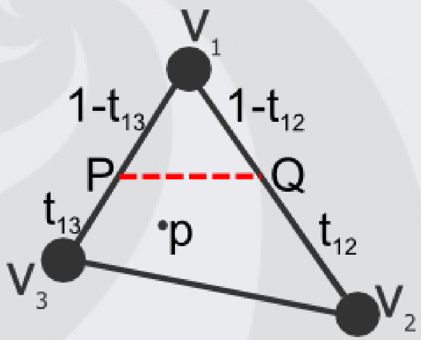
$$\lambda_1(1 - t_{12})(1 - t_{13}) - \lambda_2 t_{12}(1 - t_{13}) - \lambda_3 t_{13}(1 - t_{12}) = 0$$

$$\lambda_1 - \lambda_2 \frac{t_{12}}{(1 - t_{12})} - \lambda_3 \frac{t_{13}}{(1 - t_{13})} = 0$$

Level lines: straight and parallel

*level lines f^{*-1} are the **union of straight segments***

*level lines f^{*-1} are **parallel***



$$\lambda_1 - \lambda_2 \frac{t_{12}}{(1-t_{12})} - \lambda_3 \frac{t_{13}}{(1-t_{13})} = 0$$

$$\frac{t_{12}}{(1-t_{12})} = \frac{l - f(v_2)}{f(v_1) - l}$$

$$\frac{t_{13}}{(1-t_{13})} = \frac{l - f(v_3)}{f(v_1) - l}$$

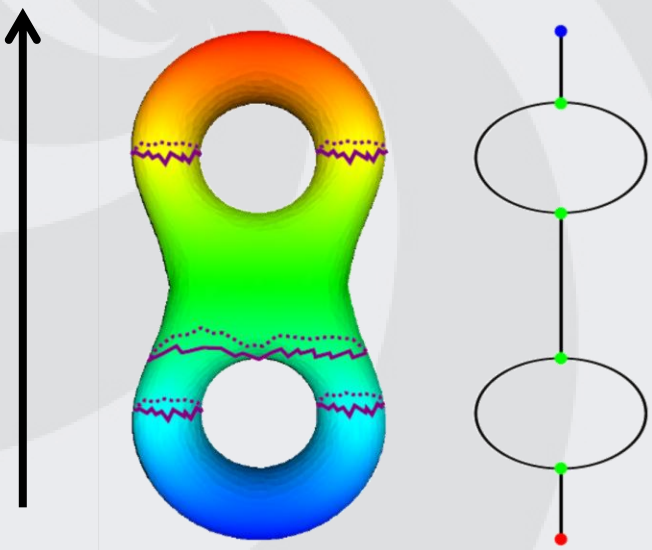
$$\lambda_1 - \lambda_2 \frac{l - f(v_2)}{f(v_1) - l} - \lambda_3 \frac{l - f(v_3)}{f(v_1) - l} = 0$$

$$\lambda_1(l - f(v_1)) + \lambda_2(l - f(v_2)) + \lambda_3(l - f(v_3)) = 0$$

but recalling that $\lambda_1 + \lambda_2 + \lambda_3 = 1$ it gives us:

$$\lambda_1 f(v_1) + \lambda_2 f(v_2) + \lambda_3 f(v_3) = l$$

Reeb graph on triangulated surfaces



On triangulated surfaces
the definition of Reeb graph
is even more robust:
when f^* is a Morse function
the definition of Reeb graph
goes as in the smooth case.

Reeb graph on triangulated surfaces

*When f^* is **not a Morse function**,*
*i.e. when there are **multiple saddles** for f^*,*
the definition of Reeb graph holds
but the graph might have
*nodes with **connectivity higher than 3**.*

Loop lemma on triangulated surfaces

Genus of triangulated surfaces

*For orientable, closed, triangulated, connected surfaces the **number of loops** in the Reeb graph corresponds to the **genus** of the surface*

(Cole-McLaughlin et al., 2003)

Reeb graph of X

$$\beta_1(R_{f^*}(X)) = g(X)$$

The actual Reeb graph depends on f *but the number of loops in the Reeb Graph does not vary*

Loop lemma on triangulated surfaces

Genus of triangulated surfaces

*For orientable, closed, triangulated, connected surfaces the **number of loops** in the Reeb graph corresponds to the **genus** of the surface*

(Cole-McLaughlin et al., 2003)

$$\chi(R_{f^*}(X)) = \beta_0(R_{f^*}(X)) - \beta_1(R_{f^*}(X))$$

$$\chi(R_{f^*}(X)) := V(X) - E(X)$$

$$V(R_{f^*}(X)) - E(R_{f^*}(X)) = \beta_0(R_{f^*}(X)) - \beta_1(R_{f^*}(X))$$

$$V(R_{f^*}(X)) - E(R_{f^*}(X)) = 1 - \beta_1(R_{f^*}(X))$$

$$\beta_1(R_{f^*}(X)) = 1 - V(R_{f^*}(X)) + E(R_{f^*}(X))$$

Loop lemma on triangulated surfaces

$$\beta_1(R_{f^*}(X)) = 1 - V(R_{f^*}(X)) + E(R_{f^*}(X))$$

n_1 = number of maxima and minima

n_{-i} = number saddles with index -i

$$V(R_{f^*}(X)) = n_1 + \sum_{i>0} n_{-i}$$

$$E(R_{f^*}(X)) = \frac{1}{2}(n_1 + \sum_{i>0}(2+i)\cdot n_{-i})$$

$$\beta_1(R_{f^*}(X)) = 1 - \frac{1}{2}(n_1 - \sum_{i>0} i \cdot n_{-i})$$

Loop lemma on triangulated surfaces

Theorem $\chi(X) = \sum_{v \in V(X)} i(v)$ [Critical Point Theorem - Banchoff, 1967]

Proof:

$$\chi(X) = \sum_{v \in V(X)} (1 - \frac{1}{2} t_c(v))$$

$$\chi(X) = V(x) - \sum_{v \in V(X)} \frac{1}{2} t_c(v)$$

$$\chi(X) = V(X) - \frac{1}{2} T(X)$$

$$\chi(X) := V(X) - E(X) + T(X)$$

$$3T(X) = 2E(X)$$

$$\chi(X) = V(X) - \frac{1}{2} T(X)$$

Theorem can be reformulated:

$$\chi(X) = n_1 - \sum_{i>0} i \cdot n_{-i}$$

Loop lemma on triangulated surfaces

$$\beta_1(R_{f^*}(X)) = 1 - \frac{1}{2}(n_1 - \sum_{i>0} i \cdot n_{-i})$$

$$\chi(X) = n_1 - \sum_{i>0} i \cdot n_{-i}$$

$$\beta_1(R_{f^*}(X)) = 1 - \frac{1}{2}\chi(X)$$

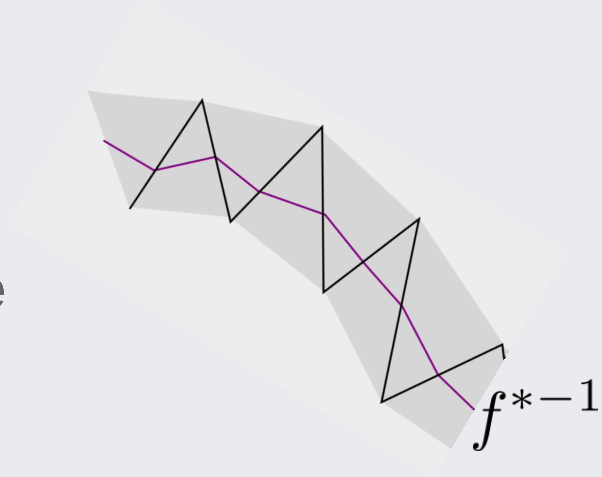
$$\chi(X) := 2 - 2g(X)$$

$$\beta_1(R_{f^*}(X)) = g(X)$$

An alternative approach?

Apropos computation…

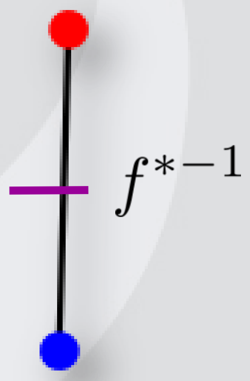
Computing the level lines of f^* is **expensive** as they **run on the surface** (i.e. **through** triangular faces)



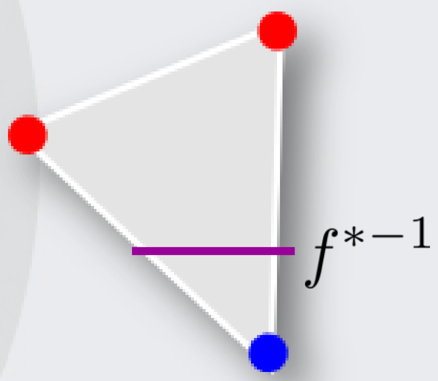
It would be nice to have a **computationally simpler** alternative.

But the alternative must be **theoretically sound**

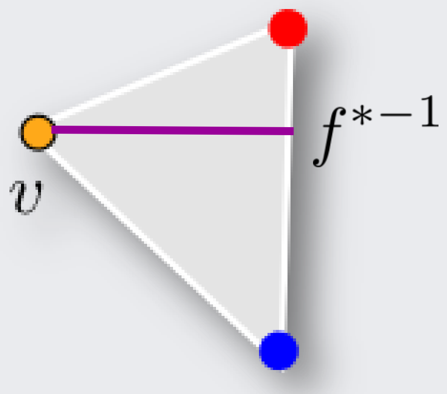
Cross simplexes



Cross edge



Cross face



Cross vertex

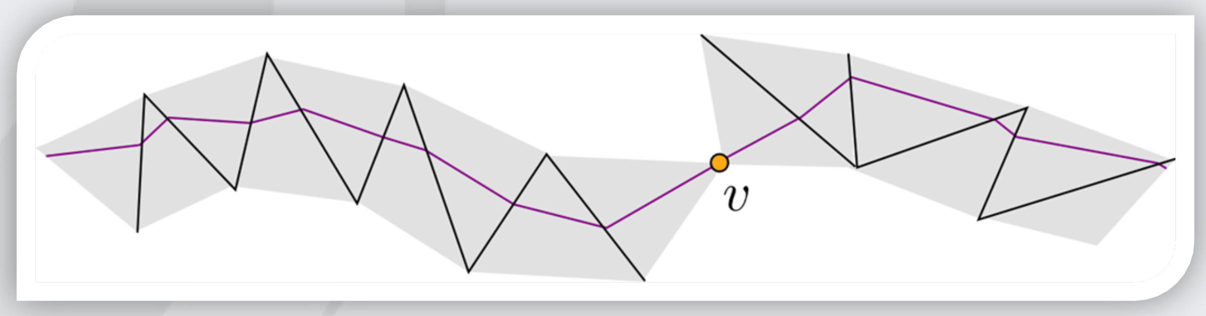
Level strip

*The **level strip** of a function f^* for α is the set of :*

-all the cross-triangles for α

-all the cross-edges for α

-the vertex v such that $f^*(v) = \alpha$



Homotopy between LS and f^{*-1}

LS is not a simplicial complex - some outer edges are missing

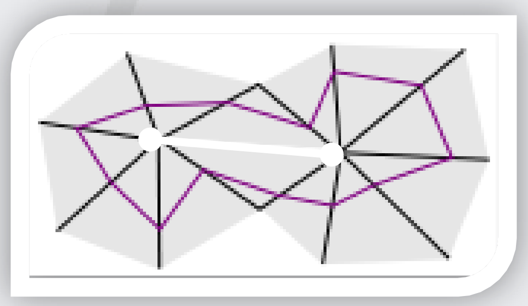
Homotopy between LS(α) and f^{*-1}

For each v $\in$V (X), the union of the interior of the simplices in the level strip LS(v) has the same homotopy type of the level set $f^{*-1}(f(v))$.

In addition, for any other value `in between'

(i.e. for which there is no v such that $\alpha = f(v)$),

the level strip LS(α) has the same homotopy type of $f^{*-1}(\alpha)$.



Deformation retraction

Given two topological spaces X, Y, such that Y $\subseteq$ X, a deformation retraction is a map :

$$h : \mathrm{X} \times [0,1] \to \mathrm{X}$$

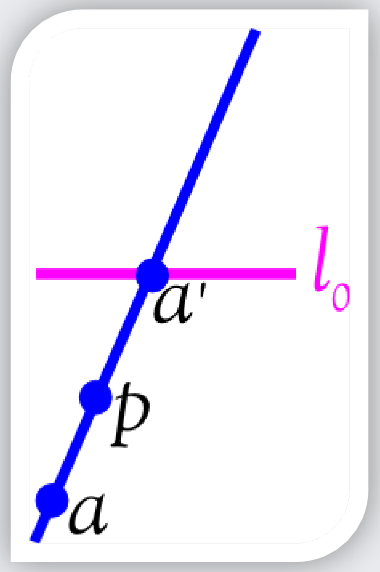
such that

- h is continuous
- $\forall x \in \mathrm{X},\ h(x,0) = x\ (i.e.\ h(.,0) = \mathrm{Id}_x)$
- $\forall x \in \mathrm{X},\ h(x,1) \in \mathrm{Y}$
- $\forall y \in \mathrm{Y},\ \forall t \in [0,1],\ h(y,t) = y$

Homotopy between LS and f^{*-1}

The interior of an edge face has the same homotopy type of the interior of the segment that crosses it

$$t \in [0,1]$$
$$p := a + t(a' - a)$$



Homotopy between LS and f^{*-1}

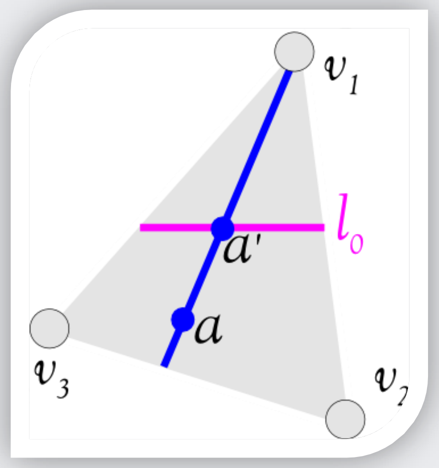
The interior of a triangular face has the same homotopy type of the interior of the segment that crosses it

Proof.

$$v_1 = (1,0,0)$$

$$a = (a_1, a_2, a_3)$$

$$a' = (\lambda_1, \lambda_2, \lambda_3)$$



$$\begin{vmatrix} 1 & 0 & 0 \\ a_1 & a_2 & a_3 \\ \lambda_1 & \lambda_2 & \lambda_3 \end{vmatrix} = 0$$

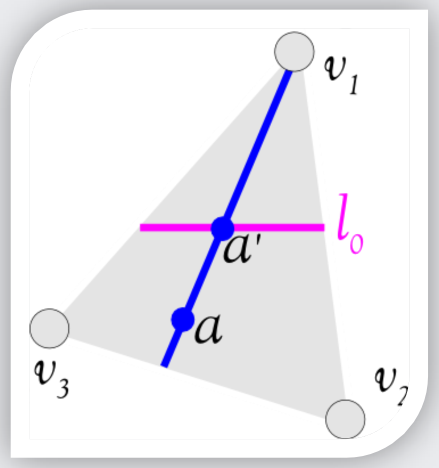
$$a_2\lambda_3 - a_3\lambda_2 = 0$$

$$a_2\lambda_3 = a_3\lambda_2$$

Homotopy between LS and f^{*-1}

The intersection between the radial line through a and the generic level line at l is:

$$\begin{cases} \lambda_1 f(v_1) + \lambda_2 f(v_2) + \lambda_3 f(v_3) = l \\ a_2\lambda_3 = a_3\lambda_2 \\ \lambda_1 + \lambda_2 + \lambda_3 = 1 \end{cases}$$



Whose solution is:

$$\lambda_1 = \frac{a_2 l + a_3 l - a_2 f(v_2) - a_3 f(v_3)}{a_2 f(v_1) - a_2 f(v_2) + a_3 f(v_1) - a_3 f(v_3)}$$

$$\lambda_2 = \frac{a_2(f(v_1) - l)}{a_2 f(v_1) - a_2 f(v_2) + a_3 f(v_1) - a_3 f(v_3)}$$

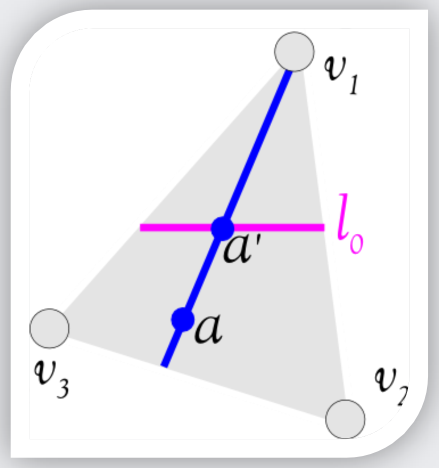
$$\lambda_3 = \frac{a_3(f(v_1) - l)}{a_2 f(v_1) - a_2 f(v_2) + a_3 f(v_1) - a_3 f(v_3)}$$

Homotopy between LS and f^{*-1}

The equation of f at a is:* $a_1 f(v_1) + a_2 f(v_2) + a_3 f(v_3) = l_a$

interior of a triangle

The map $h : \mathrm{X} \times [0,1] \to \mathrm{X}$
*projects each point **a** ∈ X to the intersection between the radial line through **a** and a level set at*
$l = l_a - t(l_a - l_0)$
where l_0 *is the value of the target level set.*



- clearly h is continuous
- $\forall a \in \mathrm{X},\ h(a,0) = a$ (that is the intersection between the radial line through a and the level set l_a)
- $\forall a \in \mathrm{X},\ h(a,1) \in \mathrm{Y}$ (where Y is the interior of the level set in X)
- $\forall a \in \mathrm{Y},\ \forall t \in [0,1],\ h(a,t) = a$

End proof

Contour strips

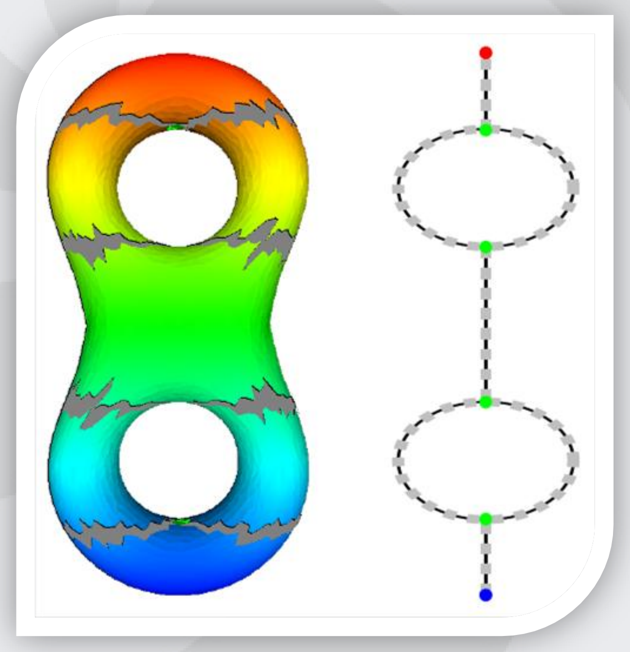
*The **contour strip of a vertex v** CS(v)*
*is the **connected component of LS(v)** that contains v.*

*If **v** is a **regular vertex** the CS(v) will contain*
*a **unique cycle***

*If **v** is either a **maximum or a minimum** the CS(v) will contain*
*the vertex **v itself***

*If **v** is a **saddle** the CS(v) will contain*
*a **number of cycles** equal to **m+1** (being m saddle multiplicity)*

Contour strip Reeb graph (CSRG)



*Each **point** of the **CSRG** corresponds to a contour strip:*

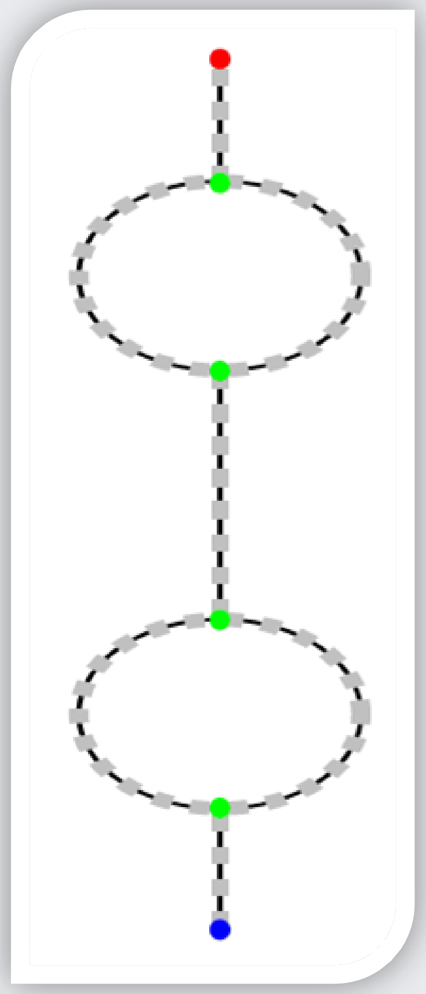
*- the **nodes** of the Reeb graph correspond to the **CS** of **critical vertices**.*

*- **all other points** in the graph correspond to the **CS** of **non-critical values***

*The **adjacency** of the **points of the Rg follows** the **adjacency** of the corresponding **contour strips***

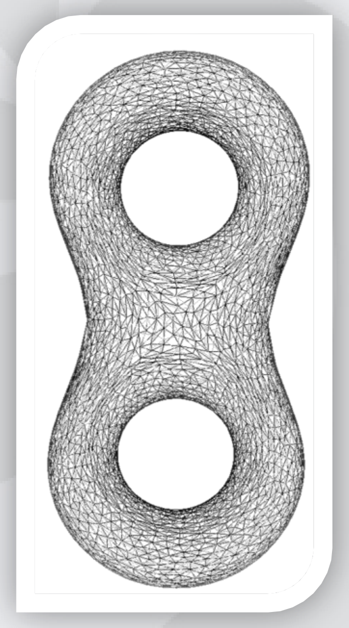
Contour strips adjacency

Given two vertices $\boldsymbol{v_1}$ *and* $\boldsymbol{v_2}$*, regular or saddle,*
such that $f(v_1) < f(v_2)$
their CS are ***adjacent***
if they ***share at least one cross face*** *and*
there is no other overlapping contour strip *for* $\boldsymbol{v_3}$
such that $f(v_1) < f(v_3) < f(v_2)$



If ***v*** *is a* ***minimum*** *or a* ***maximum***
the above definition applies
assuming that any other contour strip ***overlaps CS(v)***
if ***it overlaps St(v)***

A simpler approach?



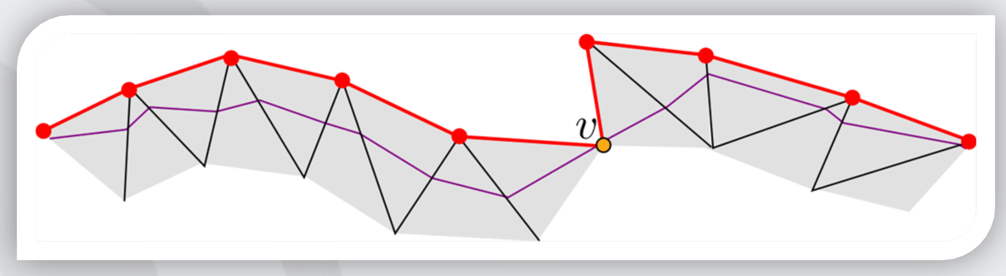
Level strips are an interesting approach because they do not require computing f^{*-1}.

But is there a simpler approximation of the level strip (V, E, T)? (…hence of the level line of f^{*}?)

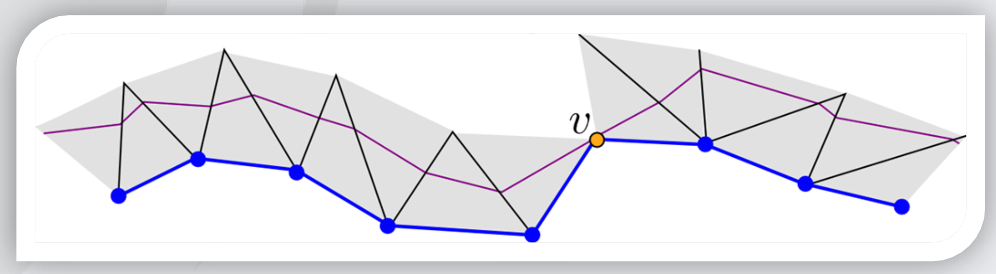
One approach that uses only mesh edges and vertices (V,E)?

Upper and lower level set

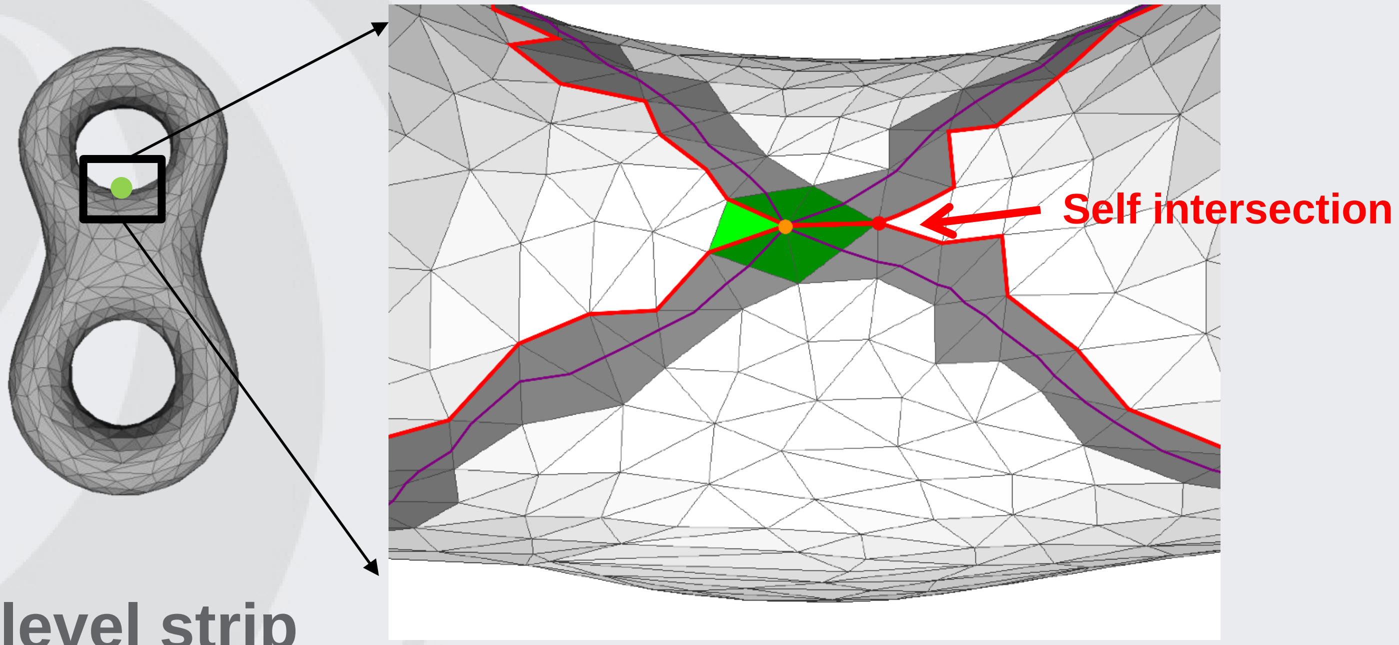
Upper Level Set



Lower Level Set



Upper level set

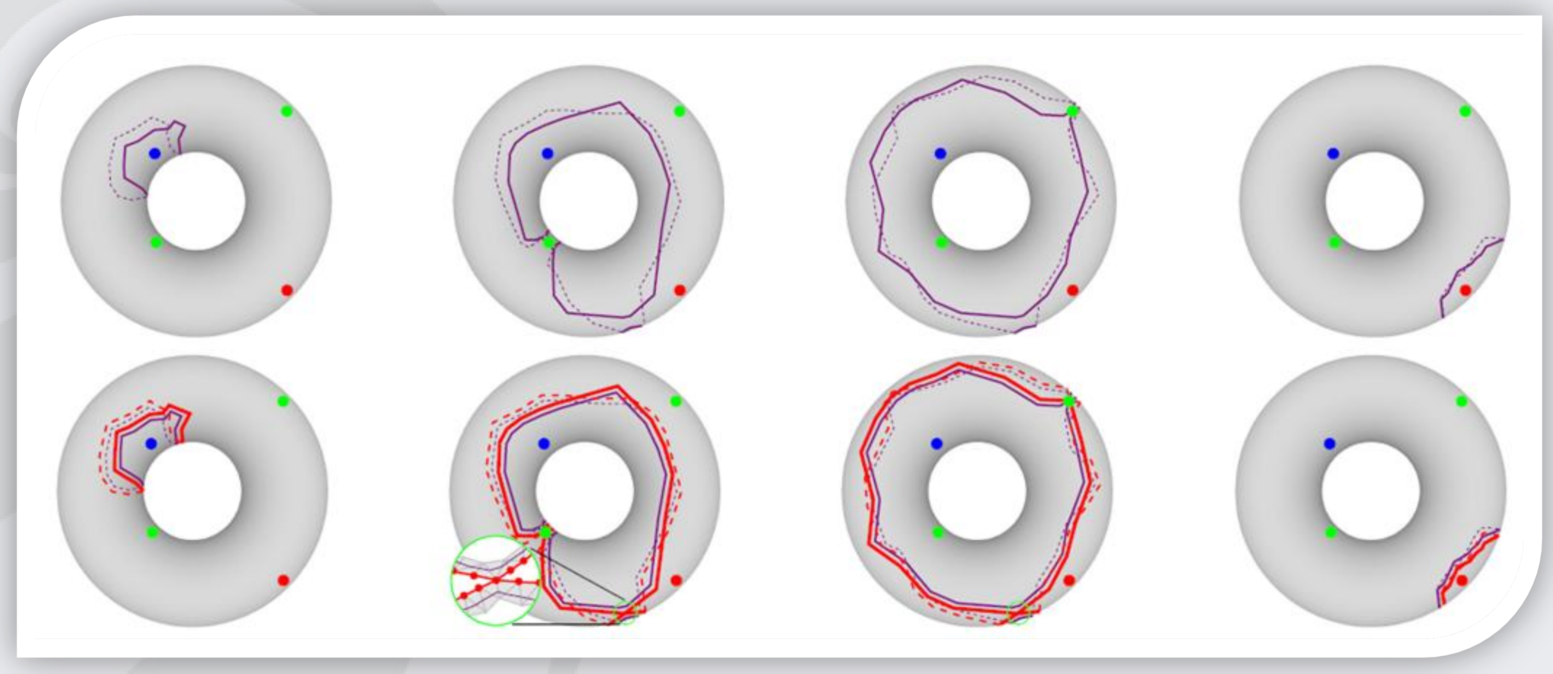


level strip

upper **level line**

level line of f^*

Counterexample

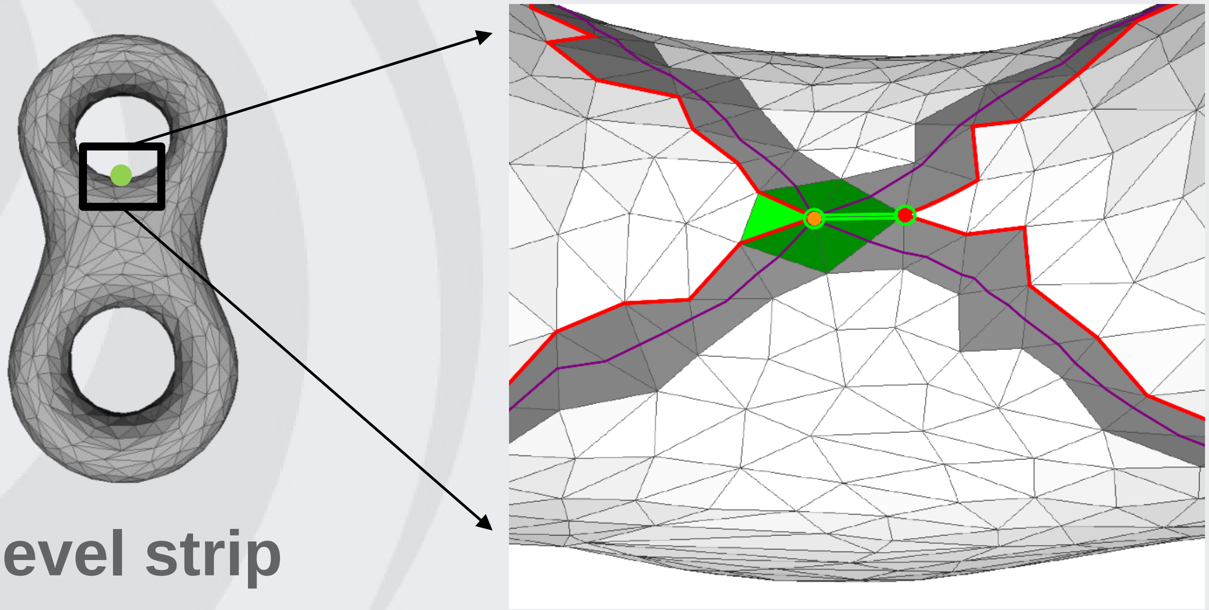


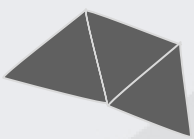
a b c d

In this case, the graph corresponding to the connected components of the ULS has **no loops** and **violates the Loop Lemma**

Solution: augmented *ULS*

Connected components** of the augmented Upper (lower) level set made up of edges and vertices with **possible multiple presence** are **topologically equivalent to the connected components of f^{*-1}





level strip

upper **level set**

level line of f^{*}

double presence

Multiplicity of an edge in augmented ULS

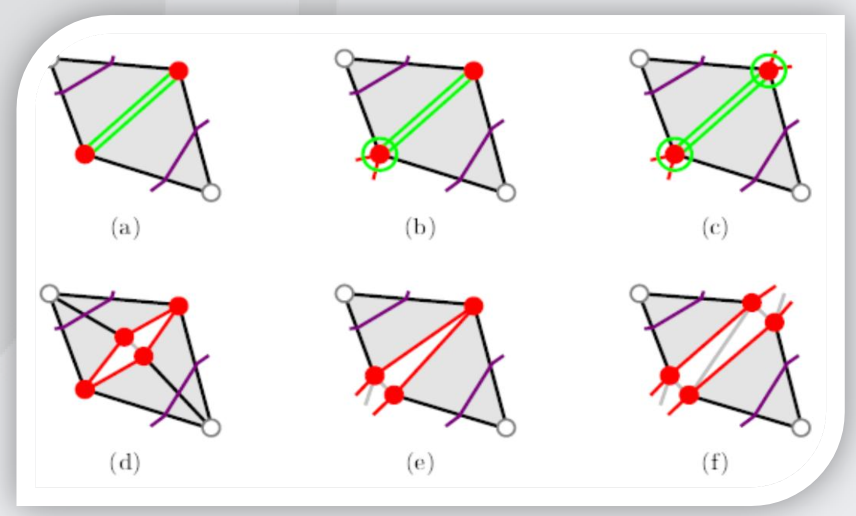
The **multiplicity** of an edge in the AULS(α)

is equal to **the number of faces** that are **adjacent to the edge** and **also belong to the level strip LS(α).**

An edge e$\in$ULS(α) has **multiplicity 2** in ULS(α) if:

$$\exists t_i, t_j \in LS(\alpha) | e \subset t_i, e \subset t_j$$

Any other edge e$\in$ULS(α) has **multiplicity 1**.



Multiplicity of a vertex in augmented ULS

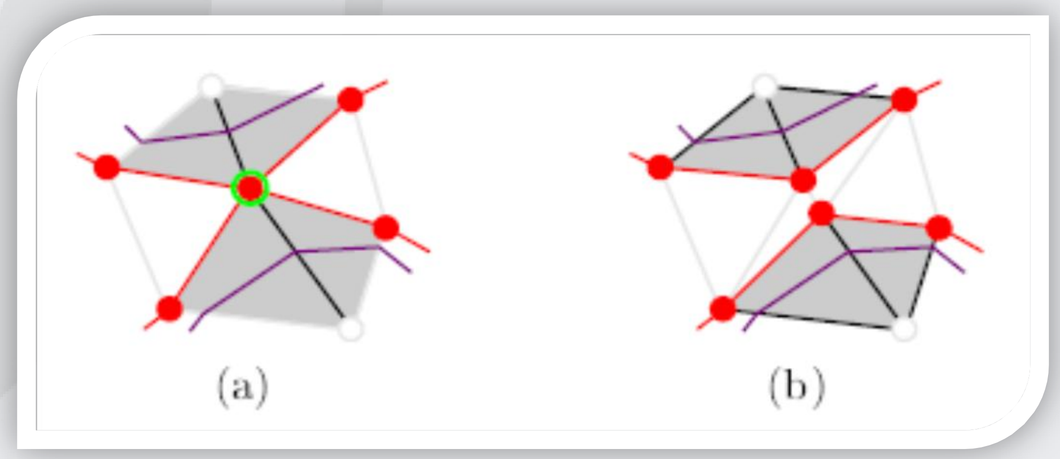
The multiplicity of a vertex v in the ULS(v) is equal to

1/2 the **sum of the multiplicity** of the **edges** in the augmented ULS

adjacent to the vertex

$$m(v) = \frac{1}{2} \sum_{e \in ULS(\alpha), v \prec e} m(e)$$

where m(e) is the multiplicity of the edge in ULS(α).



Is augmented ULS theoretically sound?

The **contour** γ(v) is the connected component of the **augmented** ULS(v)

Homotopy equivalence between a connected component of f^{*-1} and the contour γ(v)

A connected component of a level line f^{*-1} containing v

is homotopy equivalent to the corresponding

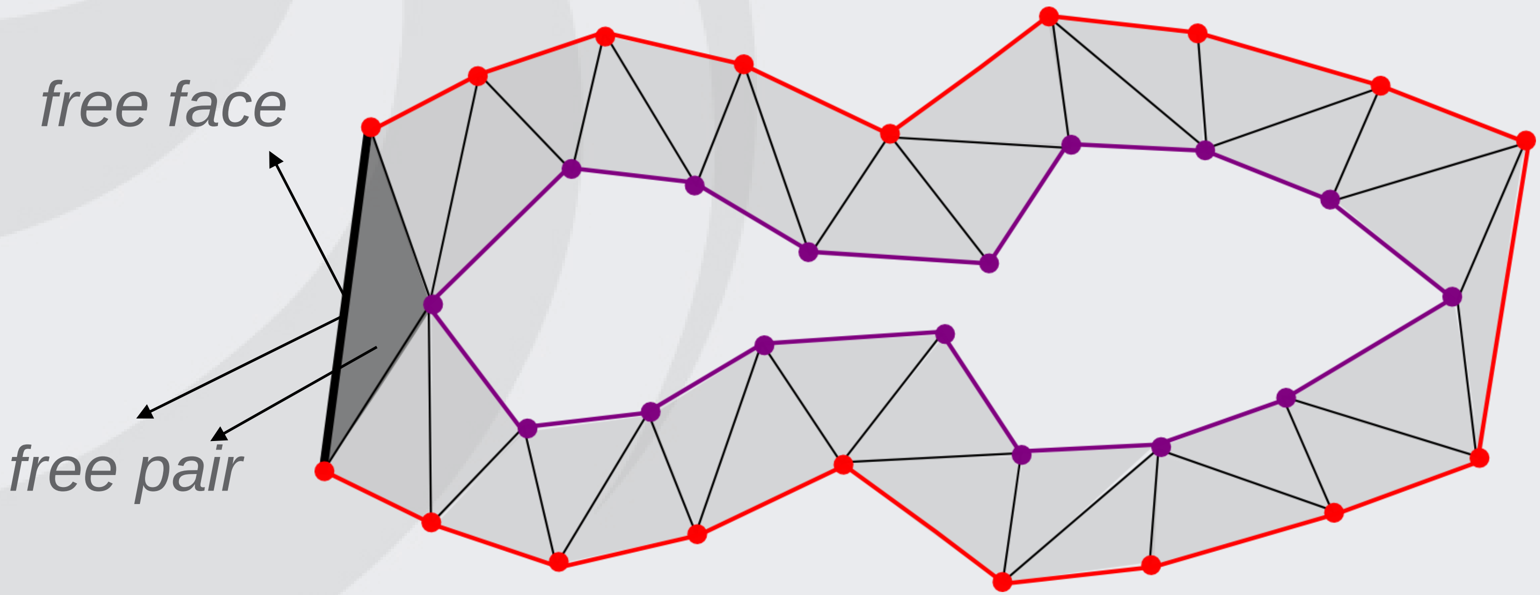
connected component of the augmented ULS (the contour γ(v))

if the latter contains at least one edge.

Free faces and free pairs

$\tau \in K$ is a **free face** iif there exists $\sigma \in K$ such that $\tau \subseteq \sigma$ and σ is unique, i.e. there is no other $\sigma' \in K$ such that $\tau \subseteq \sigma'$

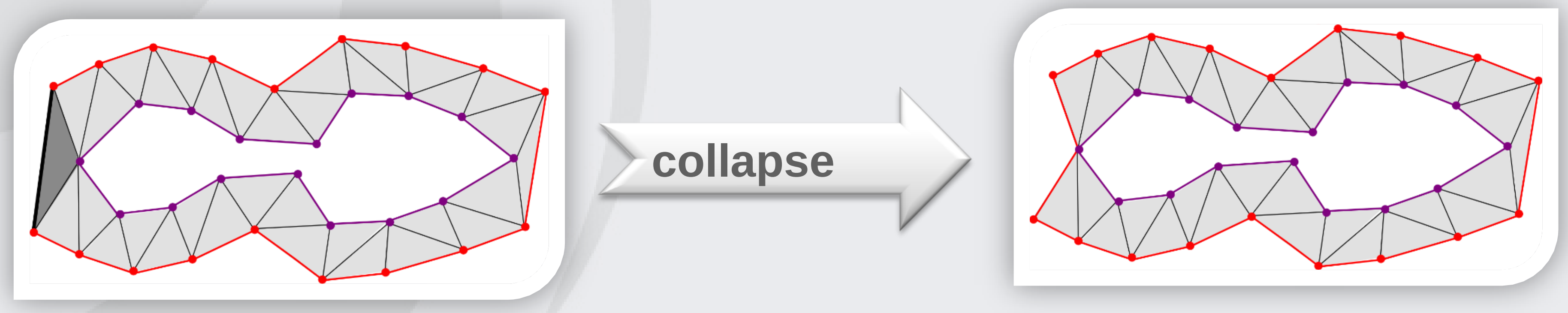
The pair (τ, σ) is a **free pair**.



Collapses

An **elementary collapse** of K is a subcomplex K' that can be obtained from K by removing one free pair.

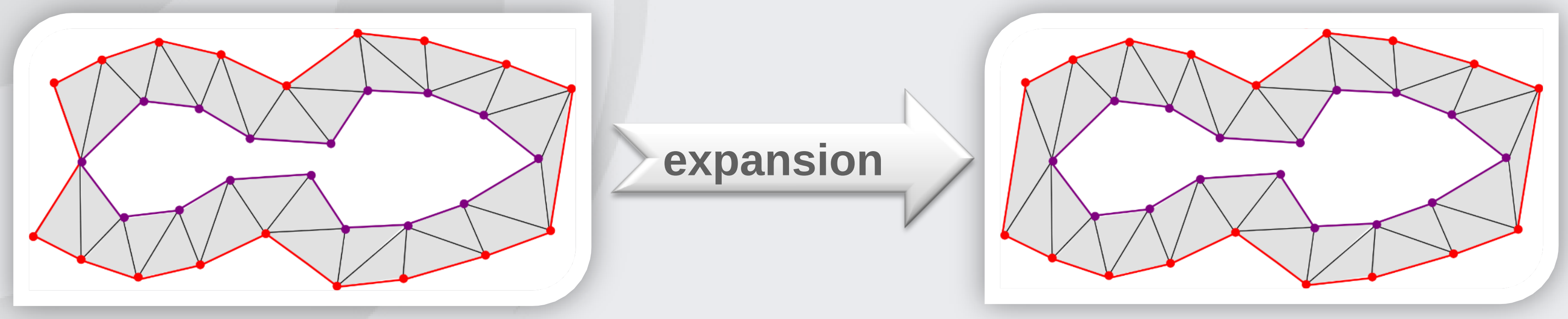
A **collapse** of K is a subcomplex K' that can be obtained from K via a sequence of elementary collapses.



Expansions

An **elementary expansion** of K is a subcomplex K' that can be reduced to K by removing one free pair.

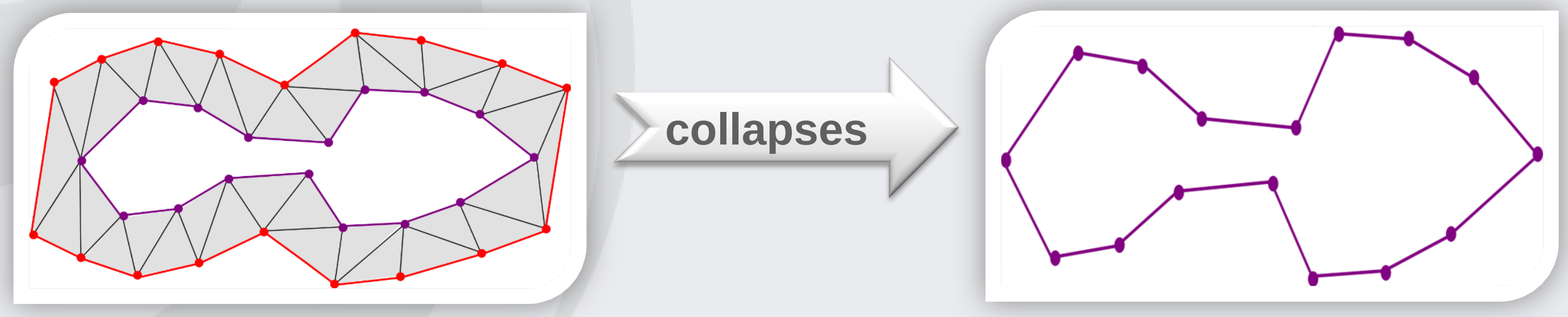
An **expansion** of K is a subcomplex K' that can be reduced to K via a sequence of elementary expansions.



Simple homotopy equivalence

Two simplicial complexes K and M are **simple-homotopy equivalent** if M can be obtained from K via a sequence of collapses and expansions.

If K and M are simple-homotopy equivalent they are also **homotopy equivalent**.

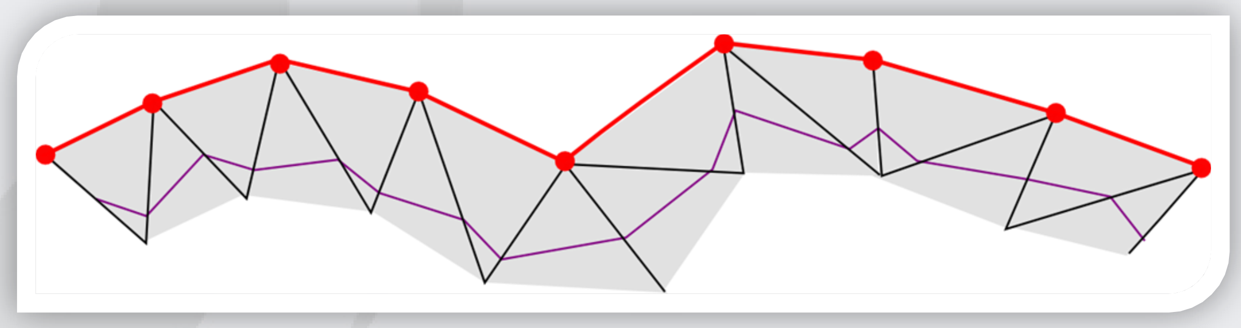


homotopy equivalent.

Homotopy equivalence: cc of f^{*-1} and γ(v)

Homotopy equivalence between a connected component of f^{*-1} and the contour γ(v)

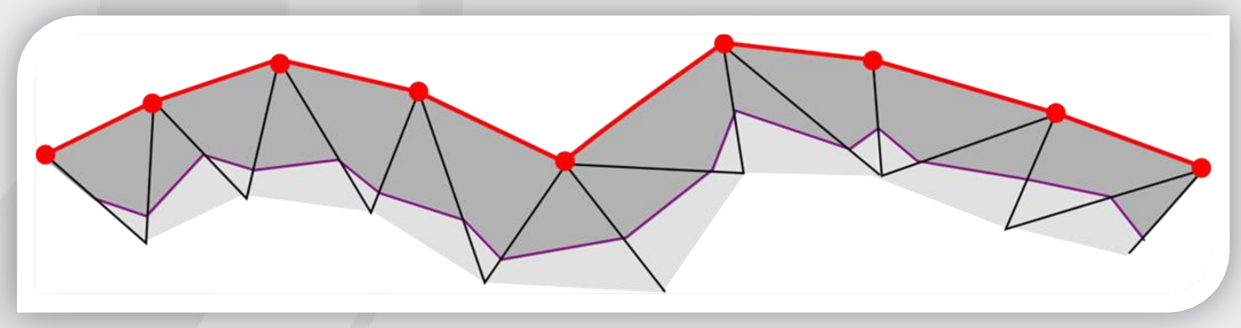
A connected component of a level line f^{*-1} containing v
is homotopy equivalent to the corresponding
connected component of the augmented ULS (the contour γ(v))
if the latter contains at least one edge.



Homotopy equivalence: cc of f^{*-1} and AULS

Homotopy equivalence between a connected component of f^{*-1} and the contour $\gamma(v)$

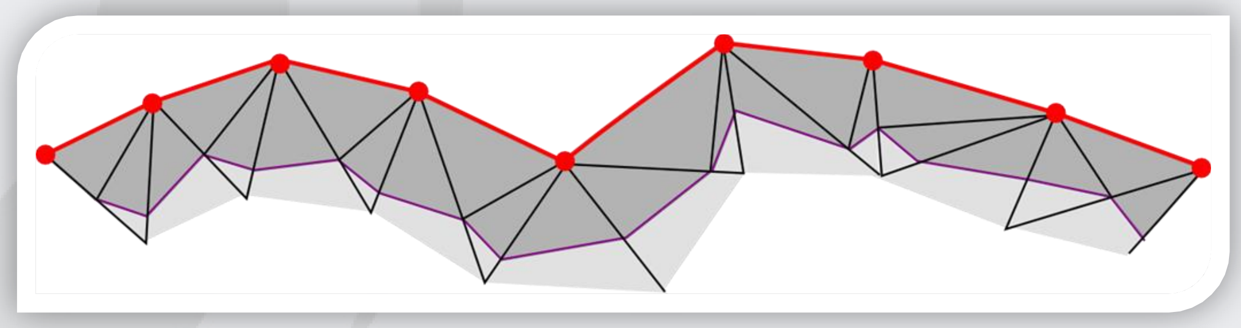
A connected component of a level line f^{*-1} containing v
is homotopy equivalent to the corresponding
connected component of the augmented ULS (the contour $\gamma(v)$)
if the latter contains at least one edge.



Homotopy equivalence: cc of f^{*-1} and AULS

<u>**Homotopy equivalence between a connected component of f^{*-1} and the contour**</u> γ(v)

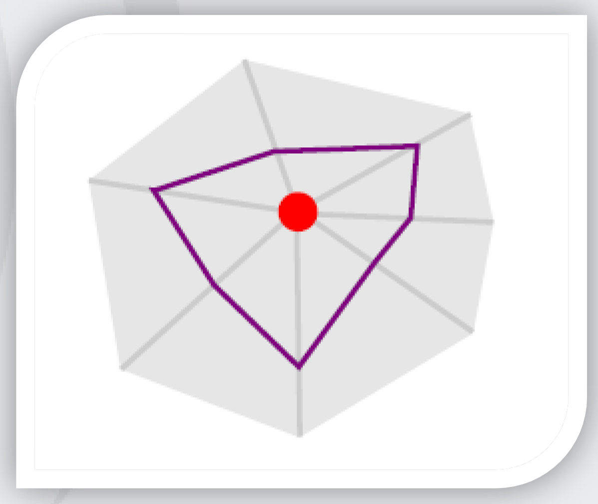
A connected component of a level line f^{*-1} containing v
is homotopy equivalent to the corresponding
connected component of the augmented ULS (the contour γ(v))
if the latter contains at least one edge.



Homotopy equivalence

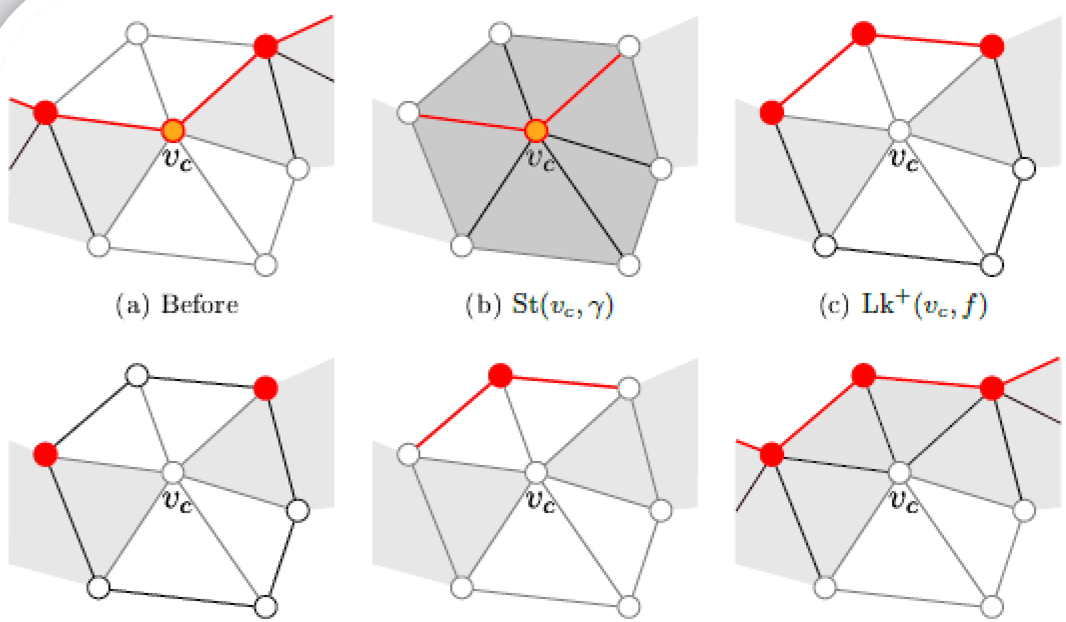
The contour $\gamma(v)$ must contain at least one edge,

otherwise the counterexample shown below becomes possible.



Contours' update

$$\gamma_{upd}(v) = \gamma(v) - \mathrm{St}(v, \gamma(v)) + (\mathrm{Lk}^+(v, f) - \mathrm{Lk}(v, \gamma(v)))$$



(a) Before (b) $\mathrm{St}(v_c, \gamma)$ (c) $\mathrm{Lk}^+(v_c, f)$

(d) $\mathrm{Lk}(v_c, \gamma)$ (e) $\mathrm{Lk}^+(v_c, f) - \mathrm{Lk}(v_c, \gamma)$ (f) After

Contours' adjacency

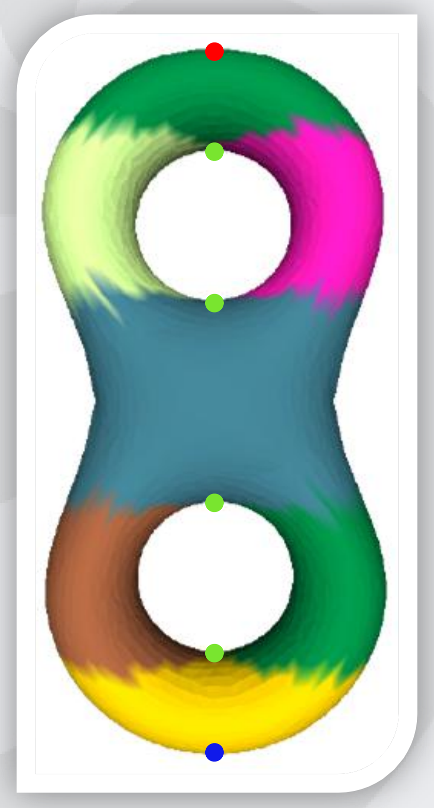
Two contours $\gamma(v_i)$, $\gamma(v_j)$ are adjacent if:

$$f(v_i) < f(v_j)$$

and there exists a contour γ such that:

$$\gamma \subseteq \gamma_{upd}(v_i) \ \ and \ \ \gamma \subseteq \gamma(v_j)$$

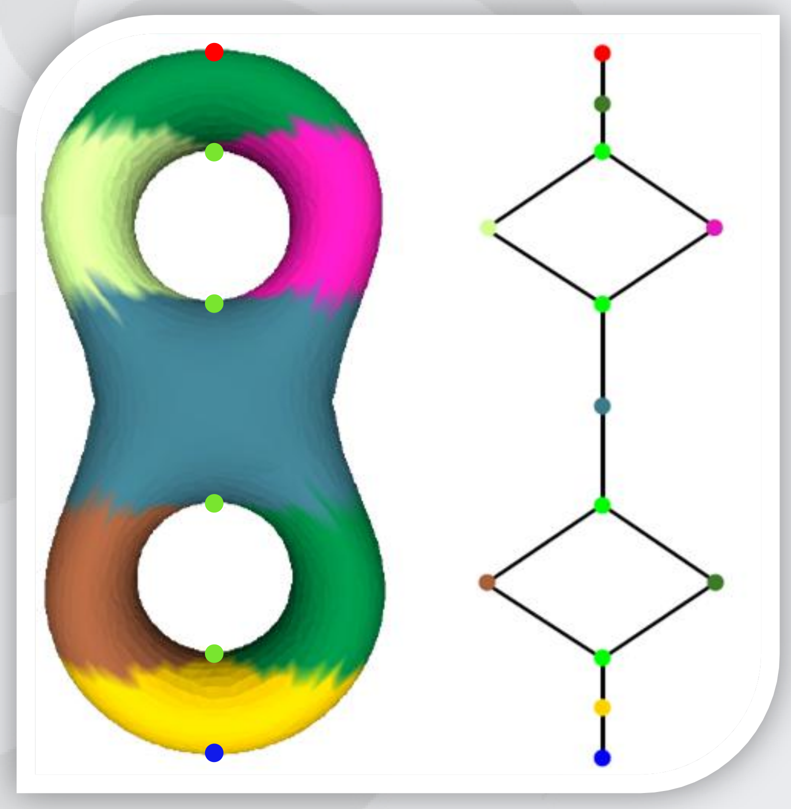
Segment



A **segment** is
the **portion of surface**
delimited by the contours
passing by the critical points

The simplified Reeb graph - SRG

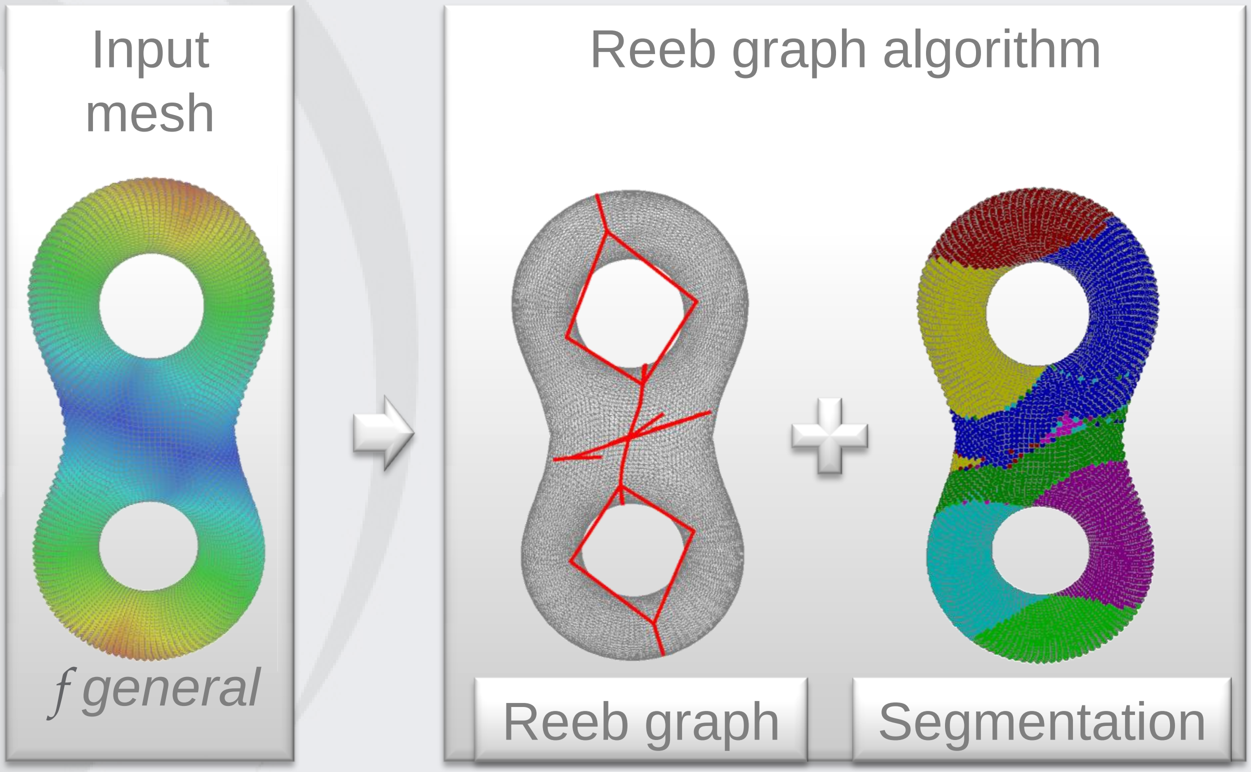
The simplified Reeb graph (SRG) is a graph that has:



- *one node for each* ***critical point***
- *one node for each segment.*

Each segment-node is adjacent to the nodes corresponding to the ***two critical points*** *whose* ***contours delimit the segment****.*

Algorithm stream



The basic idea of the algorithm

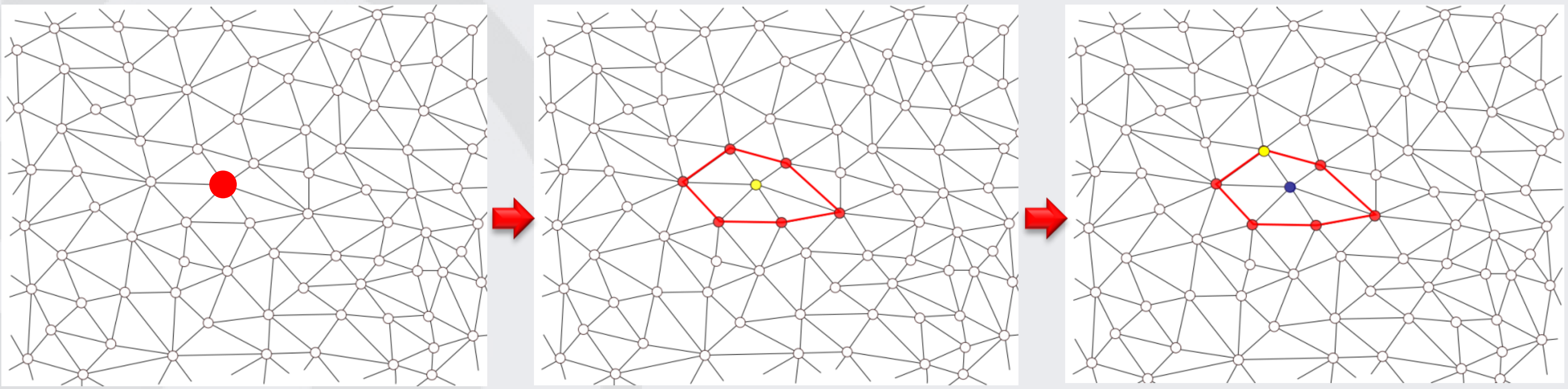
Contour = each **connected component** of an augmented **upper level line (AULL)**

Given a mesh and a general function f **:**

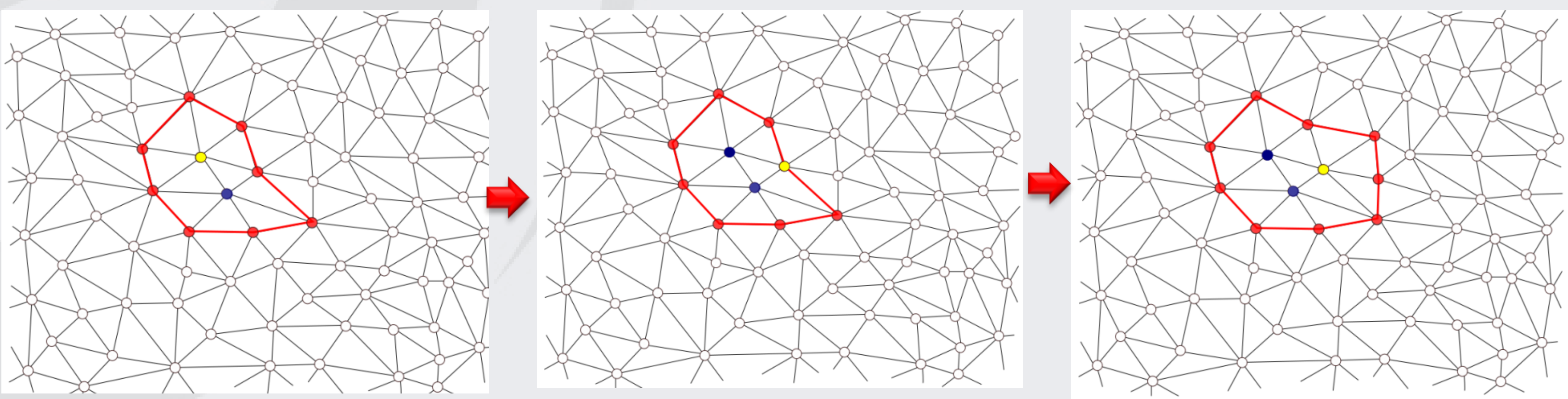
1. **initialize** a **contour** at each **local minima**;
2. **evolve** incrementally all the contours**;**
3. perform either **split or merge** operations at **critical points**;
4. **terminate** when all contours have reached a **local maximum**.

Contour evolution

Contour evolution starts at **local minima** and selects at **each step**



a **candidate vertex** has the lowest value of f in all contours

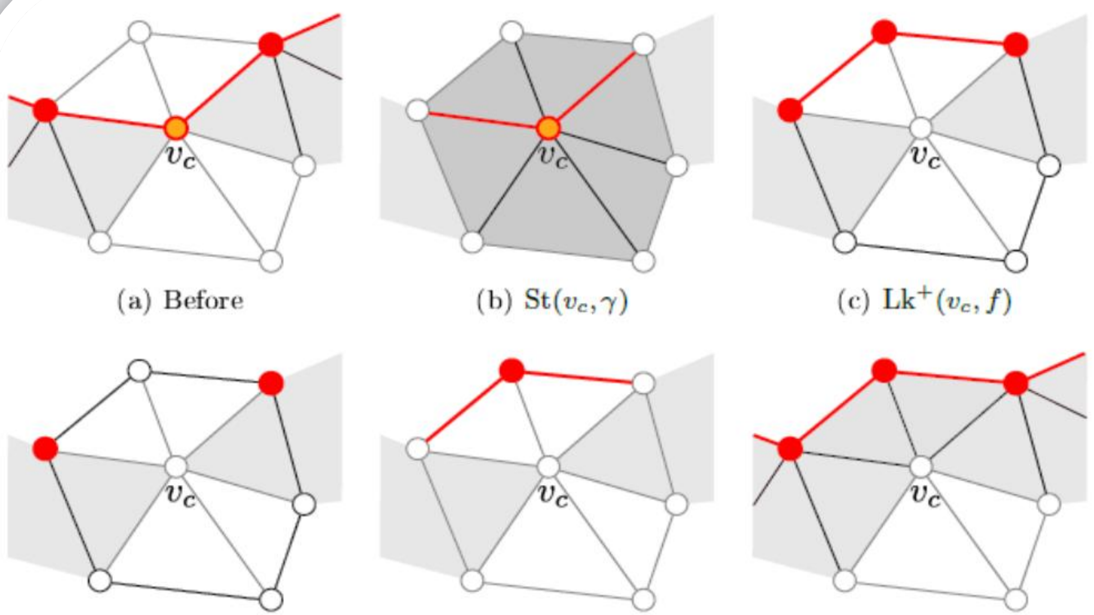


Initialize a contour at each local minima

$$
\begin{aligned}
&\Gamma \leftarrow \emptyset \\
&\Sigma \leftarrow \emptyset \\
&V_{minima} \leftarrow minima(T, f) \\
&\textbf{for each } v_{min} \in V_{minima} \\
&\quad \textbf{do} \begin{cases}
\textbf{comment: } \gamma \text{ is a } \textit{multiset} \text{ of vertices and edges} \\
\\
\sigma_{min} \leftarrow \text{new } \sigma(\{v_{min}\}) \\
\gamma \leftarrow \text{new } \gamma(\{v_{min}\}) \\
\sigma(\gamma) \leftarrow \text{new } \sigma(\emptyset) \\
\Gamma \leftarrow \Gamma \cup \{\gamma\} \\
\Sigma \leftarrow \Sigma \cup \{\sigma_{min}\} \cup \{\sigma(\gamma)\} \\
adjacency(\Sigma) \leftarrow adjacency(\Sigma) \cup \{(\sigma_{min}, \sigma(\gamma))\}
\end{cases}
\end{aligned}
$$

Contour evolution

$$
\begin{aligned}
&\textbf{procedure } \textsc{AdvanceContour}(v_c, \gamma)\\
&\quad \gamma \leftarrow \gamma - \mathrm{St}(v_c, \gamma)\\
&\quad \gamma \leftarrow \gamma \cup (\mathrm{Lk}^+(v_c, f) - \mathrm{Lk}(v_c, \gamma))
\end{aligned}
$$



(a) Before (b) $\mathrm{St}(v_c, \gamma)$ (c) $\mathrm{Lk}^+(v_c, f)$

(d) $\mathrm{Lk}(v_c, \gamma)$ (e) $\mathrm{Lk}^+(v_c, f) - \mathrm{Lk}(v_c, \gamma)$ (f) After

Evolve incrementally all the contours

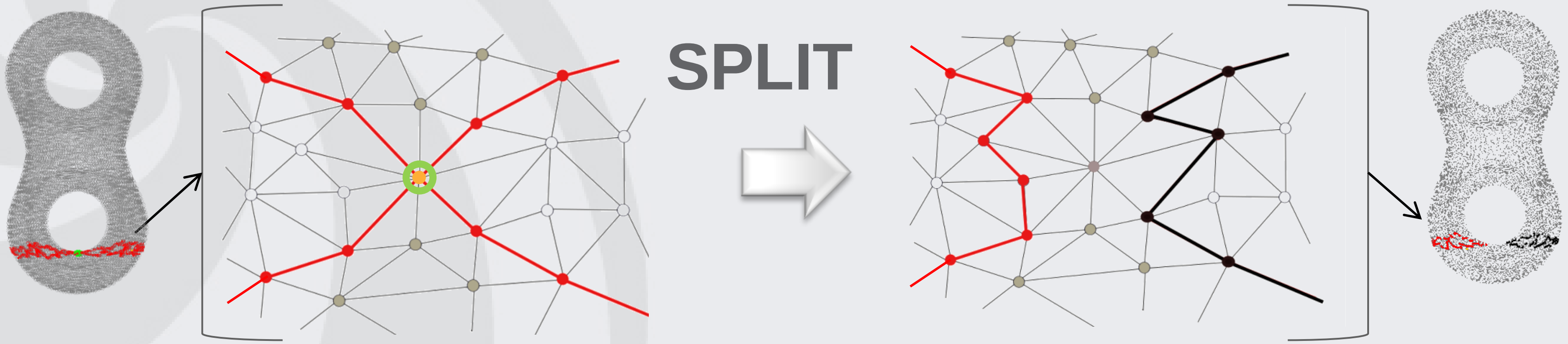
$$
\textbf{while}\ (\Gamma \neq \emptyset)
$$

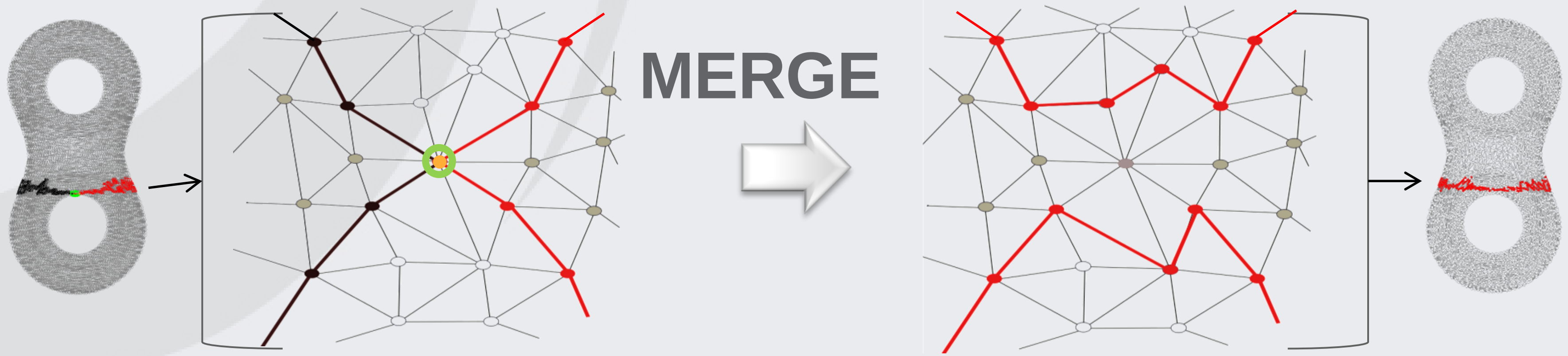
$$
\textbf{do}\ \begin{cases}
v_c \leftarrow \arg\min_{v \in \Gamma} f(v) \\
\Gamma_c \leftarrow \{\gamma \in \Gamma \mid v_c \in \gamma\} \\
\textbf{if}\ (\textsc{isMultiple}(v_c, \Gamma_c)) \\
\quad \textbf{then}\ \textsc{mergeSplit}(v_c, \Gamma_c) \\
\quad \textbf{else}\ \begin{cases}
\textsc{advanceContour}(v_c, \Gamma_c(1)) \\
\textbf{if}\ \ \textbf{not}\ \ (isEmpty(\gamma)) \\
\quad \textbf{then}\ \sigma(\gamma) \leftarrow \sigma(\gamma) \cup \{v_c\} \\
\quad \textbf{else}\ \begin{cases}
\Gamma \leftarrow \Gamma - \{\gamma\} \\
\sigma_{max} \leftarrow \text{new}\ \ \sigma(\{v_c\}) \\
\Sigma \leftarrow \Sigma \cup \{\sigma_{max}\} \\
adjacency(\Sigma) \leftarrow adjacency(\Sigma) \cup \{(\sigma(\gamma), \sigma_{max})\}
\end{cases}
\end{cases}
\end{cases}
$$

$$
\textbf{output}\ \ (\Sigma)
$$

Contour events at critical points

Split or **merge** operations are performed each time a critical point is met:

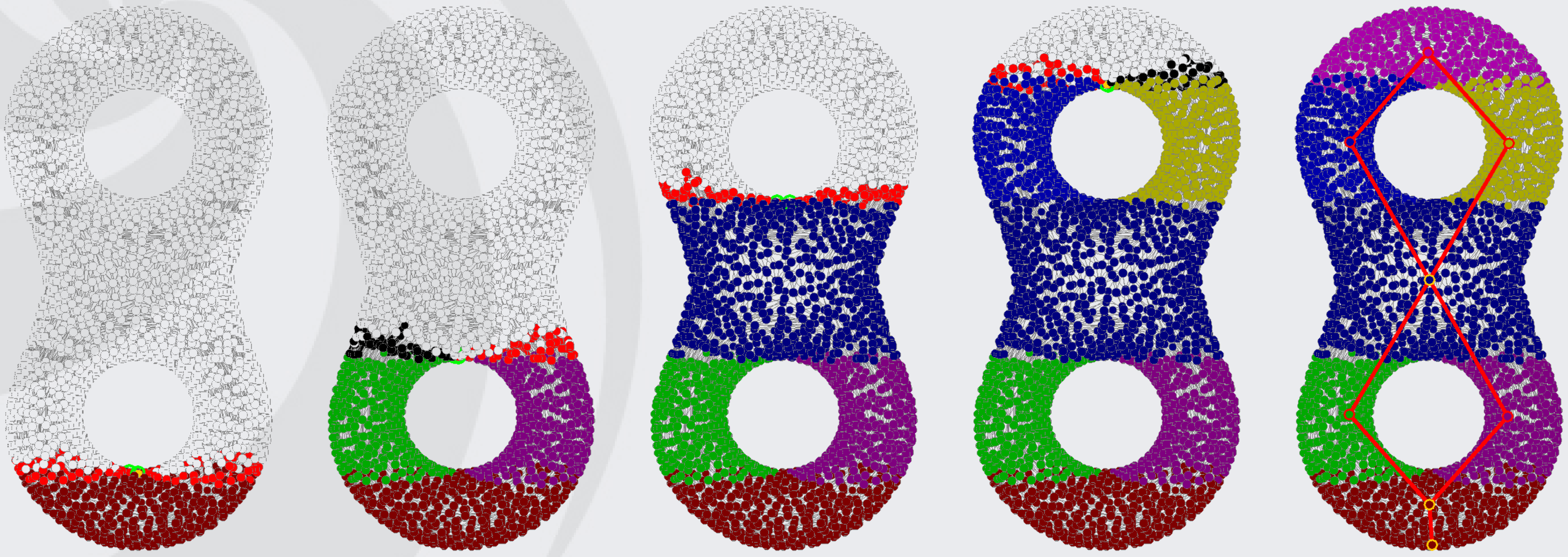




MERGESPLIT procedure

$$\gamma_{tmp} \leftarrow \text{new } \gamma(\emptyset)$$
$$\sigma_{saddle} \leftarrow \text{new } \sigma(\{v_c\})$$
$$\Sigma \leftarrow \Sigma \cup \sigma_{saddle}$$
$$\textbf{for each } \gamma_c \in \Gamma_c$$
$$\quad \textbf{do } \begin{cases} \gamma_{tmp} \leftarrow \gamma_{tmp} \cup \gamma_c \\ adjacency(\Sigma) \leftarrow adjacency(\Sigma) \cup \{(\sigma(\gamma_c), \sigma_{saddle})\} \end{cases}$$
$$\textsc{advanceContour}(v_c, \gamma_{tmp})$$
$$\Gamma_n \leftarrow \textsc{connectedComponents}(\gamma_{tmp})$$
$$\textbf{for each } \gamma_n \in \Gamma_n$$
$$\quad \textbf{do } \begin{cases} \Gamma \leftarrow \Gamma \cup \{\gamma_n\} \\ \sigma(\gamma_n) \leftarrow \text{new } \sigma(\{v \in \gamma_n\}) \\ \Sigma \leftarrow \Sigma \cup \sigma(\gamma_n) \\ adjacency(\Sigma) \leftarrow adjacency(\Sigma) \cup \{(\sigma_{saddle}, \sigma(\gamma_n))\} \end{cases}$$

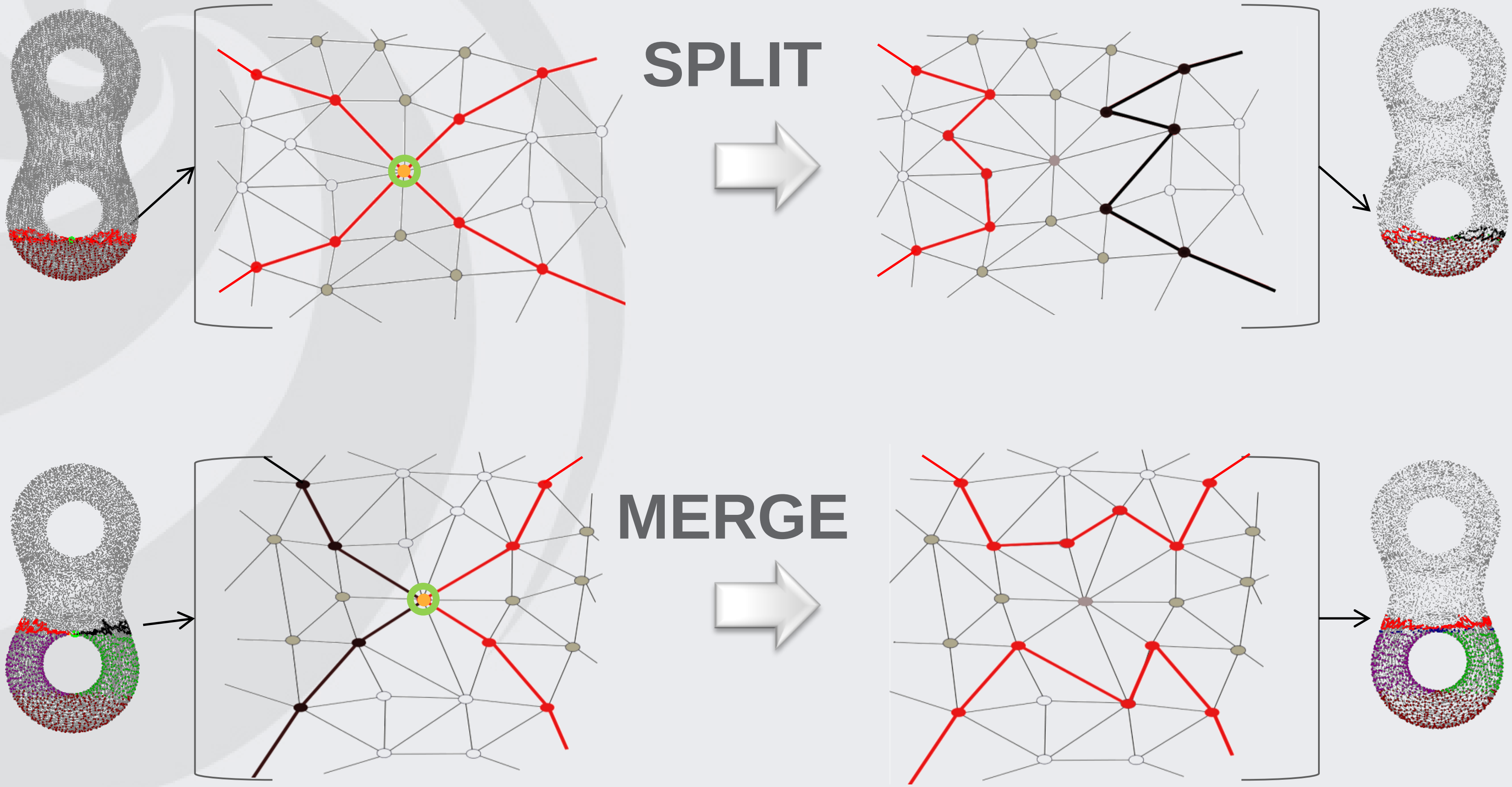
Segmentation



Incremental segmentation

When a split or merge event occurs

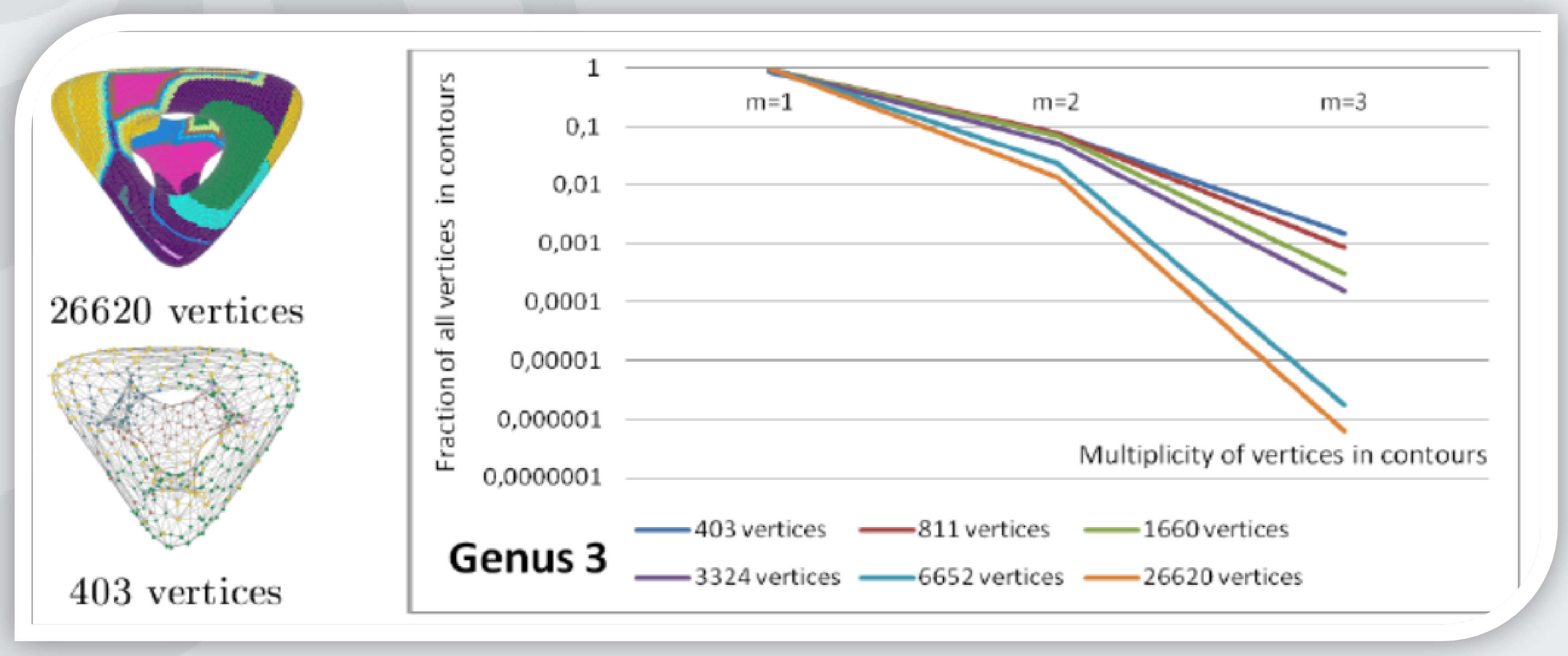
parent segments are **closed** an **new ones** are **created**



Multiple presence and mesh coarseness

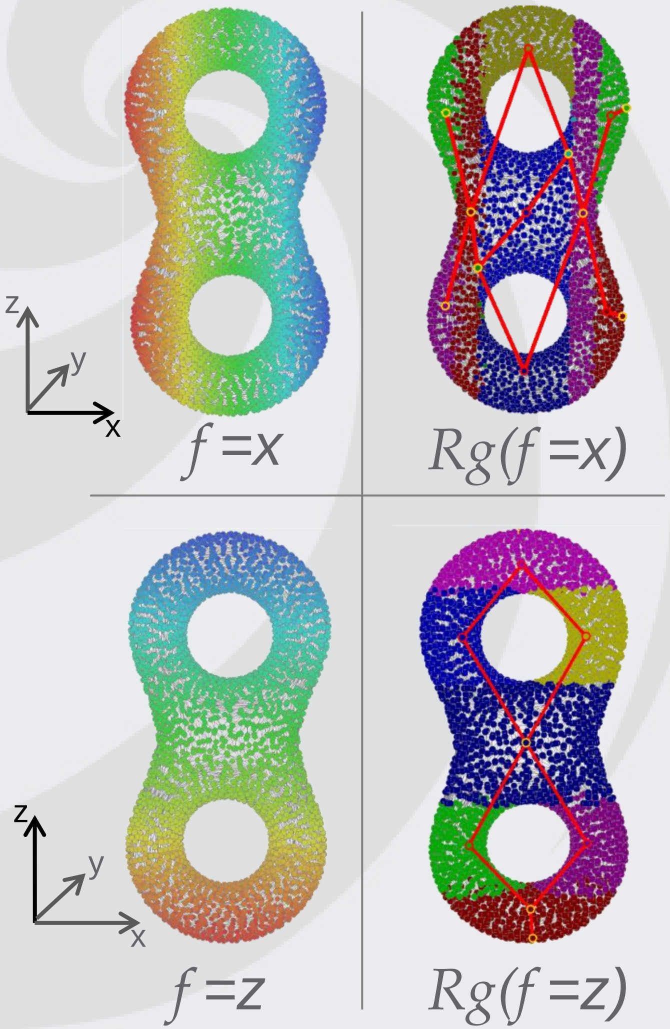
Multiple presence **arise** and **growth** with the **decrease** of mesh **sampling**:

a smaller number of vertices causes an increase of multiplicity of edges and vertices

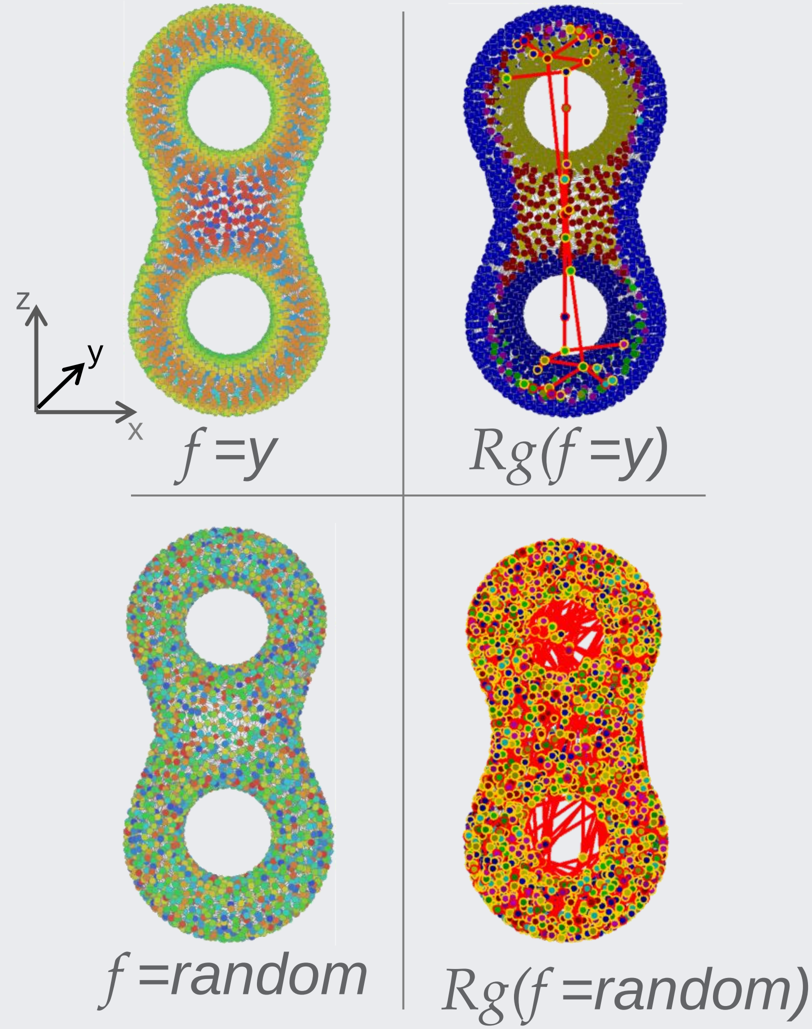


Some *f* functions and their Reeb graphs

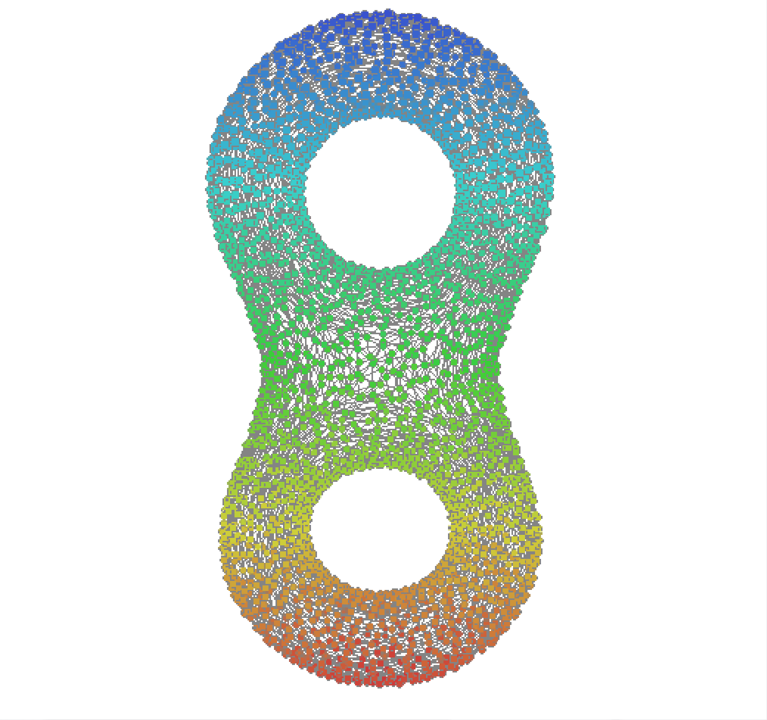
Reeb graph and segmentation

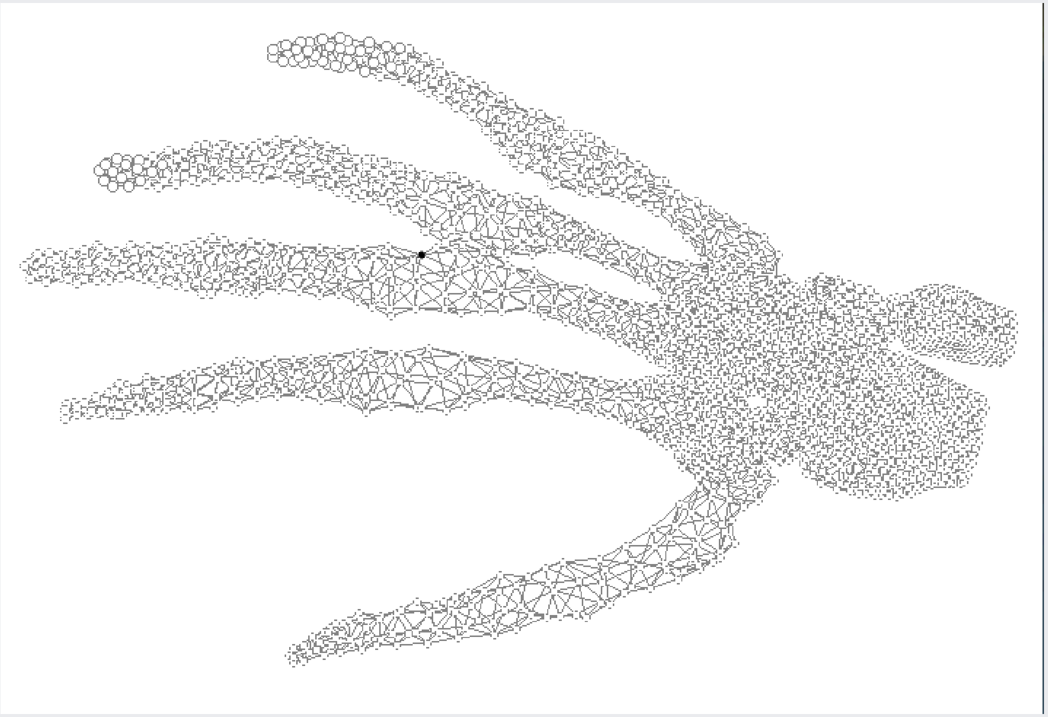


Reeb graph and segmentation



Algorithm in action





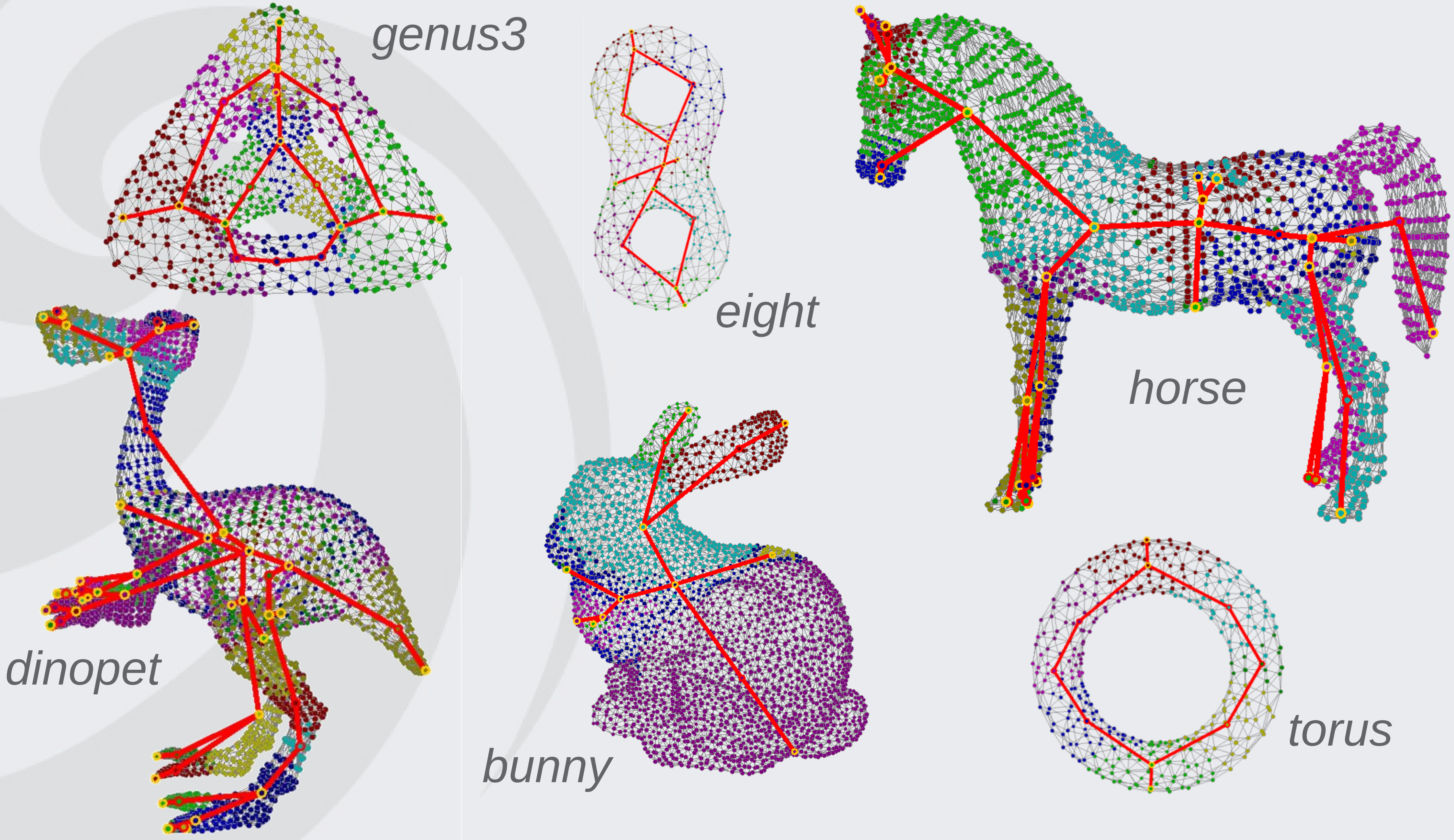
Experimental evidences

Mesh	Genus	Vertices	f
bunny	0	3052	geodesic distance from closest feature point
torus	1	359	
eight	2	190	
		319	
		382	
		766	
		3070	
		12286	
genus3	3	412	
		782	
		828	
		1660	
		3324	
		6652	
		26620	
hand-genus5	5	4037	
hand-genus8	8	3639	
heptoroid	22	10851	

Mesh	Genus	Vertices	f
eight - f=x	2	3070	height
eight - f=y			
eight - f=z			

Mesh	Genus	Vertices	f
bunny	0	3052	random
torus	1	359	
eight	2	190	
		190	
		382	
		382	
		766	
		3070	
genus 3	3	782	
		828	
hand-genus5	5	4037	
hand-genus8	8	3639	
heptoroid	22	10851	

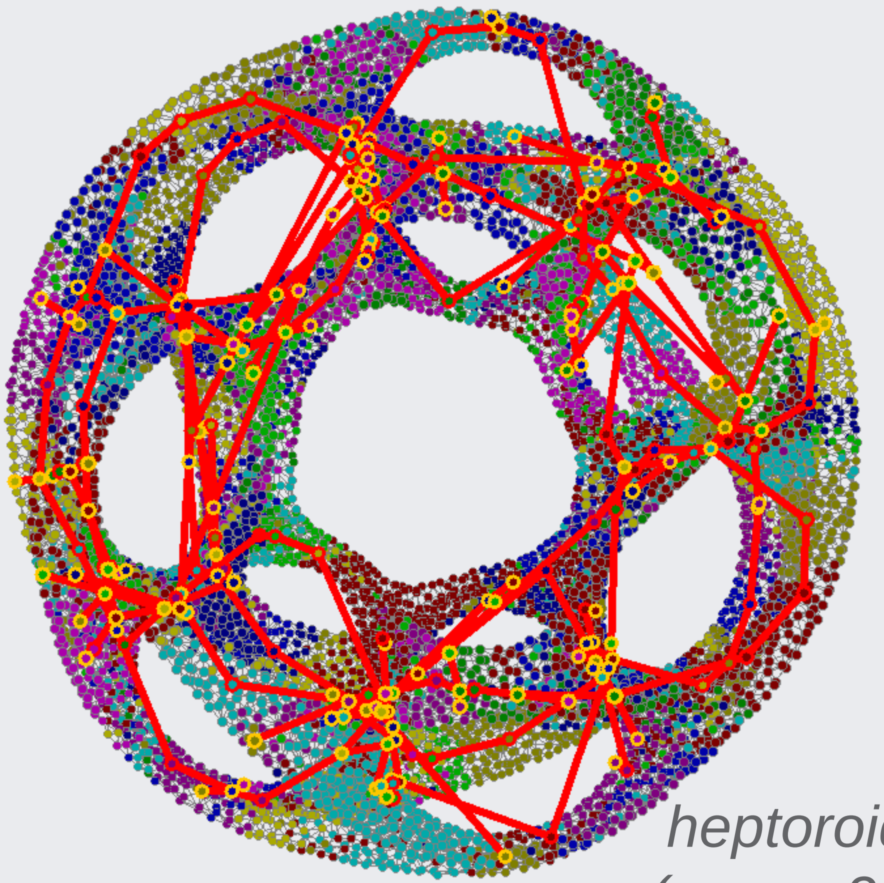
Experiment gallery



Validation test: *the number of loops in the Reeb graph corresponds to the genus of the manifold and the Rg respects the adjacencies*

Experiment gallery

hand
(genus5)



heptoroid
(genus 22)

Validation test: *the number of loops in the Reeb graph corresponds to the genus of the manifold and the Rg respects the adjacencies*

Publications

Computing the Reeb Graph for Triangle Meshes with Active Contours
L. Brandolini and M. Piastra, ICPRAM 2012 - Vilamoura, Algarve, Portugal, 6-8 February, 2012 – **Best Student Paper Award**

Simplified Reeb Graph as Effective Shape Descriptor for the Striatum
A. Pepe, L. Brandolini, M. Piastra, J. Koikkalainen, H. Juha, J. Tohka – to appear MICCAI 2012 MeshMed workshop, 1 October 2012, Nice, France

Simplified Reeb Graph for triangle meshes
L. Brandolini, M. Piastra – article submitted to *Computers & Graphics (Special Issue on Executable Papers for 3D Object Retrieval)*

The end

Thank you!

Laura Brandolini
laura.brandolini@unipv.it