Sub-Topics

- Compute bounding box
- Compute Euler Characteristic
- Estimate surface curvature
- Line description for conveying surface shape
- Extract skeletal representation of shapes
- Morse function and surface topology--Reeb graph
- Scalar field topology--Morse-Smale complex

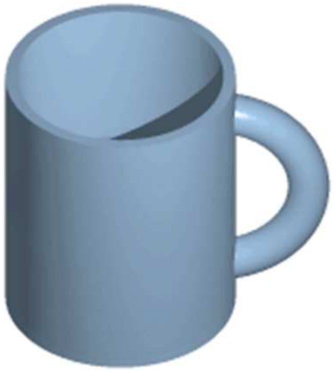
Surface Topology – Reeb Graph

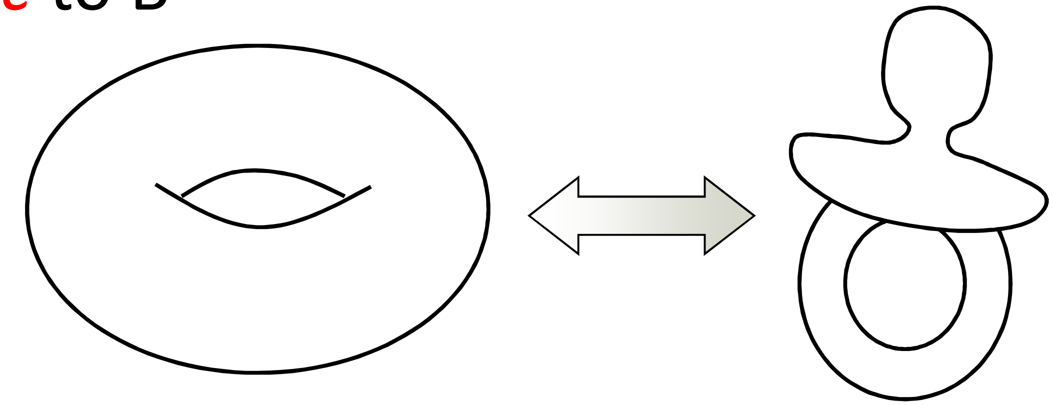
What is Topology?

- Topology studies the connectedness of spaces
- For us: how shapes/surfaces are connected

What is Topology

- The study of property of a shape that does not change under *deformation*
  - Rules of deformation
    - 1-1 and onto
    - Bicontinuous (continuous both ways)
    - Cannot tear, join, poke or seal holes
  - A is *homeomorphic* to B



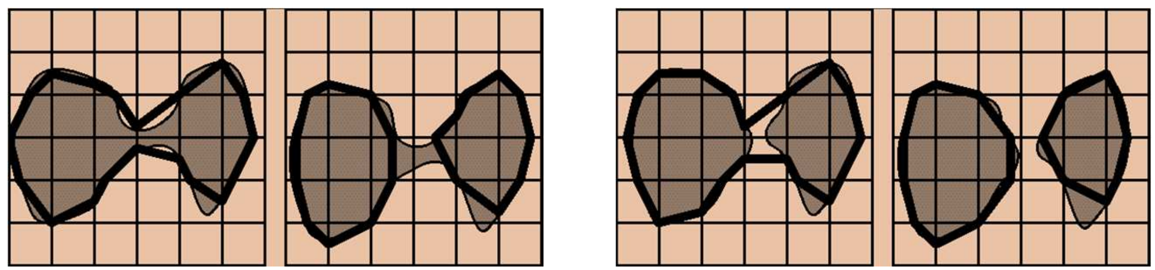


Why is Topology Important?

- What is the **boundary** of an object?
- Are there **holes** in the object?
- Is the object **hollow**?
- If the object is transformed in some way, are the changes **continuous** or abrupt?
- Is the object **bounded**, or does it extend infinitely far?

Why is Topology Important?

- Inherent and basic properties of a shape
- We want to accurately represent and preserve these properties in different applications
  - Surface reconstruction
  - Morphing
  - Texturing
  - Simplification
  - Compression



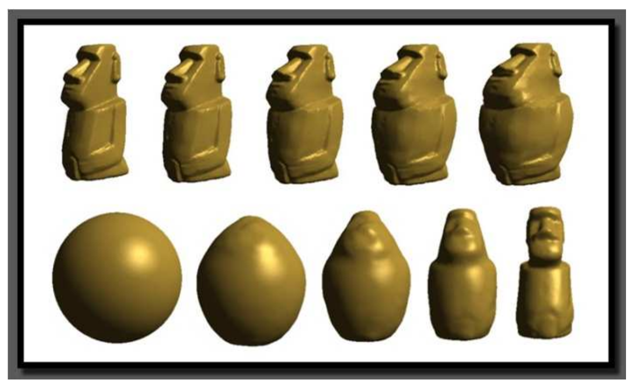
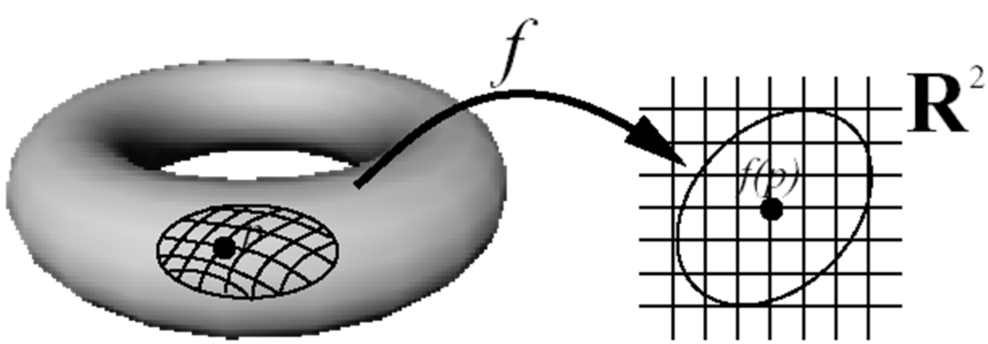
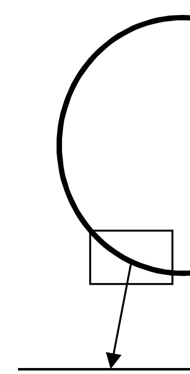


Image source:
http://www.utdallas.edu/~xxg061000/physicsmorphing.htm

Topology-Basic Concepts

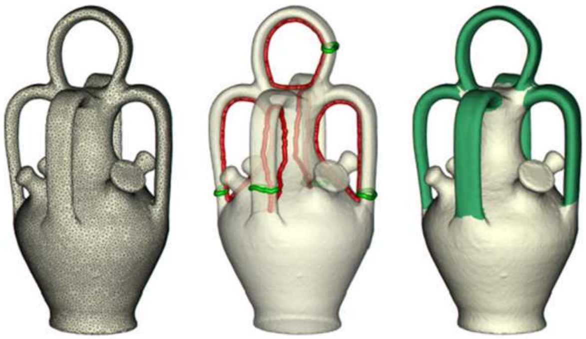
- **n-manifold**
  - Set of points $M \subset R^m$
  - Each point has a neighborhood homeomorphic to an open set of R^n
  - An **n-manifold** is a topological space that *"locally looks like"* the Euclidian space R^n





Topology-Basic Concepts

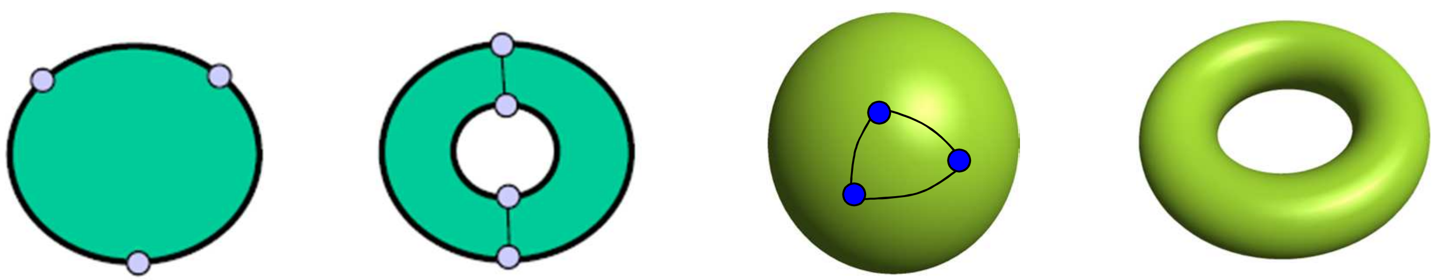
- Holes/genus
  - Genus of a surface is the maximal number of ***nonintersecting simple closed curves*** that can be cut on the surface without disconnecting it.



- Boundaries

Topology-Basic Concepts

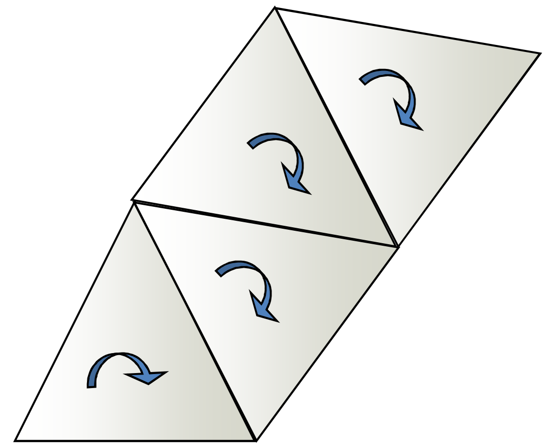
- Euler's characteristic function χ
  - $\chi(M) = V - E + F$
    - V = #vertices, E = #edges, F = #faces
  - $\chi(M)$ is independent of the polygonization
  - Specifically, $\chi(M) = 2c - 2g - h$
    - What are c, g, and h?



$\chi = 1$ $\chi = 0$ $\chi = 2$ $\chi = 0$

Topology-Basic Concepts

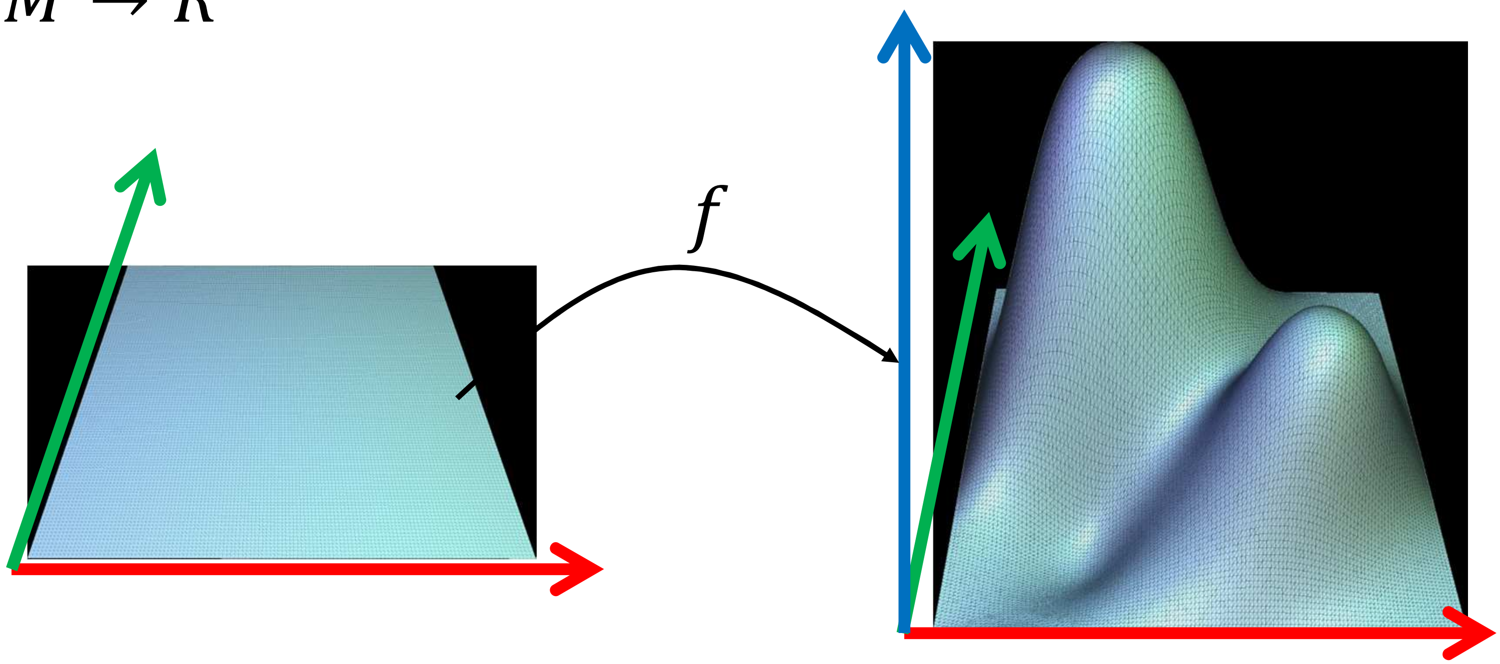
- Orientability
  - Any surface has a triangulation
  - Orient all triangles CW or CCW
  - **Orientability**: any two triangles sharing an edge have opposite directions on that edge.
  - Can distinguish in and out of the surface



Ants on a Mobius Strip by M.C. Escher

Morse Theory

- Investigates the topology of a surface by looking at *critical points* of a function on that surface. $\nabla f(p) = \left(\frac{\partial f}{\partial x}(p) \quad \frac{\partial f}{\partial y}(p)\right) = 0$
- $f: M \rightarrow R$



Morse Functions

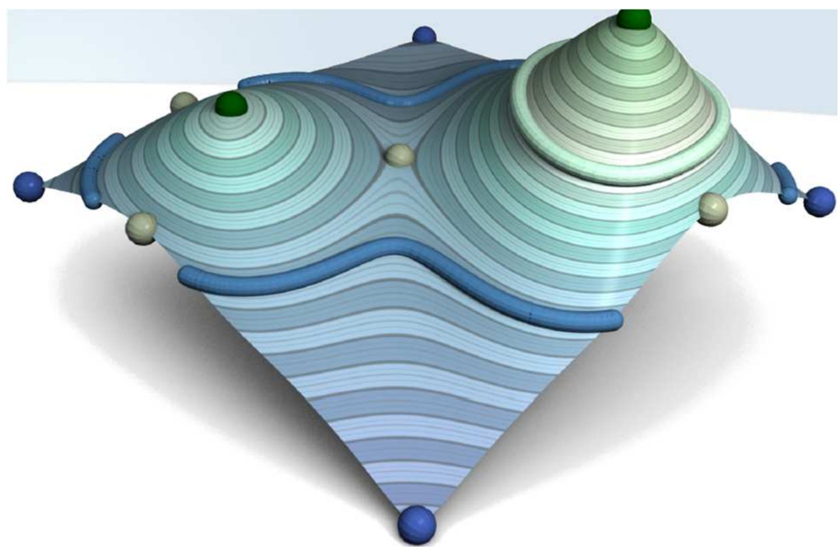
- A function f is a Morse function if
  - f is smooth
  - All critical points are isolated
  - All critical points are non-degenerate $\det(Hessian(\boldsymbol{p})) \neq 0$

$$Hessian\ f(\mathbf{p}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2}(\mathbf{p}) & \frac{\partial^2 f}{\partial x \partial y}(\mathbf{p}) \\ \frac{\partial^2 f}{\partial y \partial x}(\mathbf{p}) & \frac{\partial^2 f}{\partial y^2}(\mathbf{p}) \end{pmatrix}$$

  - A non-Morse function can be made Morse by adding small but random noise

Notion of Critical Points

- Points where ∇f vanishes
  - Minima, maxima, and saddles
  - Correspond to places where the topology of the function changes
  - The function behavior can be illustrated by isolines/level sets on the domain



- Level set of a given value i

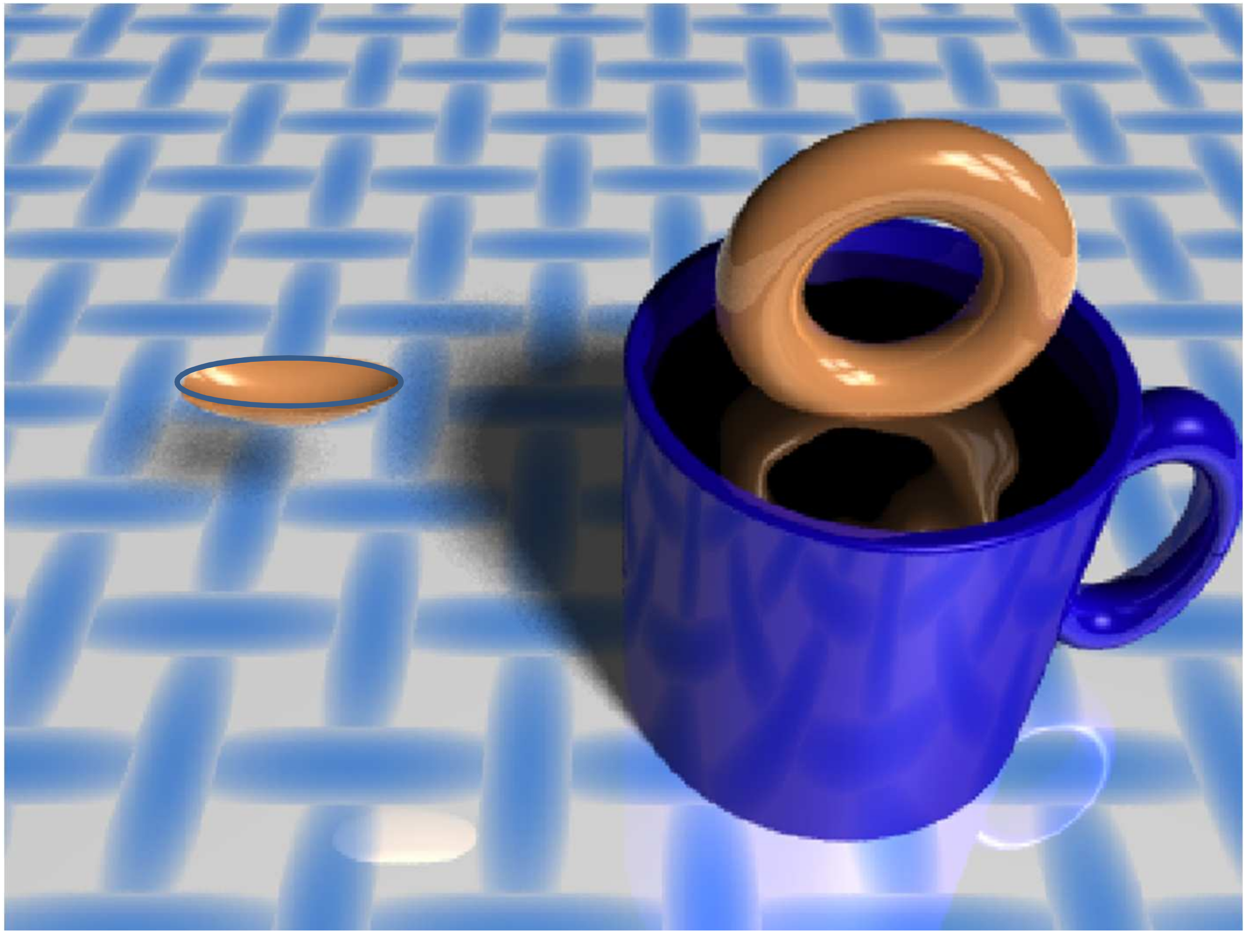
$f^{-1}(i) = \{p \in M \ \ |f(p) = i\}$

Example – dunking a doughnut

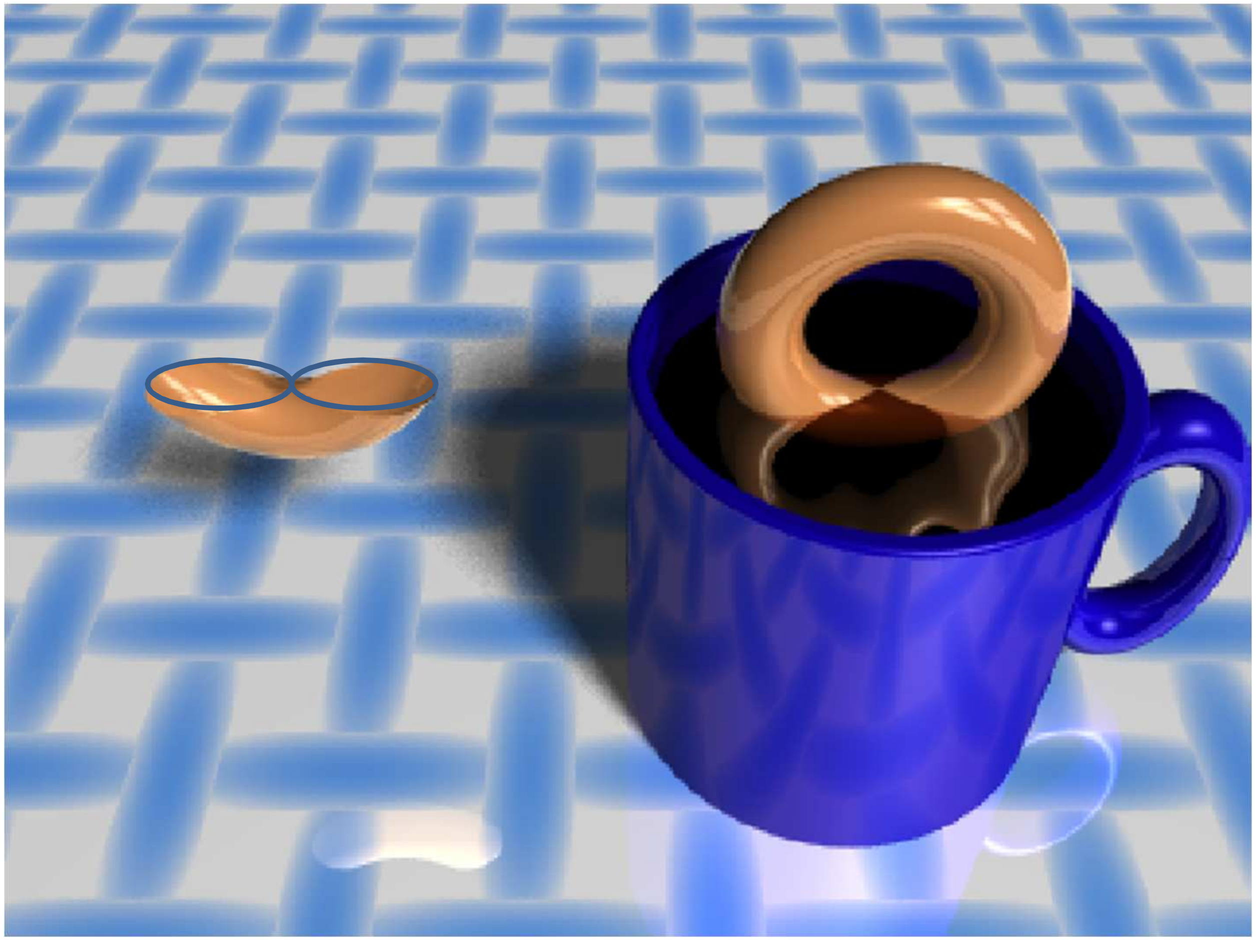
- $f(\boldsymbol{p}) = z$ (height function)

Shape analysis is a special case of scalar field analysis

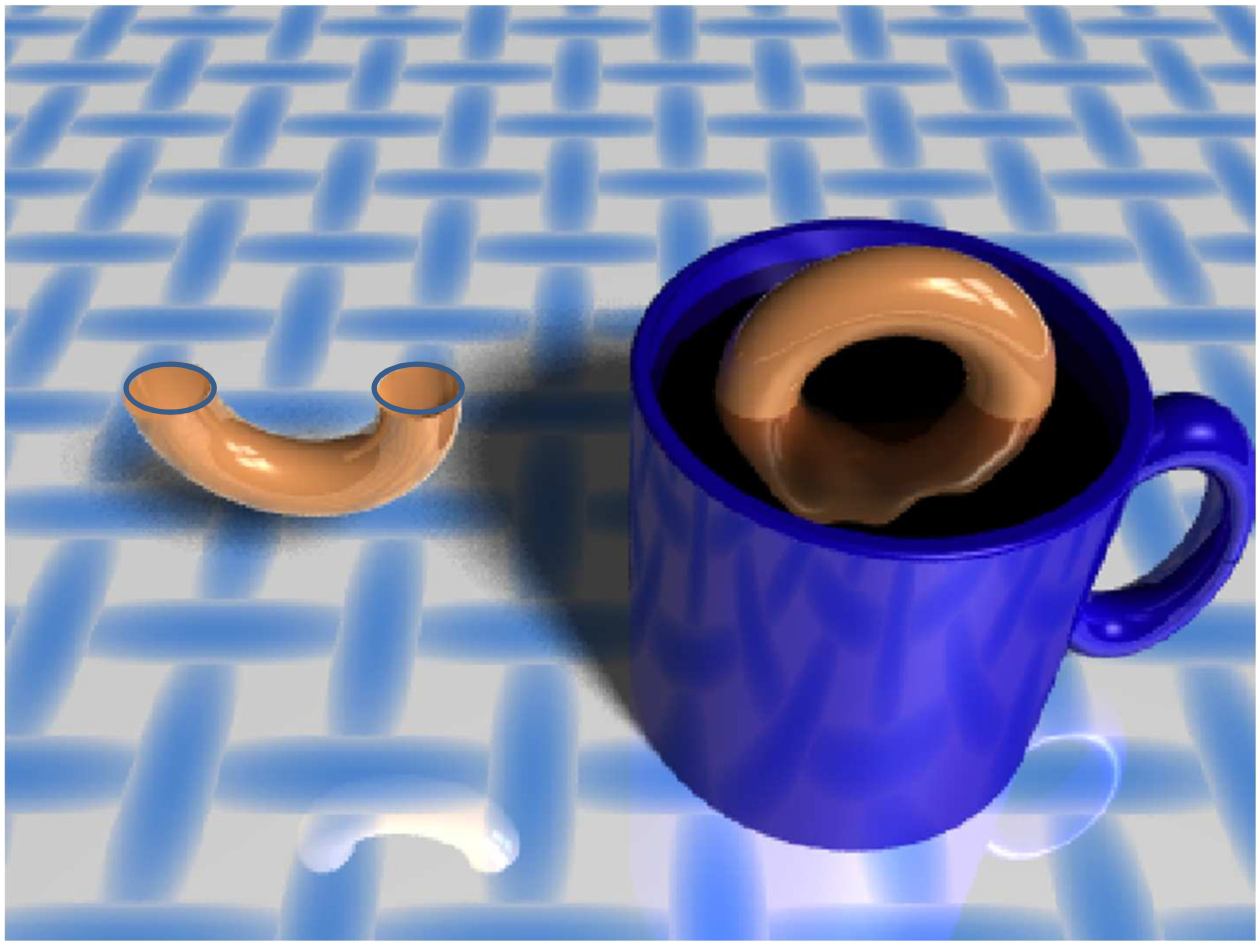
Example – dunking a doughnut



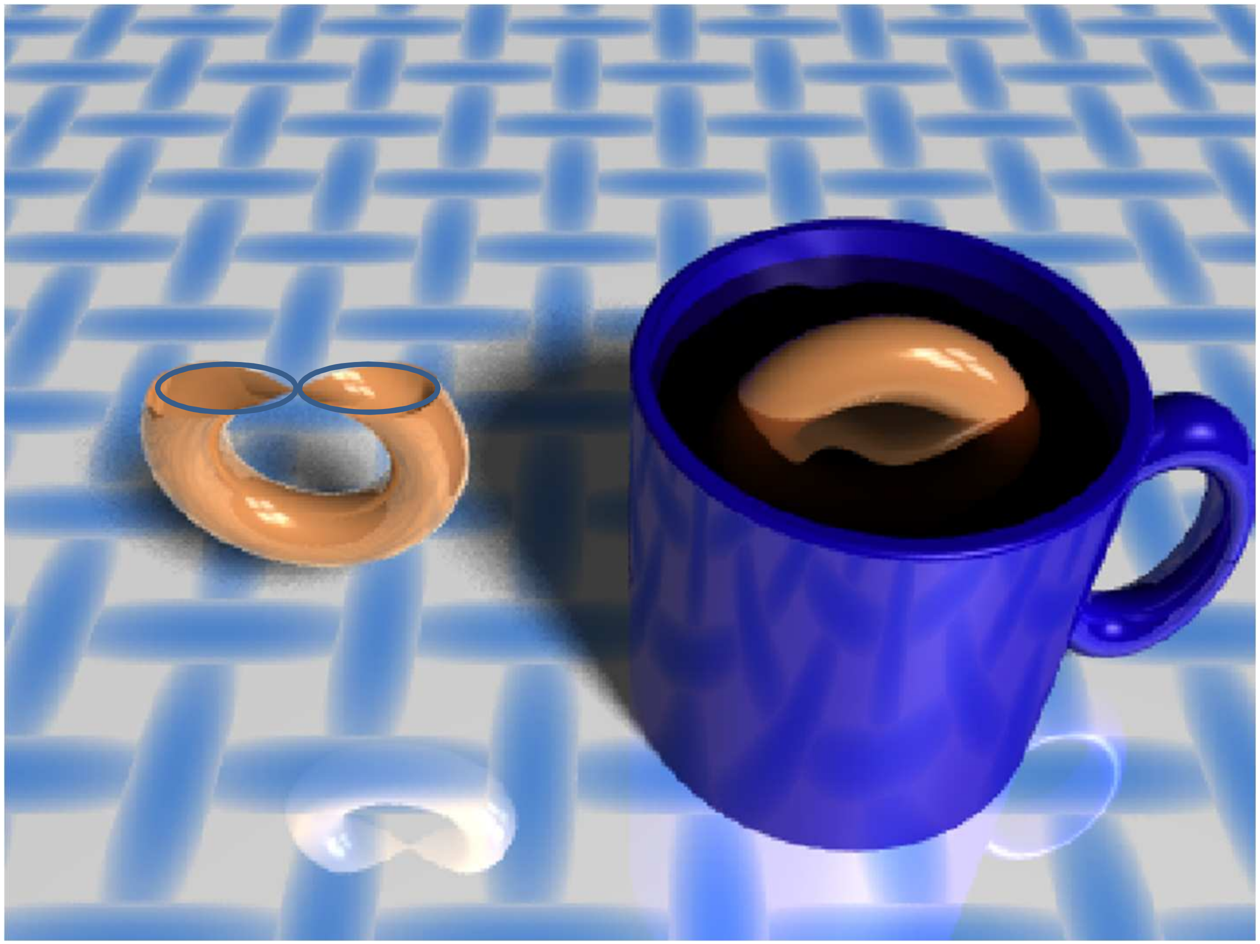
Example – dunking a doughnut



Example – dunking a doughnut



Example – dunking a doughnut



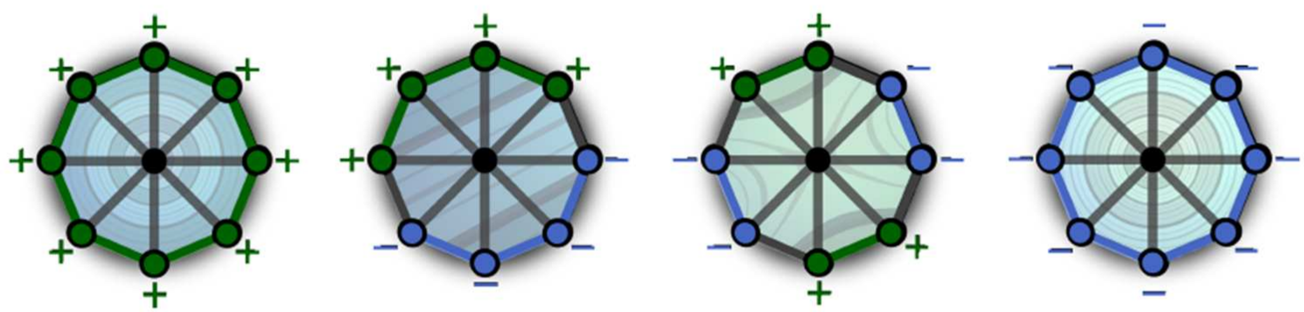
Example – dunking a doughnut

Notion of Critical Point

- Points where ∇f vanishes
  - Minima, maxim, and saddles
  - Topological changes
  - Piecewise linear interpolation
  - Barycentric coordinates on triangles
  - Only exist at vertices

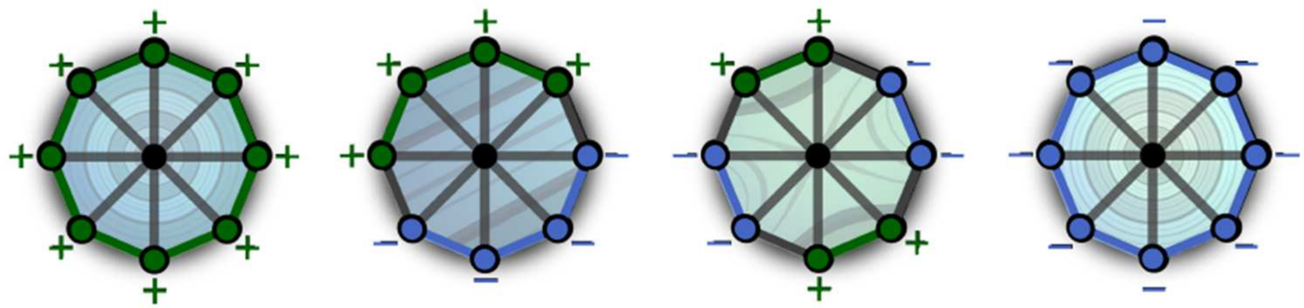
Notion of Critical Point

- Combinatorial identification
  - One-ring neighborhood (or a star) of a vertex v
    - Triangles and edges that adjacent to v



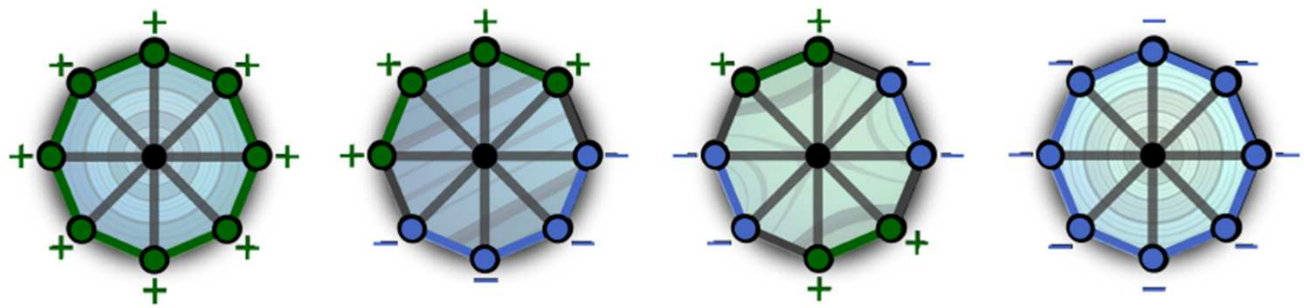
Notion of Critical Point

- Combinatorial identification
  - One-ring neighborhood (or a star) of a vertex v
    - Triangles and edges that adjacent to v
  - Link of a vertex
    - Edges of the star that do not adjacent to v



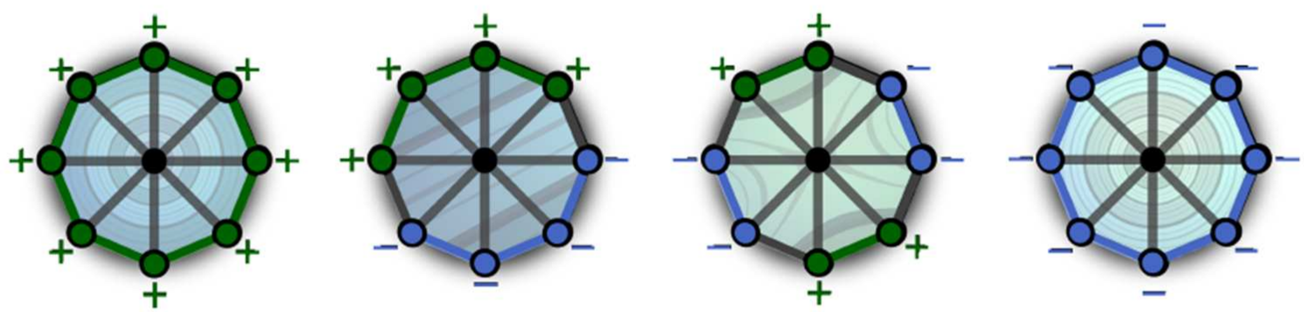
Notion of Critical Point

- Combinatorial identification
  - Lower link of v
    - Edges of the link of v whose function values are strictly smaller than $f(v)$



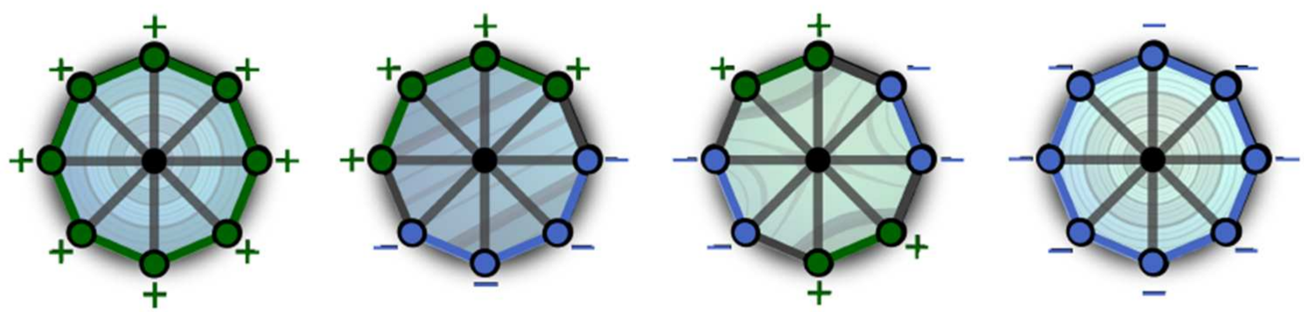
Notion of Critical Point

- Combinatorial identification
  - Lower link of v
    - Edges of the link of v whose function values are strictly smaller than $f(v)$
  - Upper link of v
    - Edges of the link of v whose function values are strictly larger than $f(v)$



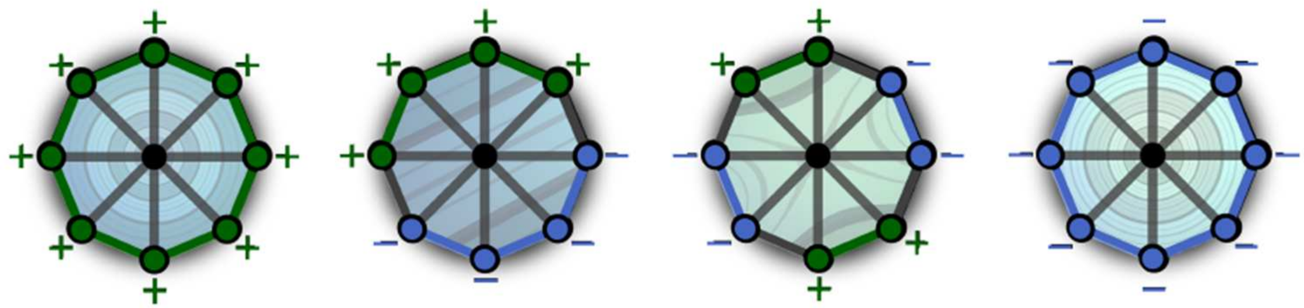
Notion of Critical Point

- Combinatorial identification
  - Minima
    - Empty lower link



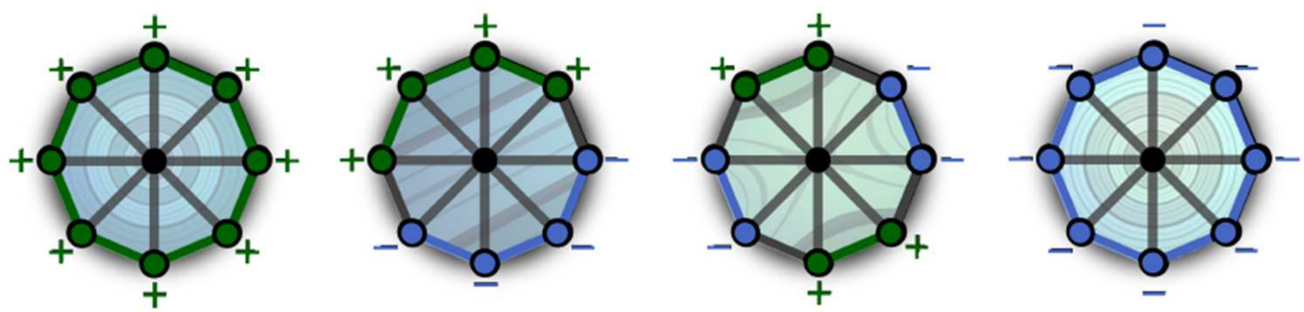
Notion of Critical Point

- Combinatorial identification
  - Minima
    - Empty lower link
  - Maxima
    - Empty upper link



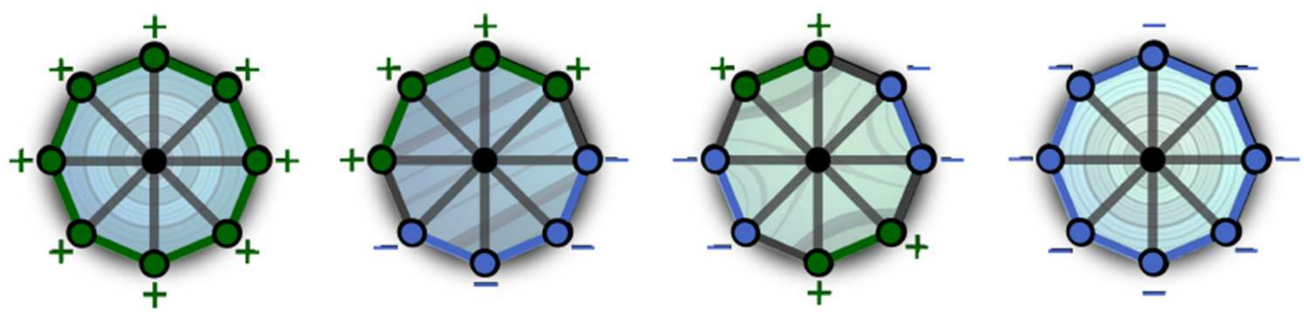
Notion of Critical Point

- Combinatorial identification
  - Minima
    - Empty lower link
  - Maxima
    - Empty upper link
  - Regular point
    - Lower and upper links both simply connected



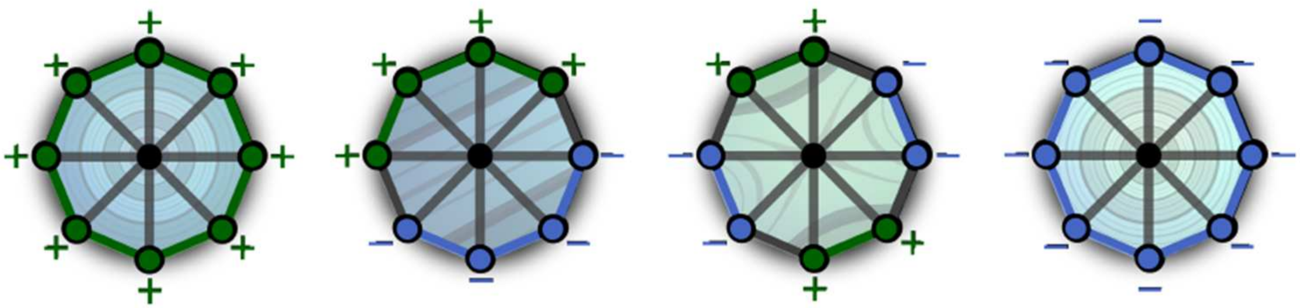
Notion of Critical Point

- Combinatorial identification
  - Everything else
    - Saddle
      - Multiple connected lower and upper links
  - Works in arbitrary dimension if replace triangles, edges, and vertices with *simplices*



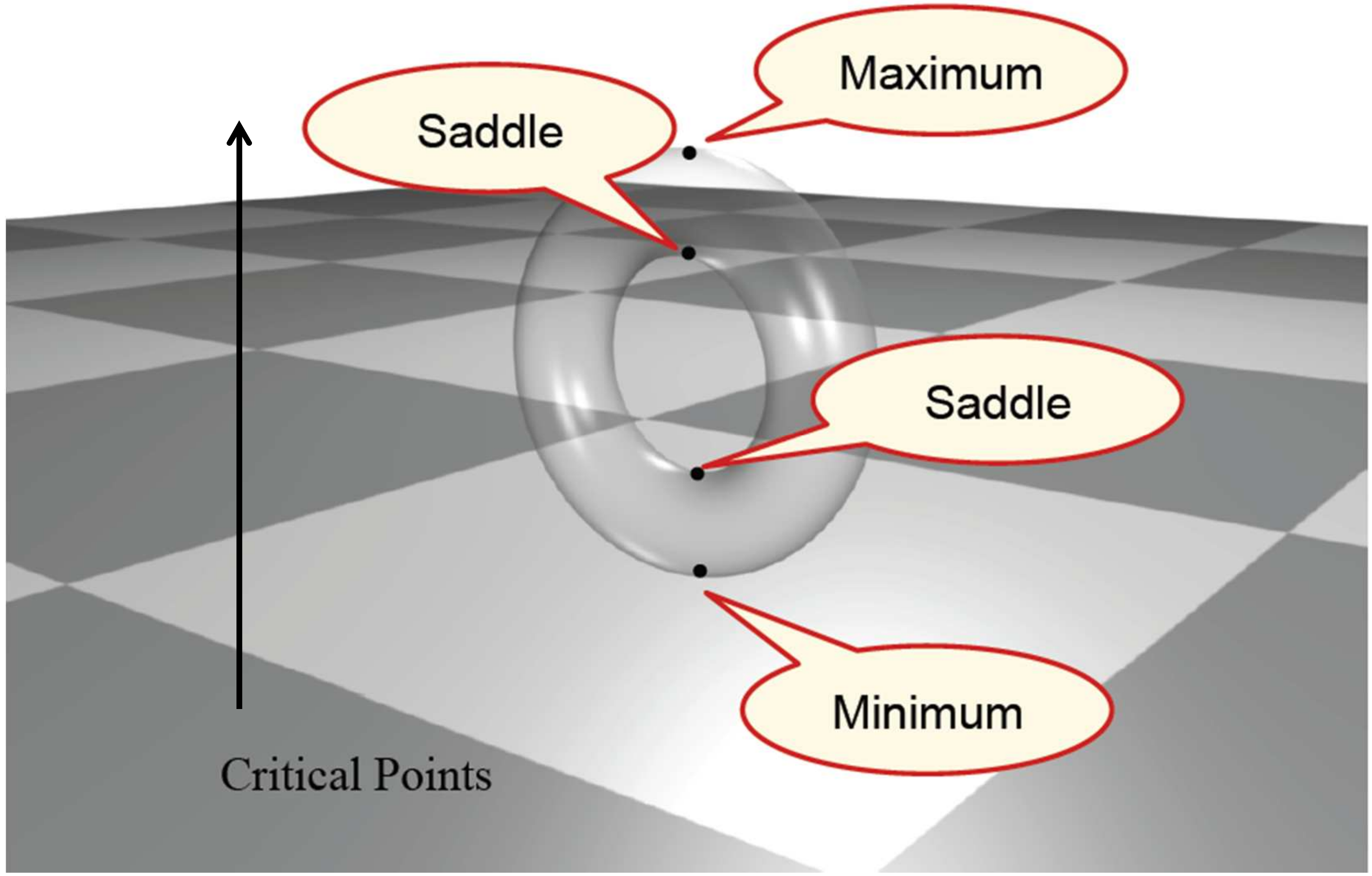
Notion of Critical Point

- Combinatorial identification
  - Everything else
    - Saddle
      - Multiple connected lower and upper links
  - Works in arbitrary dimension if replace triangles, edges, and vertices with *simplices*
  - *Value of a critical point – critical value*



How Does it Work?

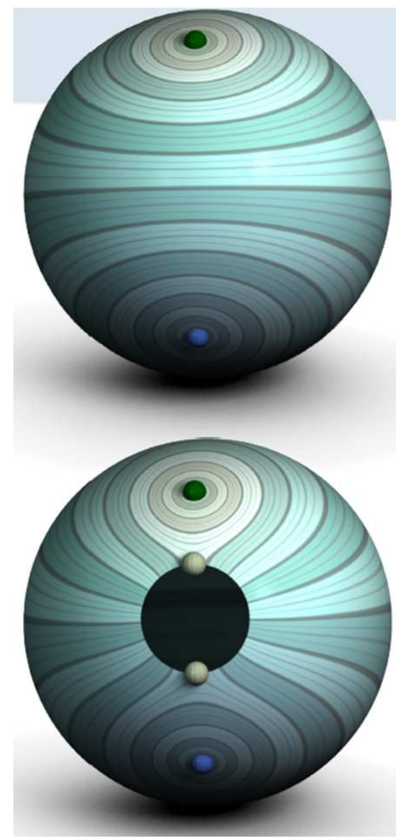
Morse function is defined as the height function (i.e. z coordinates)
Level sets obtaining by sweeping along Z direction



Review of Level Sets

- Given a domain M of dimension d
  - Level set of a given value i, $f^{-1}(i) = \{p \in M \ |f(p) = i\}$
- If i is not a critical value
  - $f^{-1}(i)$ is a $(d-1)$ manifold
- If M is closed, $f^{-1}(i)$ is closed.

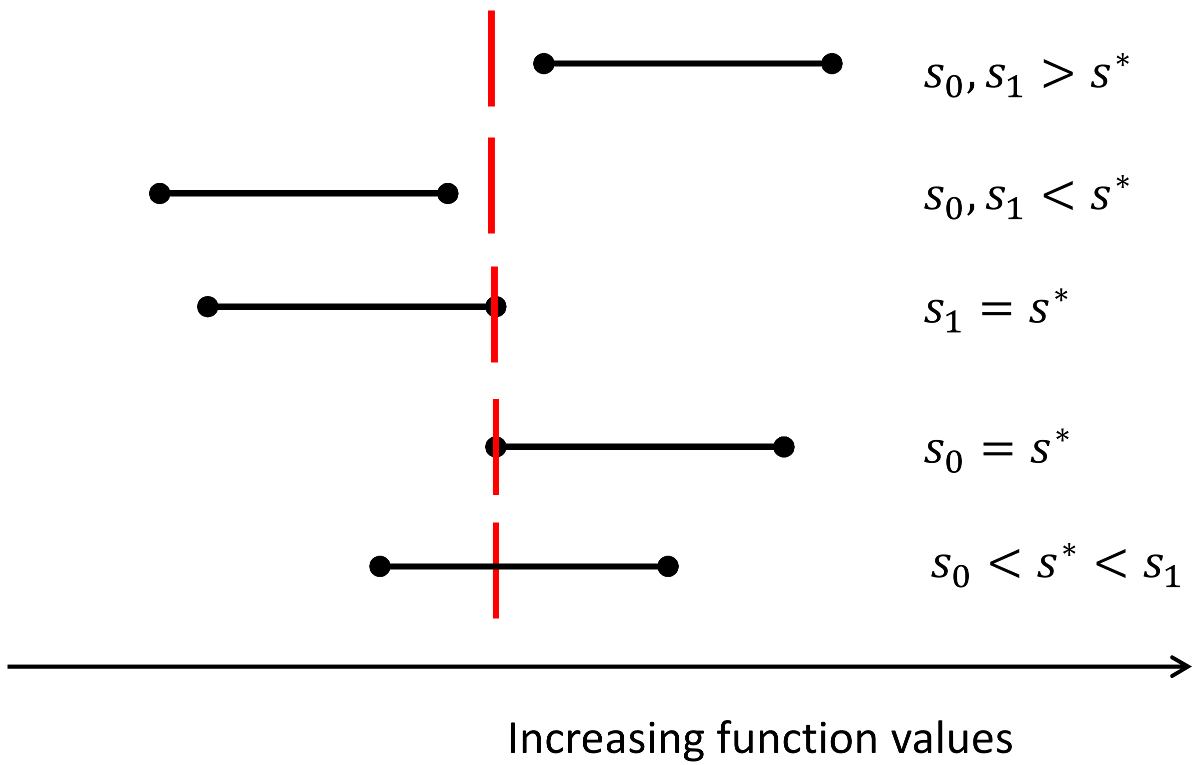
Otherwise it may be open



Level Set Extraction

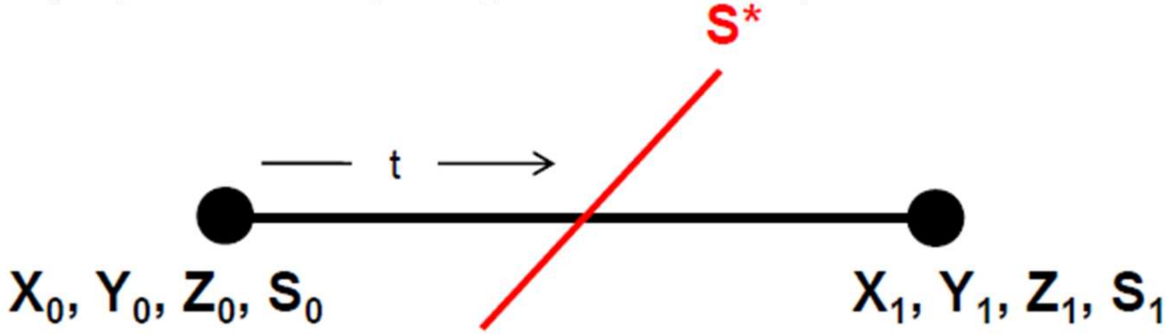
- Given a triangular mesh X and a scalar function f, the level set of a given value s^* is compute as follows
  - First, determine the intersections of the level set $f^{-1}(s^*)$ with the edges of X
  - Second, traverse through all triangles and connect the obtained intersections

Which Edges Intersect with $f^{-1}(s^*)$?



Determine Intersections with Edges

Does $f^{-1}(s^*)$ cross any edges of this square?



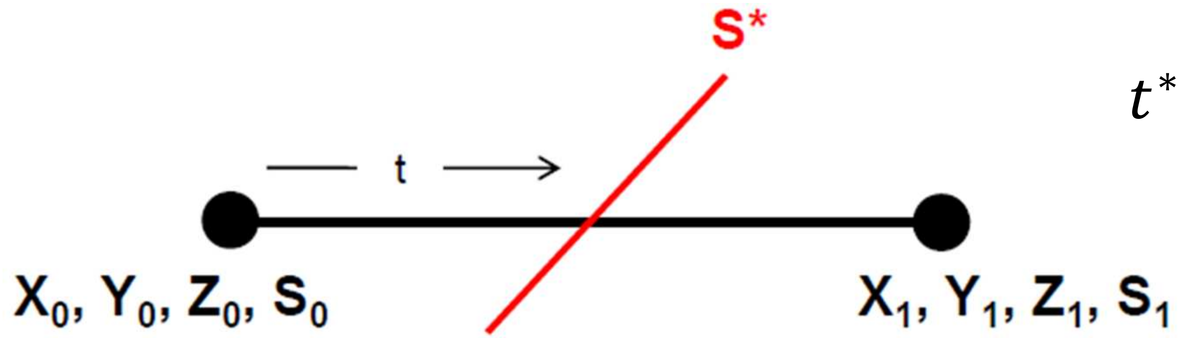
Linearly interpolating the scalar value from node 0 to node 1 gives:

$$S = (1-t)S_0 + tS_1 = S_0 + t(S_1 - S_0) \quad \textit{where } 0. \leq t \leq 1.$$

Setting this interpolated S equal to $S*$ and solving for t gives:

$$t^* = \frac{S^* - S_0}{S_1 - S_0}$$

Determine Intersections with Edges



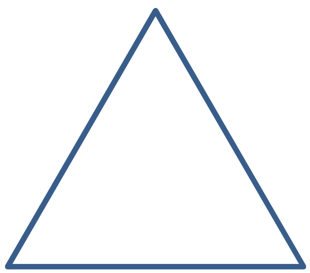
$$t^* = \frac{S^* - S_0}{S_1 - S_0}$$

If 0. ≤ t* ≤ 1., then S* crosses this edge. You can compute where S* crosses the edge by using the same linear interpolation equation you used to compute S*. You will need that for later visualization.

$$x^* = (1 - t^*)x_0 + t^*x_1$$
$$y^* = (1 - t^*)y_0 + t^*y_1$$

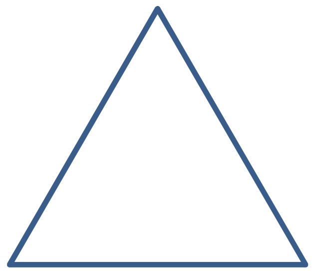
Connect Intersections

No intersection

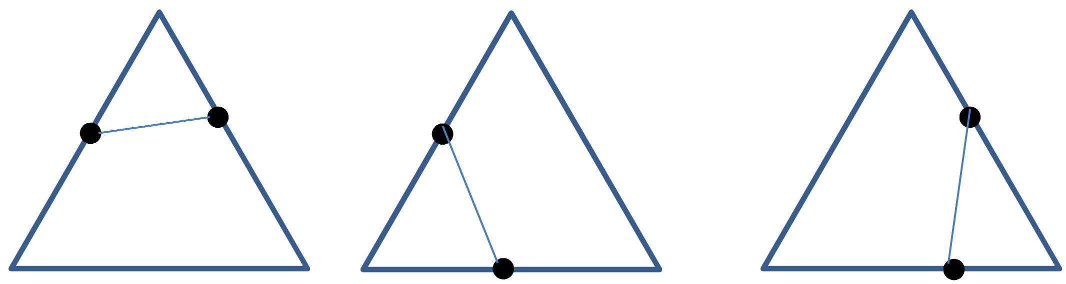


Connect Intersections

No intersection

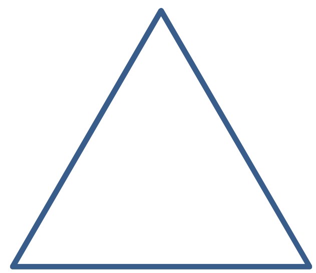


Two intersections

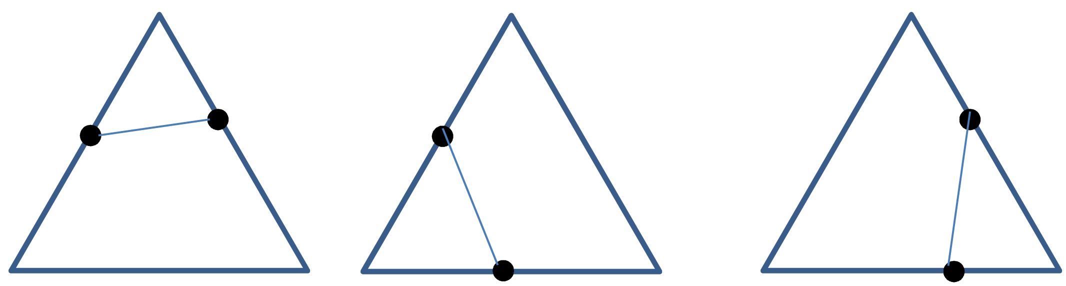


Connect Intersections

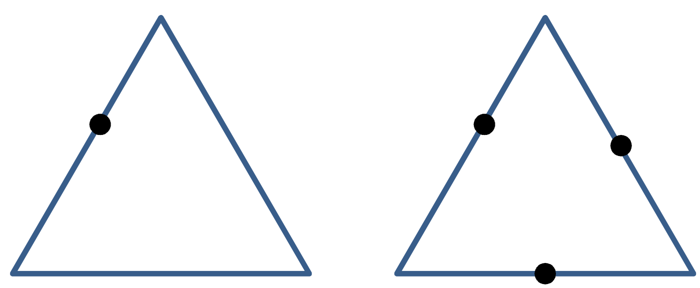
No intersection



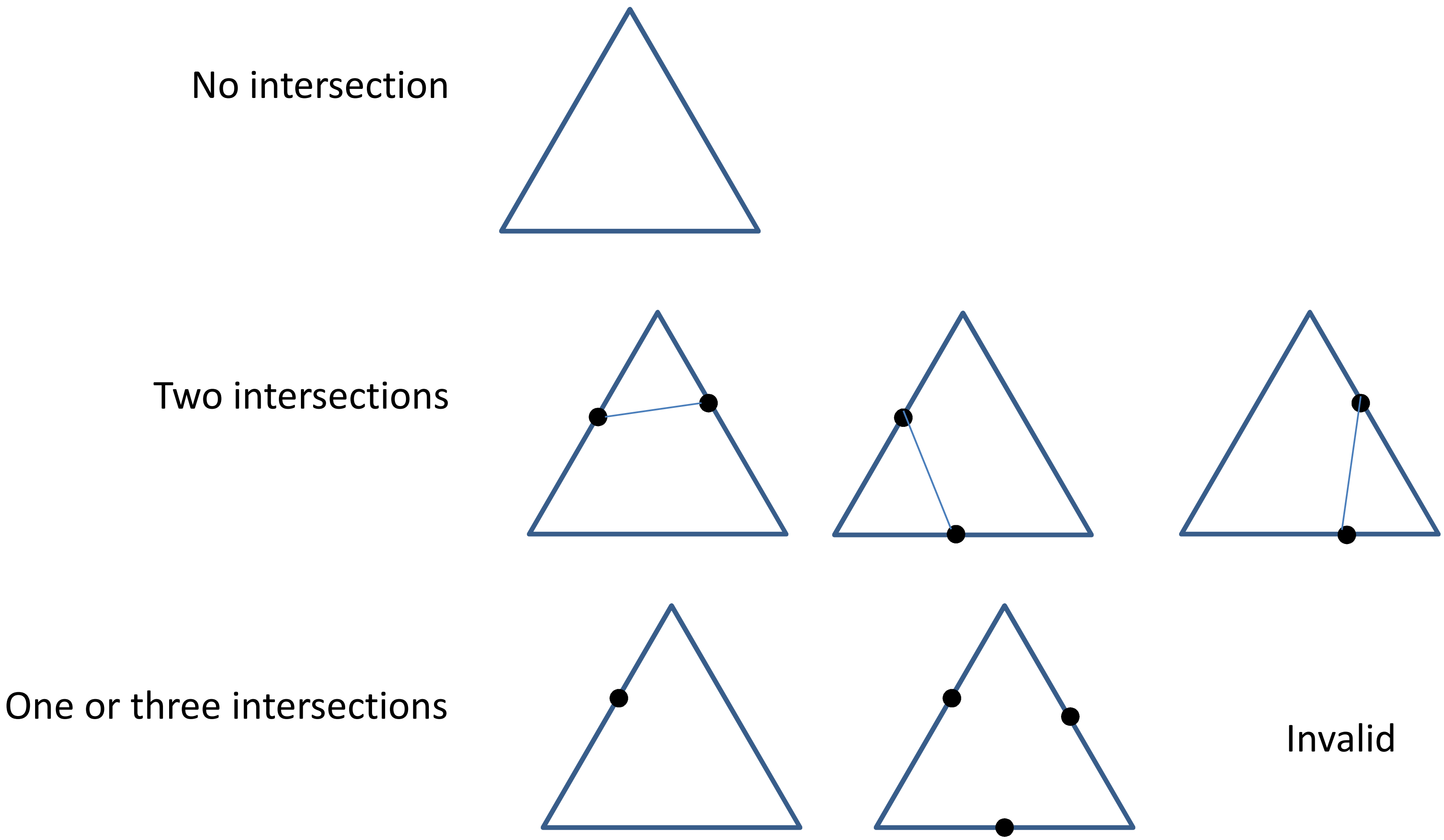
Two intersections



One or three intersections



Connect Intersections



Reeb Graph

- Given a Morse function
  - $f: M \rightarrow R$

Reeb Graph

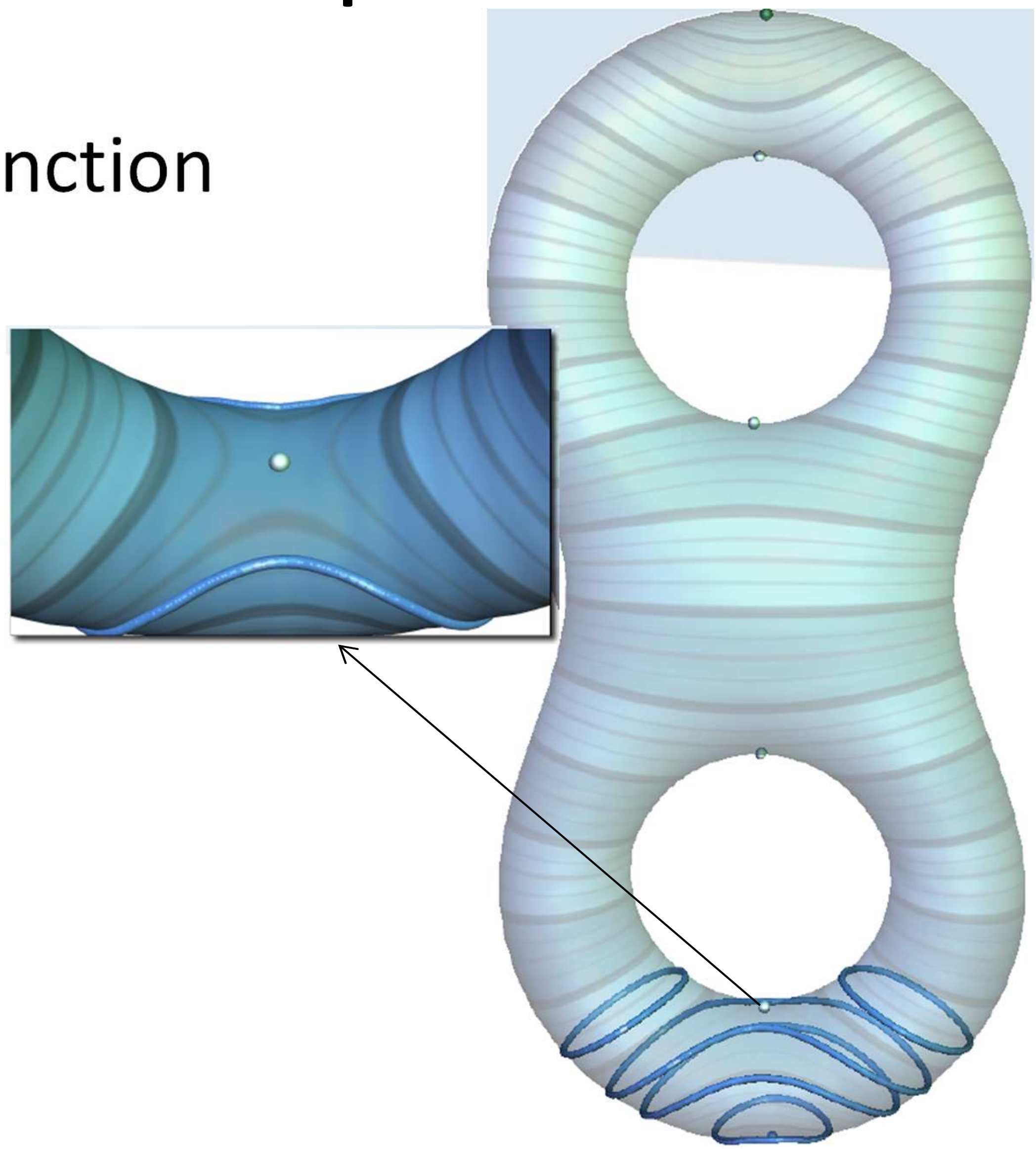
- Given a Morse function
  - $f: M \rightarrow R$

Reeb Graph

- Given a Morse function
  - $f: M \rightarrow R$

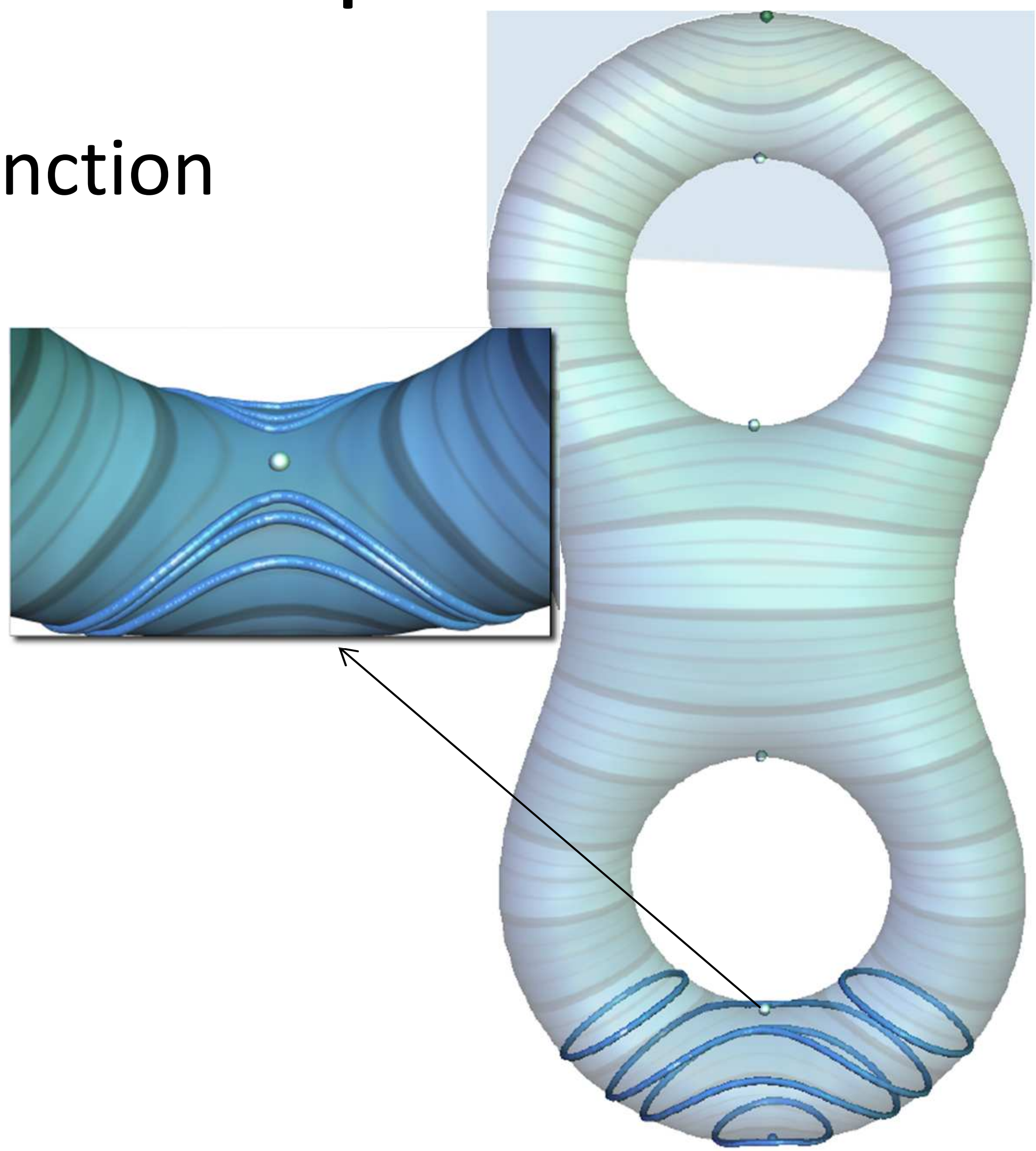
Reeb Graph

- Given a Morse function
  - $f: M \to R$



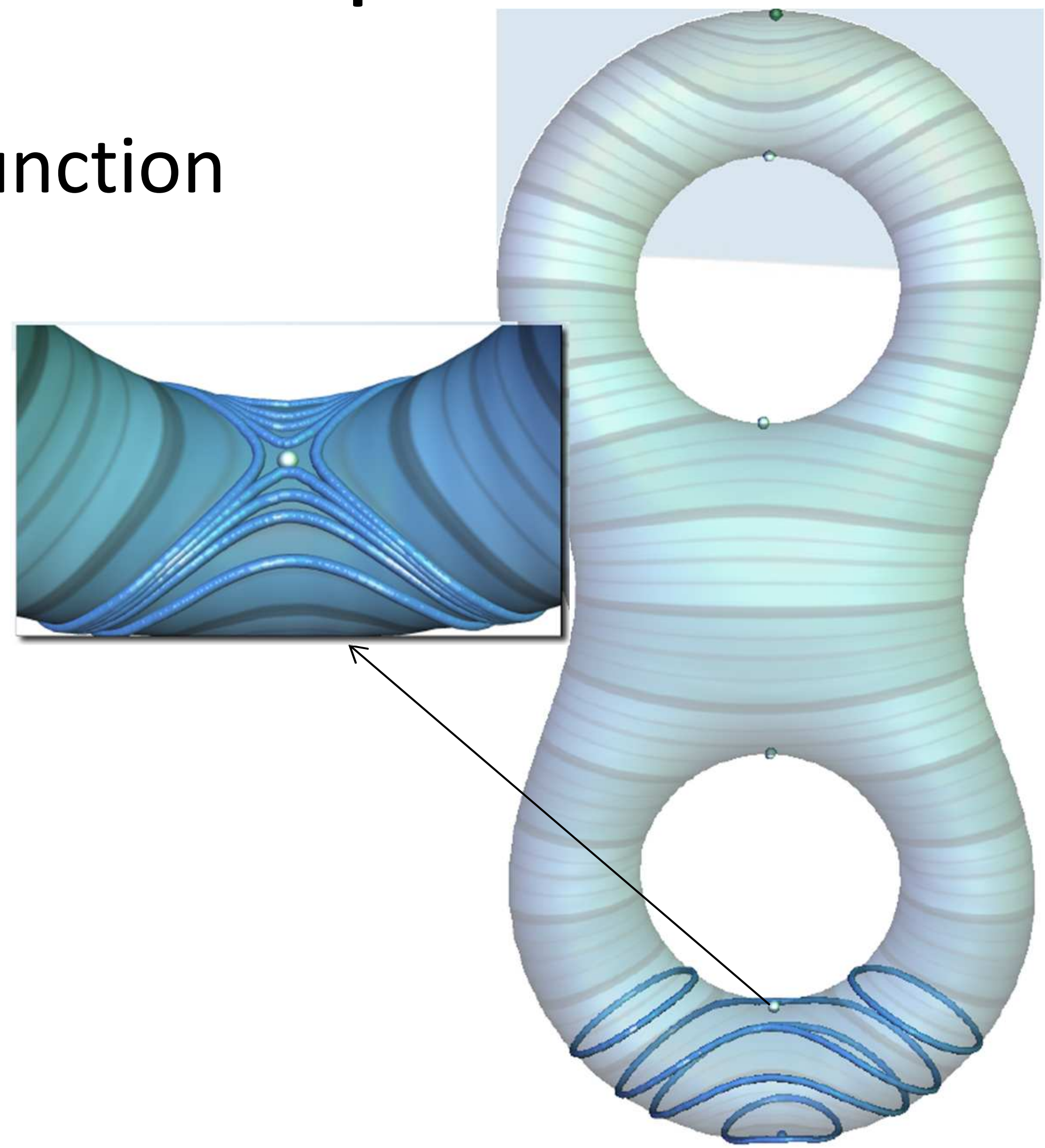
Reeb Graph

- Given a Morse function
  - $f: M \rightarrow R$



Reeb Graph

- Given a Morse function
  - $f: M \rightarrow R$



Reeb Graph

- Given a Morse function
  - $f: M \to R$

Reeb Graph

- Given a Morse function
  - $f : M \to R$

Reeb Graph

- Given a Morse function
  - $f: M \to R$

Reeb Graph

- Given a Morse function
  - $f: M \rightarrow R$

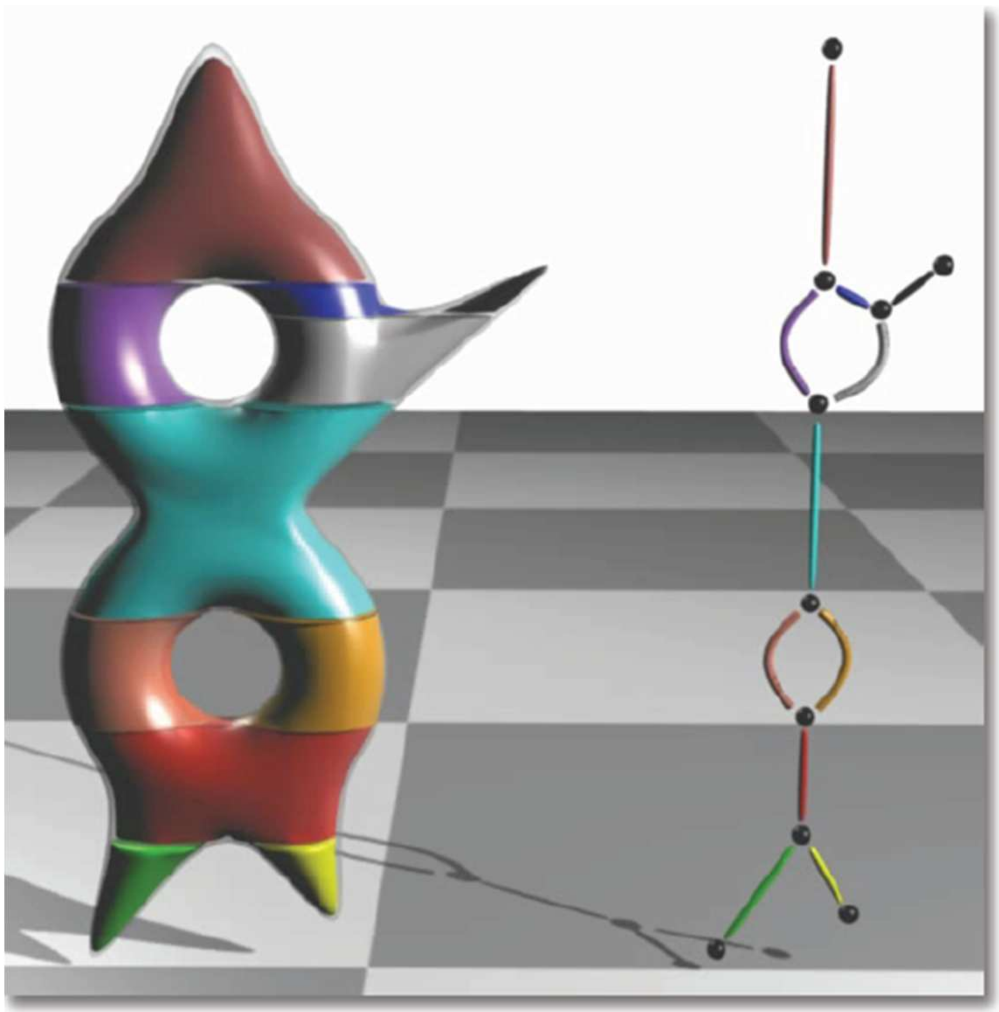
Reeb Graph

- Given a Morse function
  - $f: M \rightarrow R$

Reeb Graph

- Given a Morse function
  - $f: M \to R$
  - Reeb graph $R(f)$
    - Contour retraction of M under f

Reeb Graph

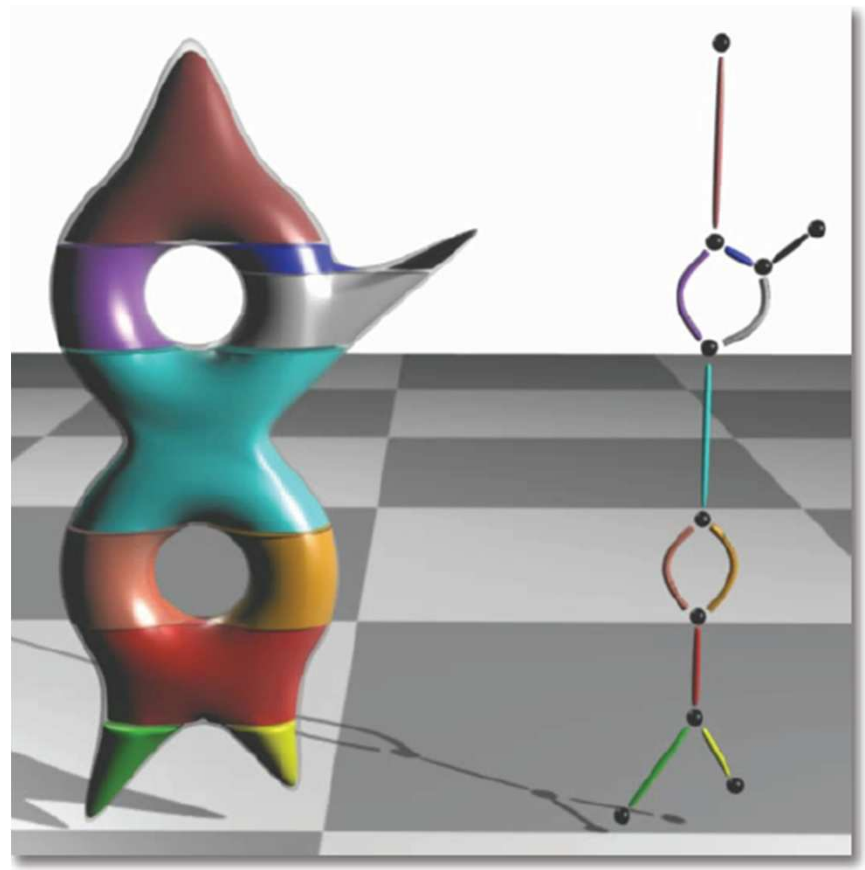


- Vertices of the graph are critical points
- Arcs of the graph are connected components (cylinders in domain)of the level sets of *f*, contracted to points

Reeb Graph Properties

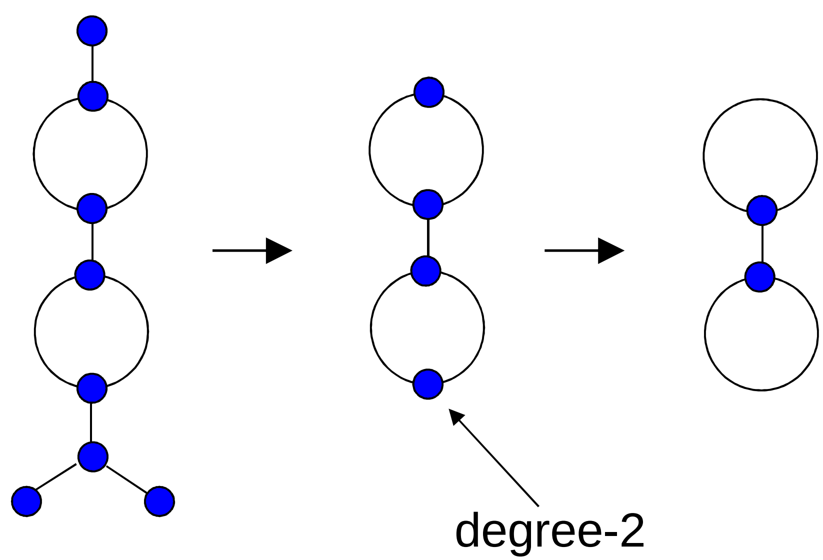
- Continuous 1-dimensional simplical complex
- Extrema: valence 1
- Saddles in 2D: valence 3 or 4 (boundary)
- Saddles in nD: valence 2 or 3

- In practice:
  - Arcs: collection of vertices
  - Simply connected domains: Carr00
  - Otherwise: Pascucci07, Tierny09, Parsa12



Reeb graphs and genus

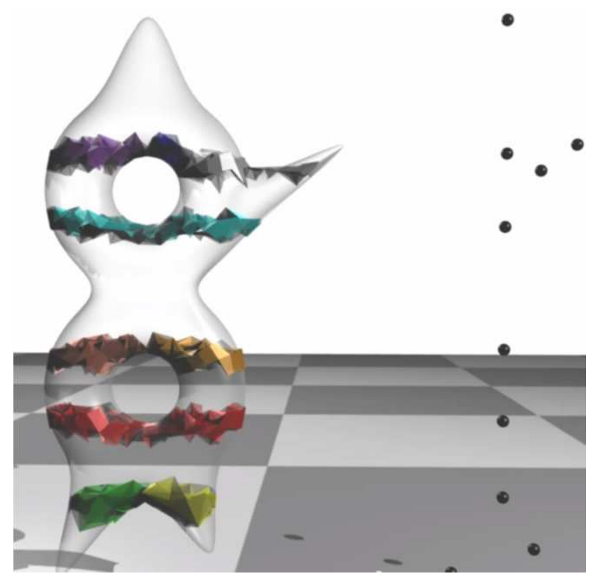
- The number of loops in the Reeb graph is equal to the surface genus
- To count the loops, simplify the graph by contracting degree-1 vertices and removing degree-2 vertices



Compute Reeb Graph Using Cylinder Maps

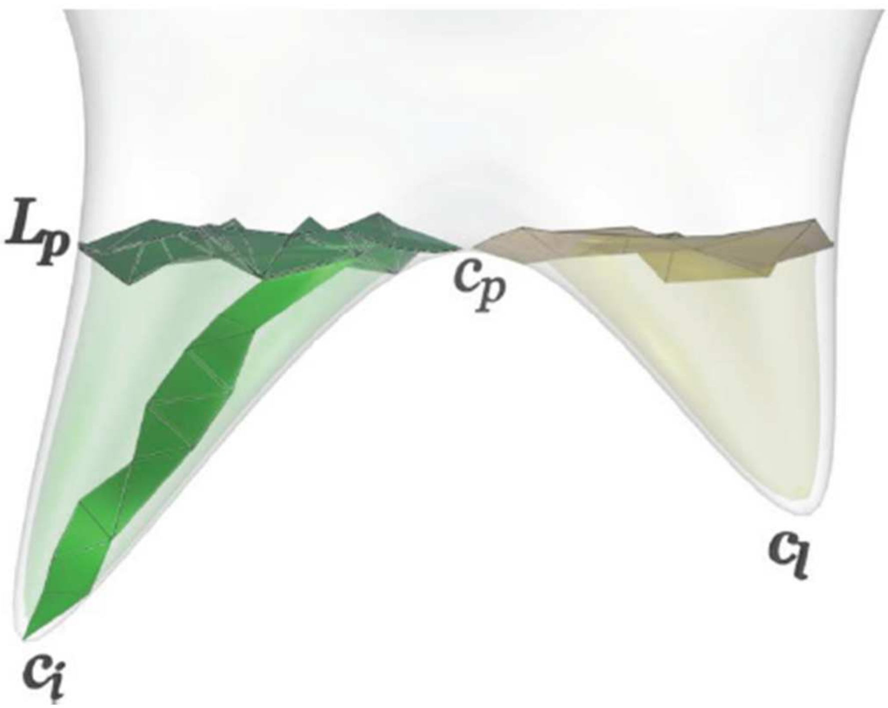
- Cylinders in domain map to arcs in Reeb graph
- Two step algorithm
  - Step I: Locate critical points
    - Sort the critical points based on their scalar value
    - compute the critical level set

    corresponding to each critical point



[Natarajan 2011]

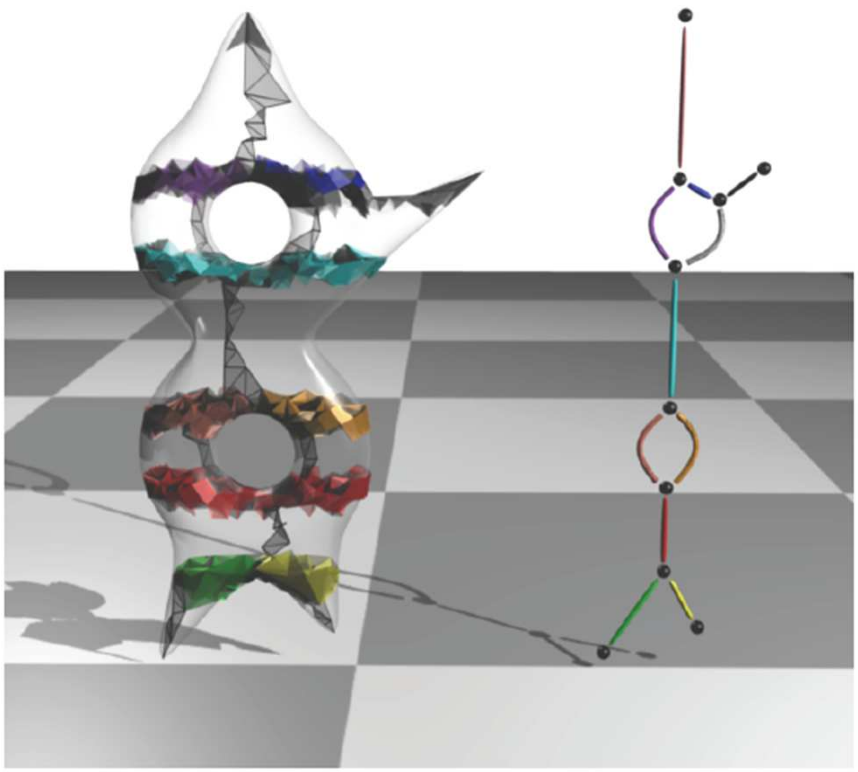
Step II - Connect Critical Points



- Cylinder represents evolution of one level set component
- Trace all level set components within a cylinder in an iteration

[Natarajan 2011]

Step II - Connect Critical Points

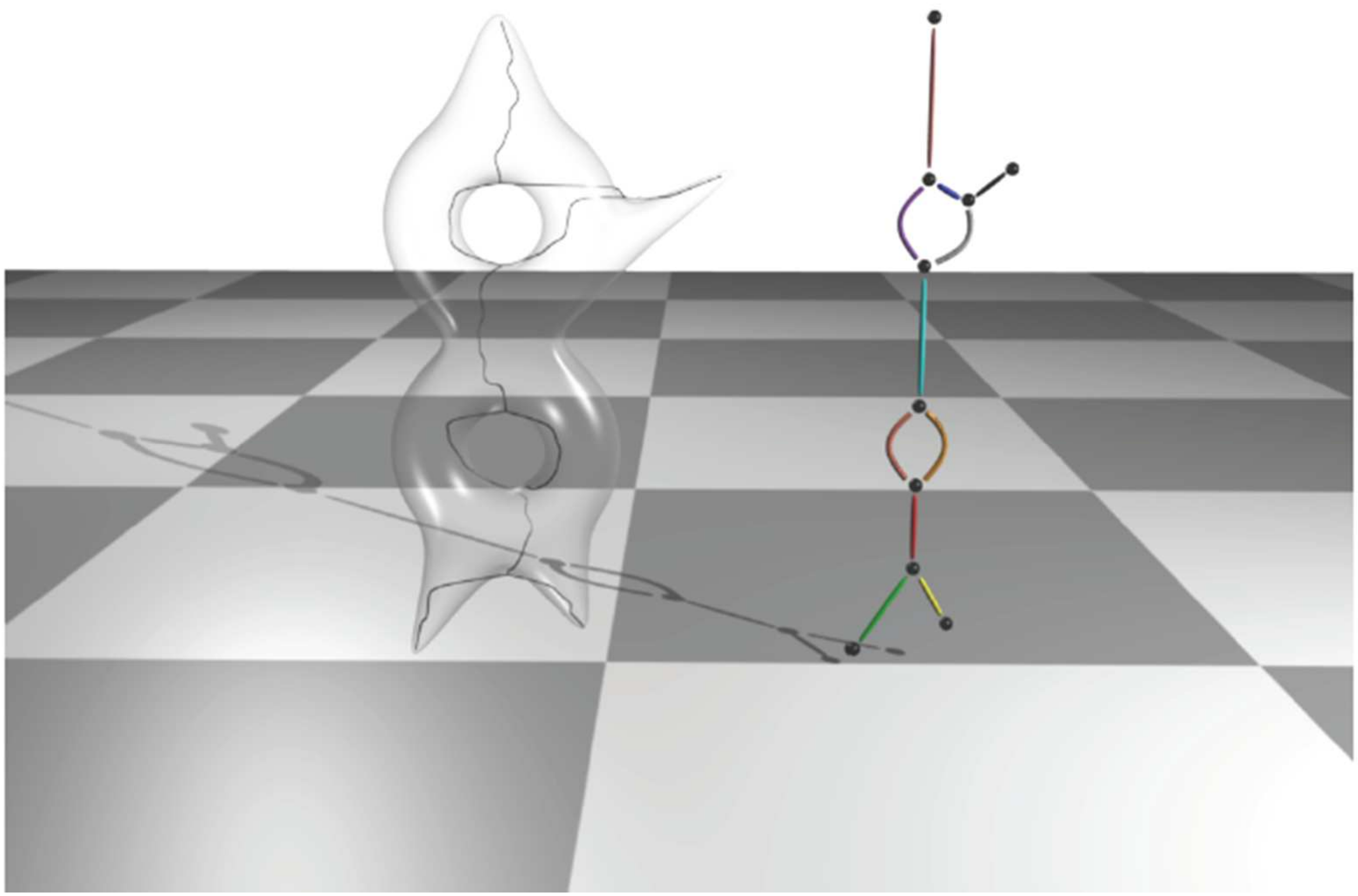


- Is-graph: stores adjacencies between triangles
- For each component
  - Start with a triangle in the component attaching to the upper link of a critical point
  - Traverse the edges until a triangle in a critical level set is reached
  - Insert corresponding arc in the Reeb graph

[Natarajan 2011]

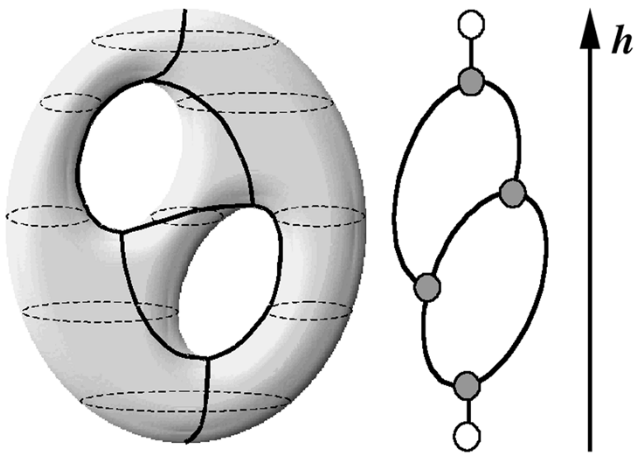
Reeb Graph Visualization

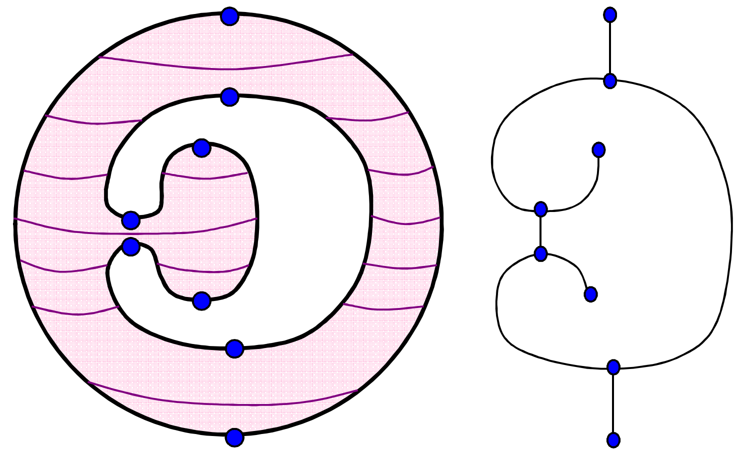
- Embedded layout

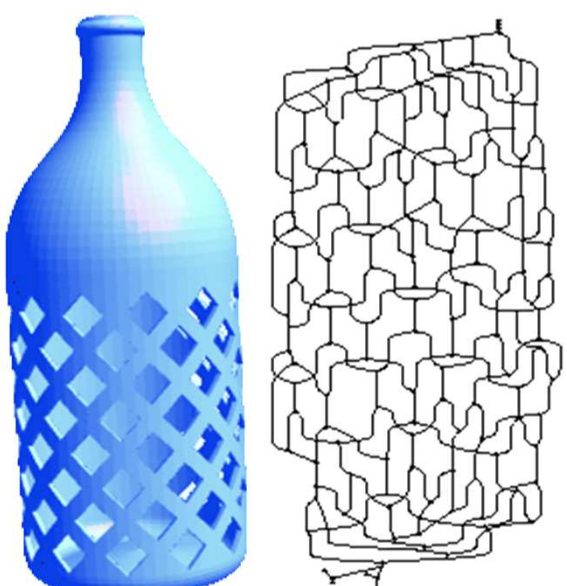


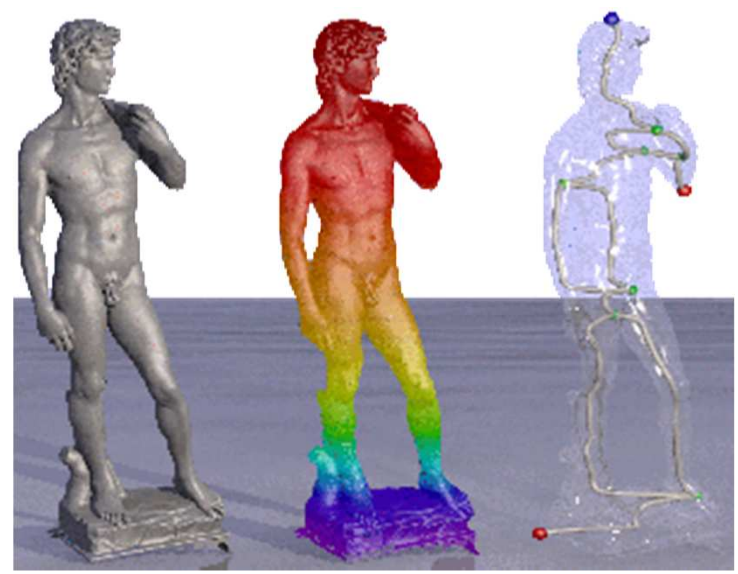
[Natarajan 2011]

Some More Reeb Graph Examples



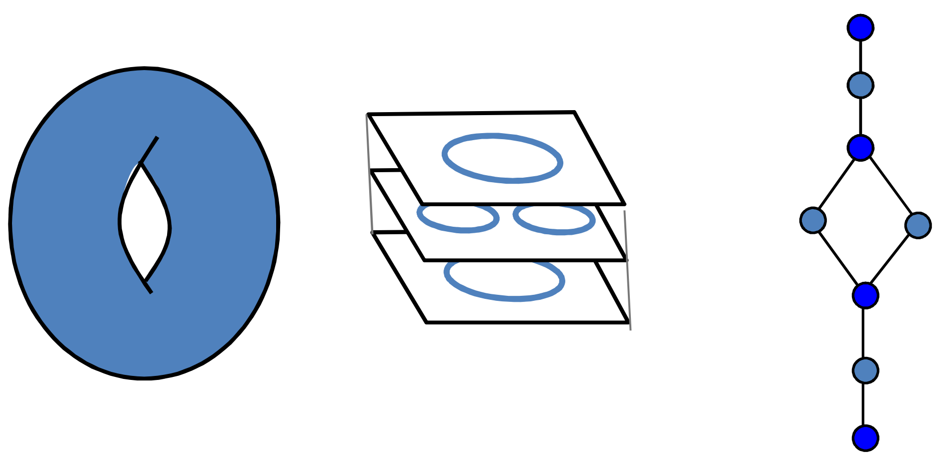






Discretized Reeb Graph

- Take the critical points and "samples" in between
- Robust because we know that nothing happens between consecutive critical points

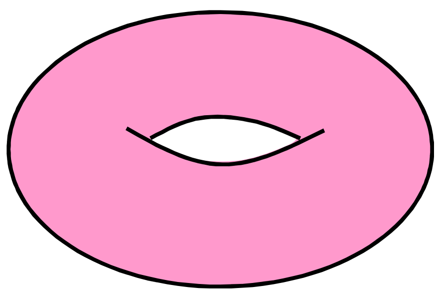


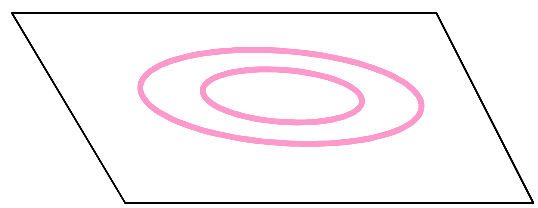
Reeb graphs for Shape Matching

- Reeb graph encodes the behavior of a Morse function on the shape
- Also tells us about the topology of the shape
- Take a meaningful function and use its Reeb graph to compare between shapes!

Choose the right Morse function

- The height function $f(\mathbf{p}) = z$ is not good enough – not rotational invariant
- Not always a Morse function





Average geodesic distance

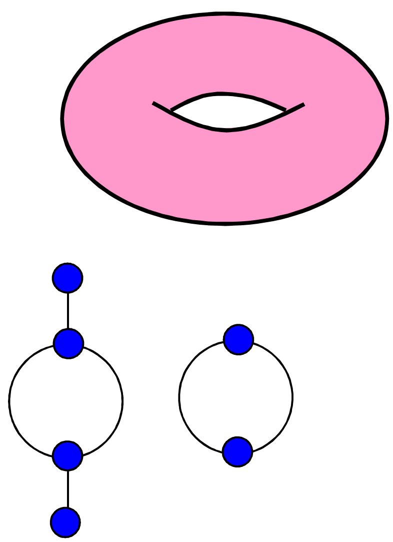
- The idea of [Hilaga et al. 01]: use geodesic distance for the Morse function!

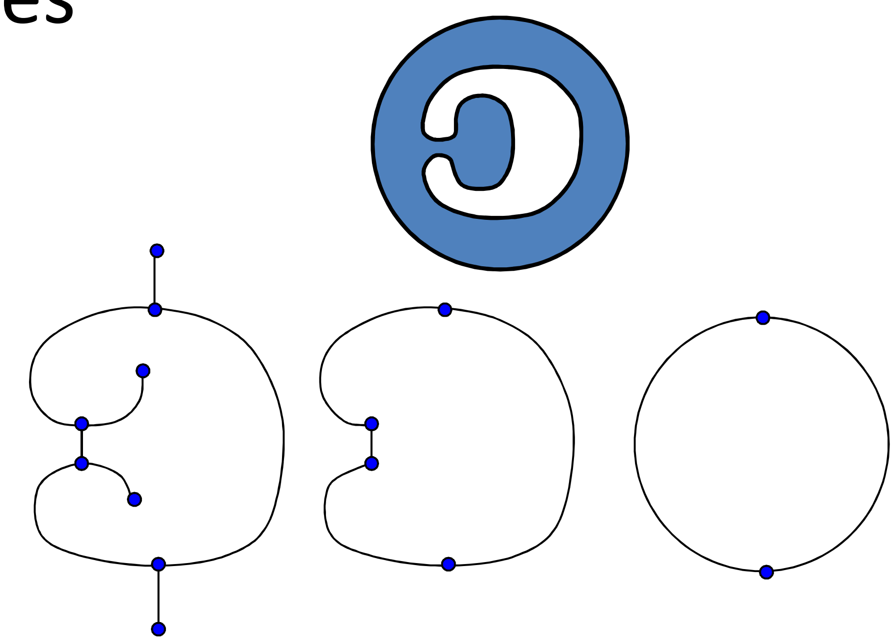
$$g(\mathbf{p}) = \int_M \text{geodist}(\mathbf{p}, \mathbf{q}) dS$$

$$f(\mathbf{p}) = \frac{g(\mathbf{p}) - \min_{\mathbf{q} \in M} g(\mathbf{q})}{\max_{\mathbf{q} \in M} g(\mathbf{q})}$$

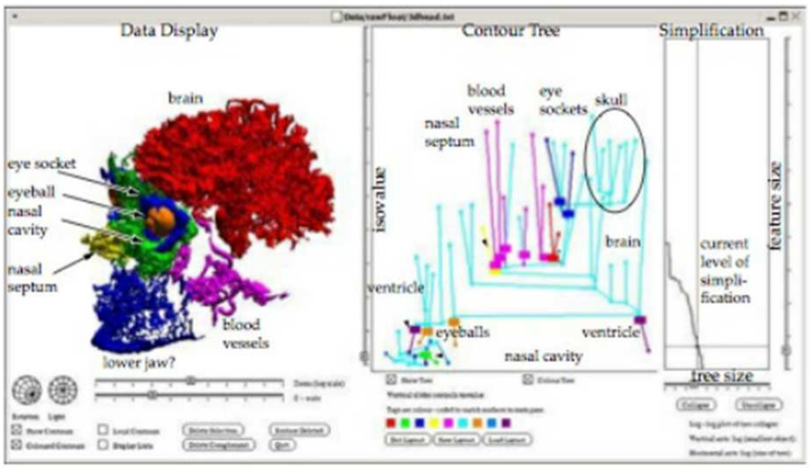
Multi-res Reeb graphs

- Hilaga et al. use multiresolutional Reeb graphs to compare between shapes
- Multiresolution hierarchy – by gradual contraction of vertices

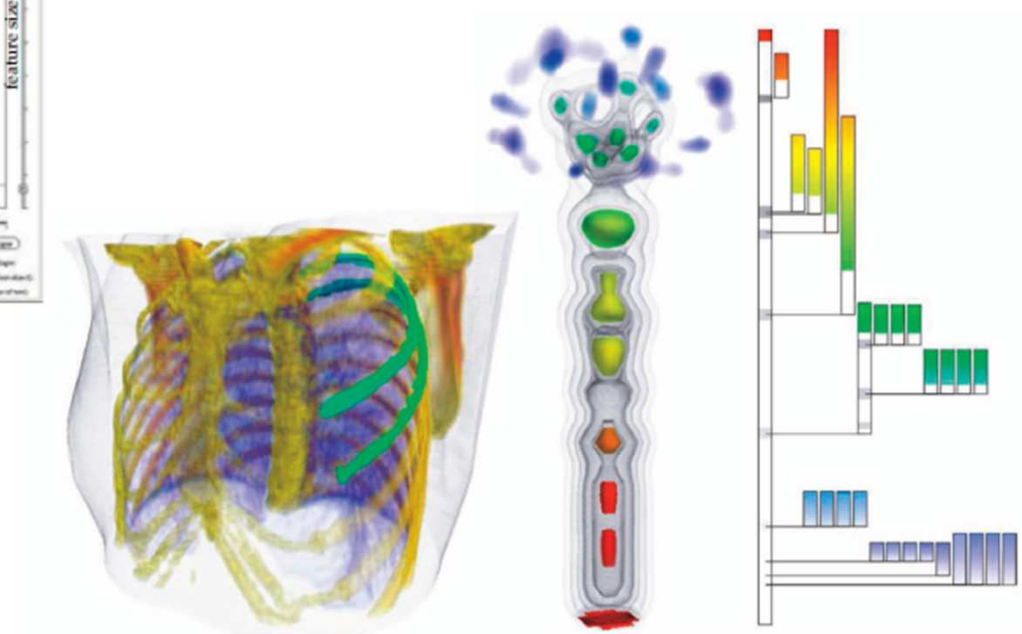




Reeb Graph Applications

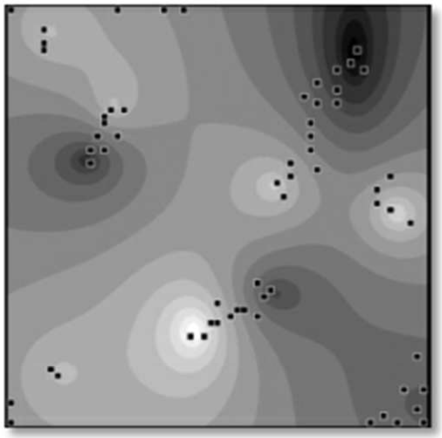


Flexible isosurfaces
[Carr et al., IEEE VIS 2004]



Topology controlled volume rendering
[Weber et al., TVCG 2007]

Reeb Graph Applications

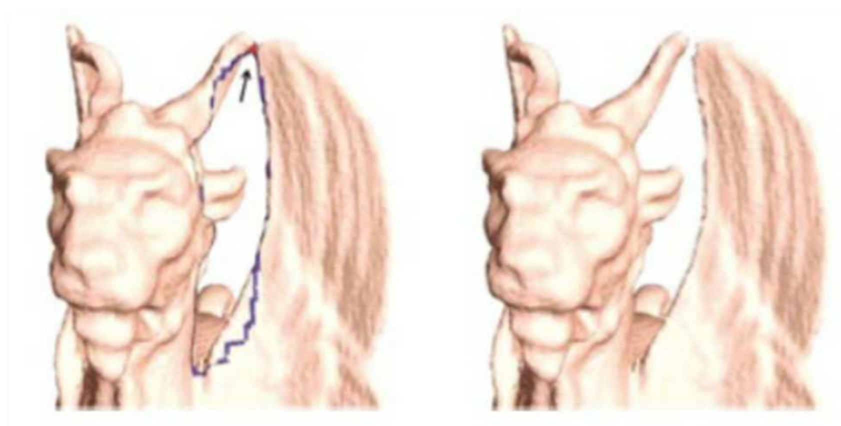


Seed sets for isosurfaces

[van Kreveld et al., SoCG 1997]

Surface parameterization

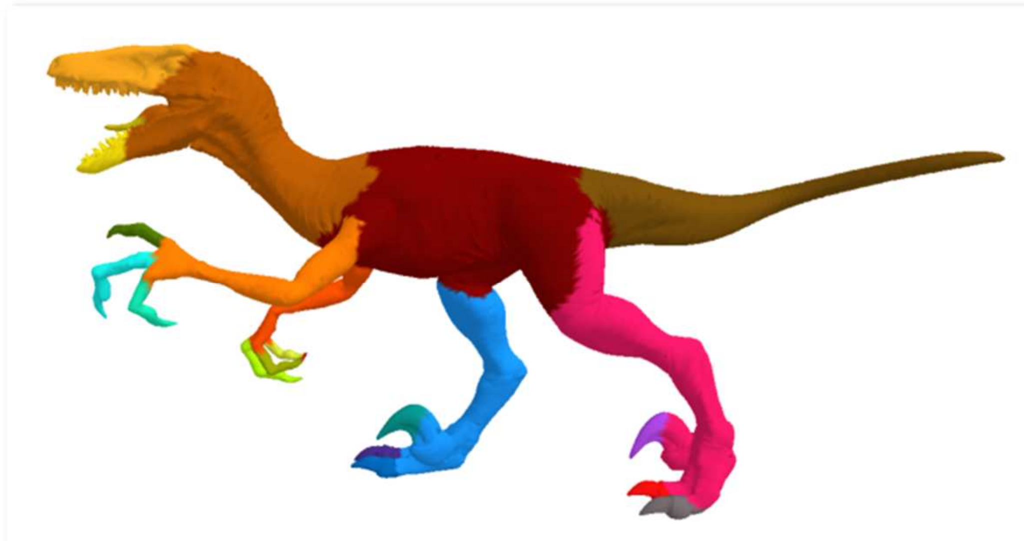
[Zhang et al., TOG 2005]

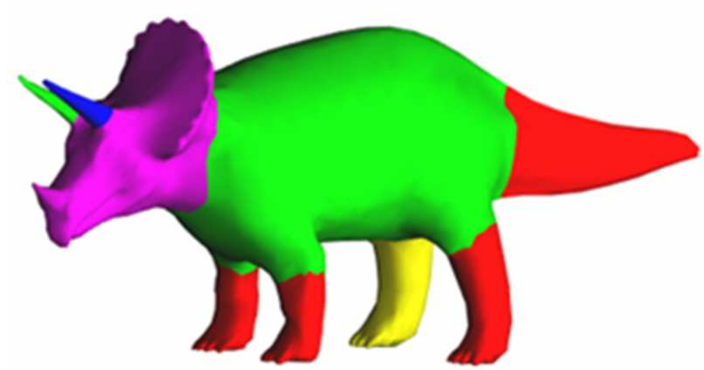


Topological simplification and cleaning

[Wood et al., TOG 2004; Pascucci et al., SIGGRAPH 2007]

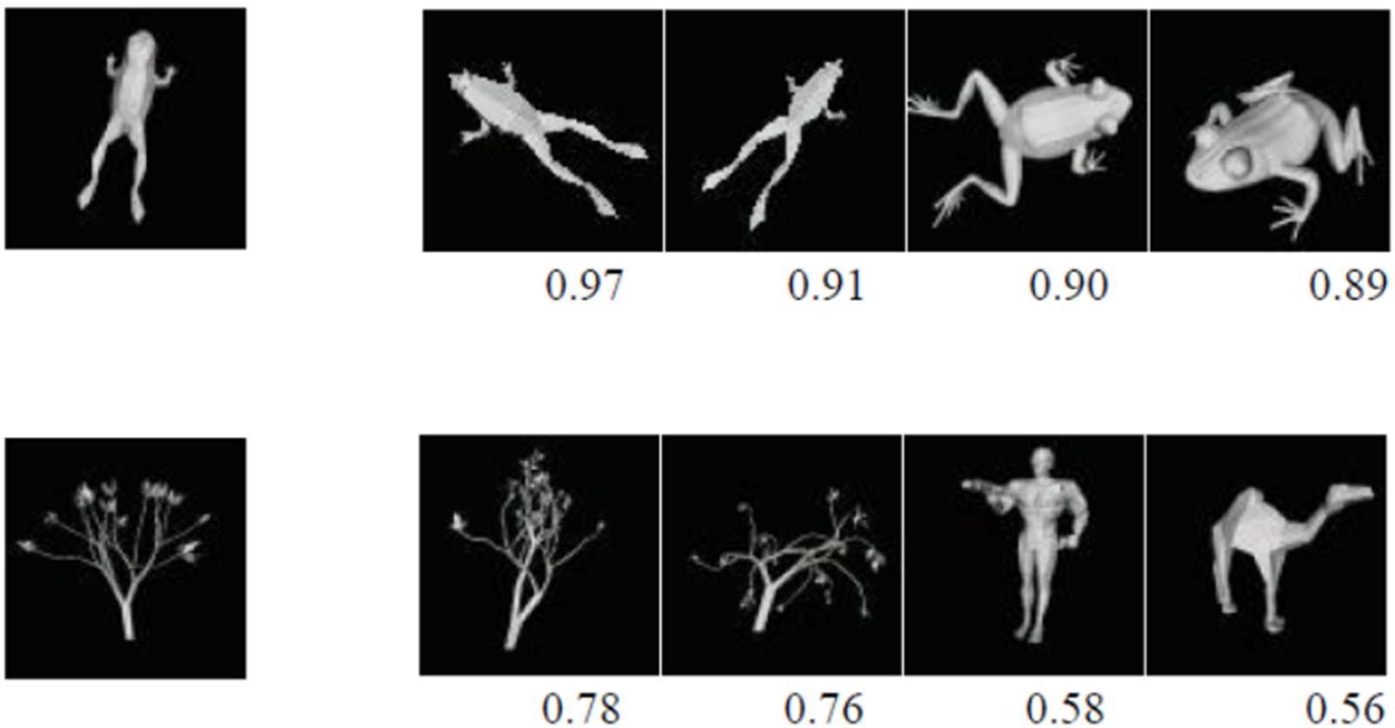
Reeb Graph Applications





Mesh segmentation [Zhang et al. 2003]

Reeb Graph Applications



Topology-based shape matching

[Hilaga et al., SIGGRAPH 2001]

Additional Reading for Reeb Graph

- V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas, "Robust On-line Computation of Reeb Graphs: Simplicity and Speed", *ACM Transactions on graphics,* pp. 58.1-58.9, Proceedings of SIGGRAPH 2007.
- S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno, "Reeb graphs for shape analysis and applications", Theor. Comput. Sci. 392(1-3): 5-22, 2008.
- J. Tierny, A. Gyulassy, E. Simon, and V. Pascucci, "Loop surgery for volumetric meshes: Reeb graphs reduced to contour trees". IEEE TVCG, Vol. 15 (6) : 1177-1184, 2009.
- H. Doraiswamy and V. Natarajan. "Output-sensitive construction of Reeb graphs", *IEEE Transactions on Visualization and Computer Graphics*, 18(1), 2012, 146-159

Acknowledgment

- Part of the materials in this lecture notes are from
  - Prof. Eugene Zhang
  - Prof. Vijay Natarajan
  - Dr. Julien Tierny