Eurographics 2015

The 36th Annual Conference of the European Association for Computer Graphics

Morse complexes for shape segmentation and homological analysis: discrete models and algorithms

Leila De Floriani[1] Ulderico Fugacci[1] Federico Iuricich[2] Paola Magillo[1]

[1]Department of Computer Science, Bioengineering, Robotics, and Systems Engineering, Univerity of Genova, Genova, Italy

[2]Department of Computer Science and UMIACS, University of Maryland, College Park (MD), USA

Computational topology and shape analysis

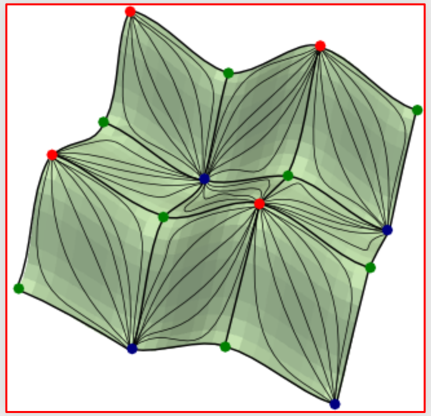
- Adapt methods of differential topology and of algebraic topology to various applied problems in scientific and engineering fields, e.g. molecular biology, sensor networks, scientific visualization, robotics
- Topology is the basis for structural shape descriptors (e.g, Reeb graphs, contour trees, Morse complexes, Betti numbers)
- Topological methods act as a geometric/combinatorial approach to shape understanding and recognition

Introduction

- Morse theory
  - topological tool for efficiently analyzing a shape by studying the behavior of a smooth scalar function *f* defined on it
- Morse complexes
  - topological shape descriptors through the critical points of function *f*
- Discrete Morse theory *[Forman, 1960]:*
  - discrete counterpart of Morse theory defined on cell complexes

Discrete shapes

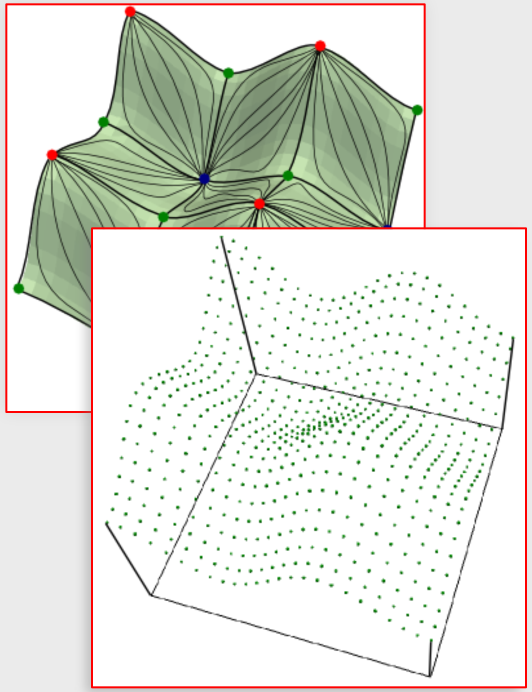
- Triangle meshes:
  - closed triangulated surfaces or irregularly sampled terrains
- Regular square grids:
  - regularly sampled terrains
- Tetrahedral meshes:
  - irregularly sampled volume data
- Regular cubic grids:
  - regularly sampled volume data



- A scalar value is associated with the vertices of the mesh or grid

Discrete shapes

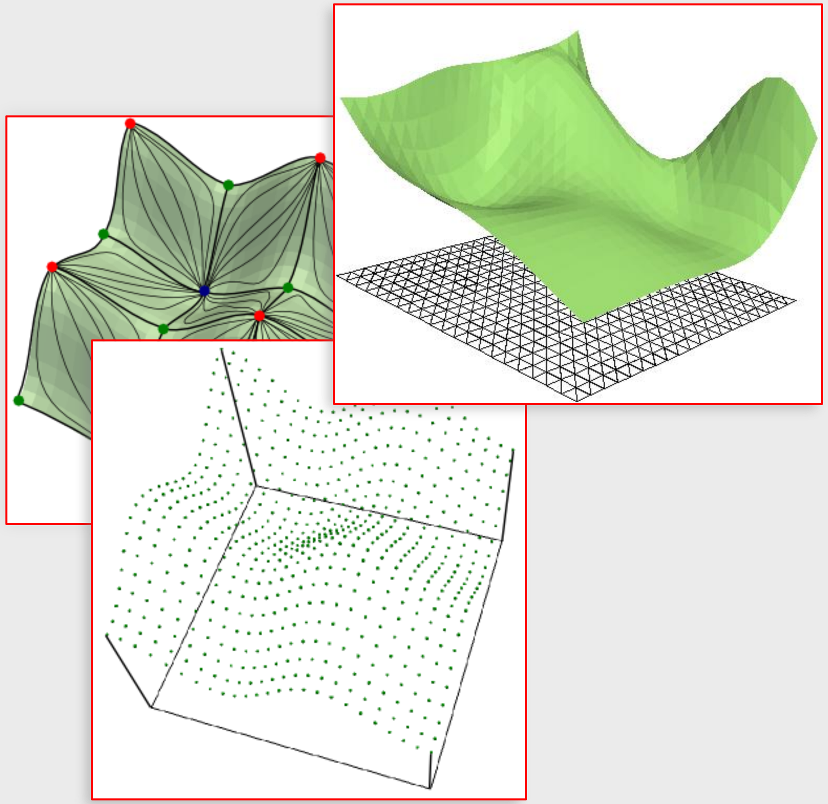
- Triangle meshes:
  - closed triangulated surfaces or irregularly sampled terrains
- Regular square grids:
  - regularly sampled terrains
- Tetrahedral meshes:
  - irregularly sampled volume data
- Regular cubic grids:
  - regularly sampled volume data



- A scalar value is associated with the vertices of the mesh or grid

Discrete shapes

- Triangle meshes:
  - closed triangulated surfaces or irregularly sampled terrains
- Regular square grids:
  - regularly sampled terrains
- Tetrahedral meshes:
  - irregularly sampled volume data
- Regular cubic grids:
  - regularly sampled volume data

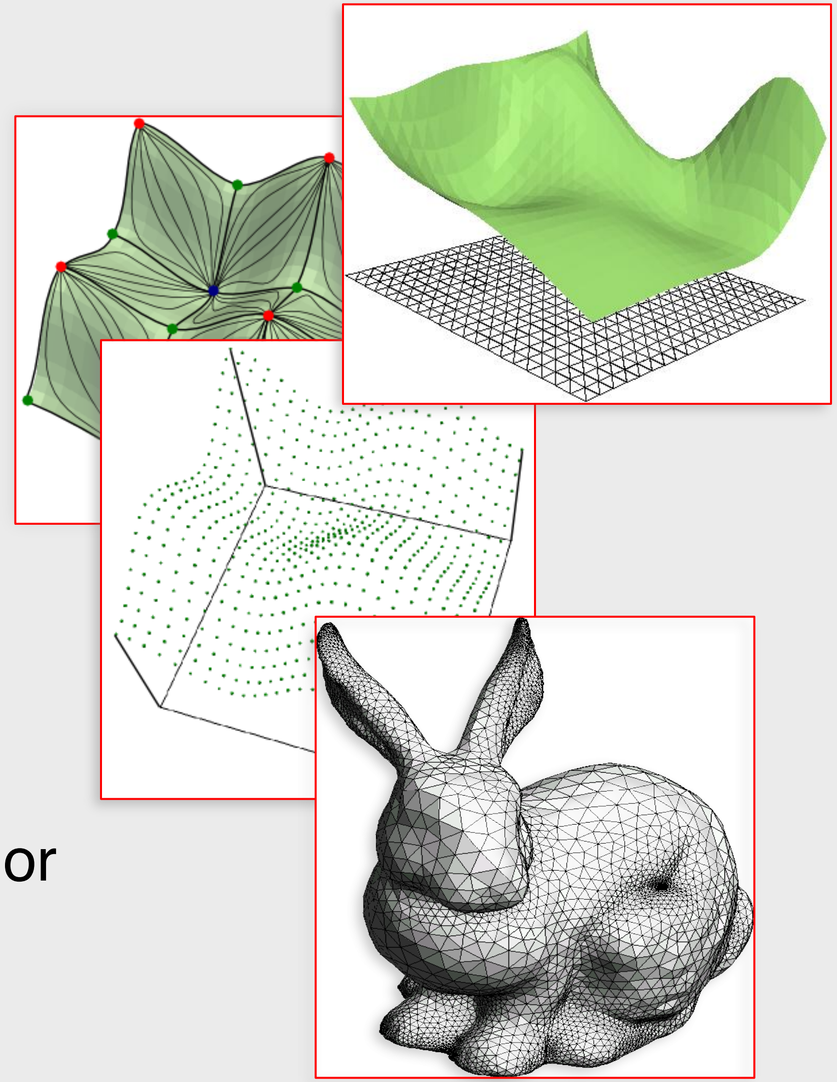


- A scalar value is associated with the vertices of the mesh or grid

Discrete shapes

- Triangle meshes:
  - closed triangulated surfaces or irregularly sampled terrains
- Regular square grids:
  - regularly sampled terrains
- Tetrahedral meshes:
  - irregularly sampled volume data
- Regular cubic grids:
  - regularly sampled volume data

- A scalar value is associated with the vertices of the mesh or grid



Applications: shape segmentation

- Segmenting the boundary of a 3D shape

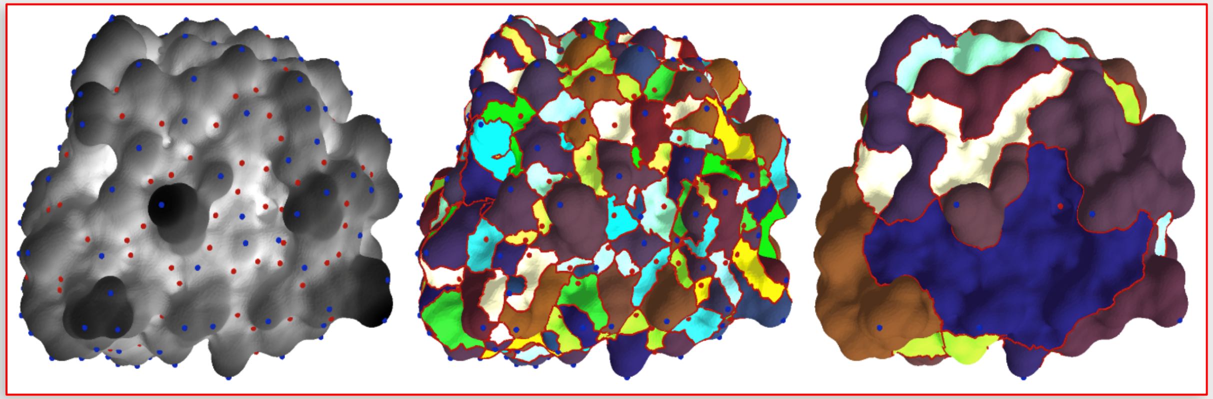


Image from [Natarajan V. et al., 2006]
Study of cavities and protrusions in an atomic density function defined on a triangulated surface

Applications: shape segmentation

- Segmenting the boundary of a 3D shape

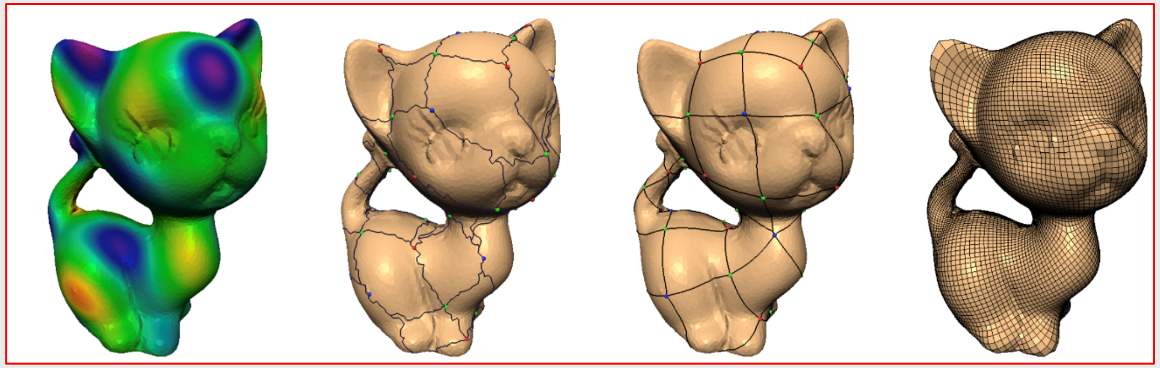


Image from [Dong S. et al., 2006] quad mesh generation from a triangle mesh by considering the eigenfunctions of the discrete Laplacian operator

Applications: shape segmentation

- Segmenting the boundary of a 3D shape
- Volume data segmentation

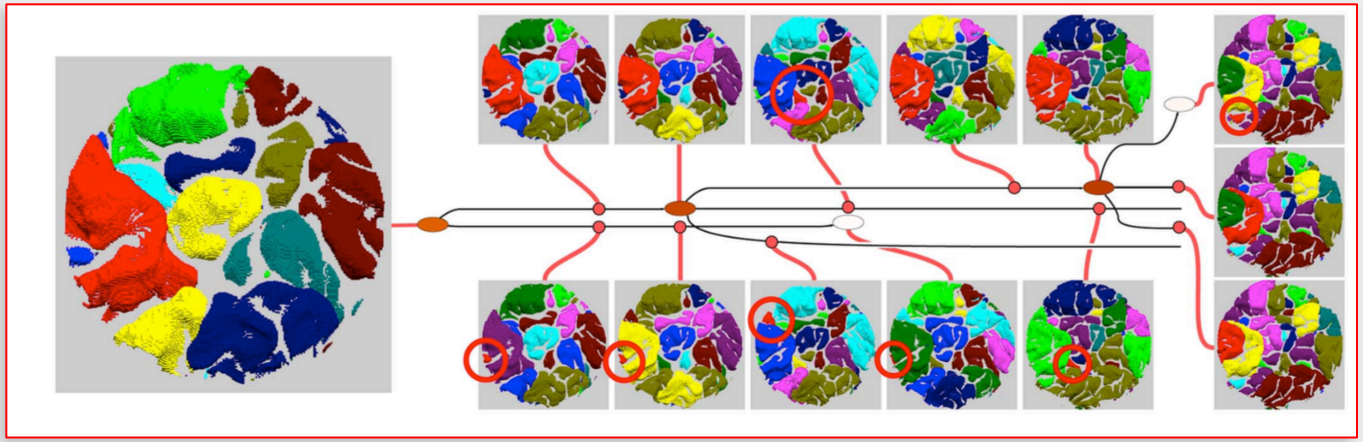


Image from [Bremer P-T. et al., 2010] burning cells tracked over time – Morse complexes at different time steps

Applications: shape segmentation

- Segmenting the boundary of a 3D shape
- Volume data segmentation
- Multi-resolution terrain analysis

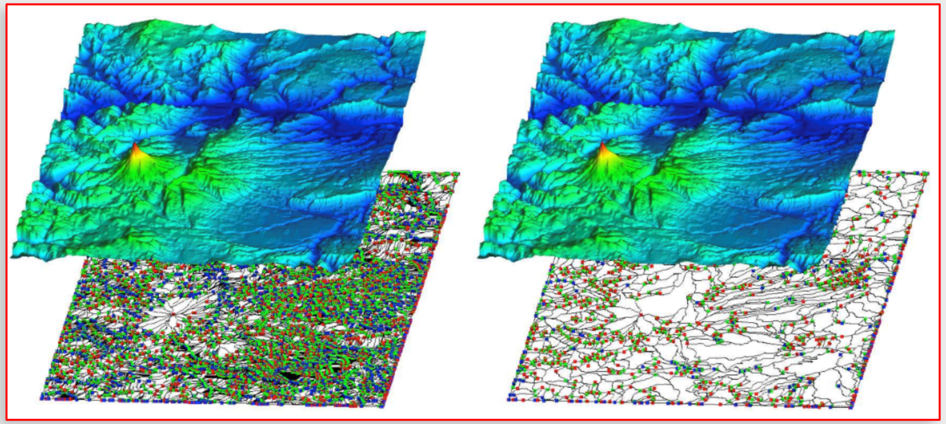


Image from [Bremer et al., 2004] network of the critical points at two levels of resolution: 10% of the critical points in the picture on the right

Applications: shape segmentation

- Segmenting the boundary of a 3D shape
- Volume data segmentation
- Multi-resolution terrain analysis
- Multi-resolution analysis of volume data

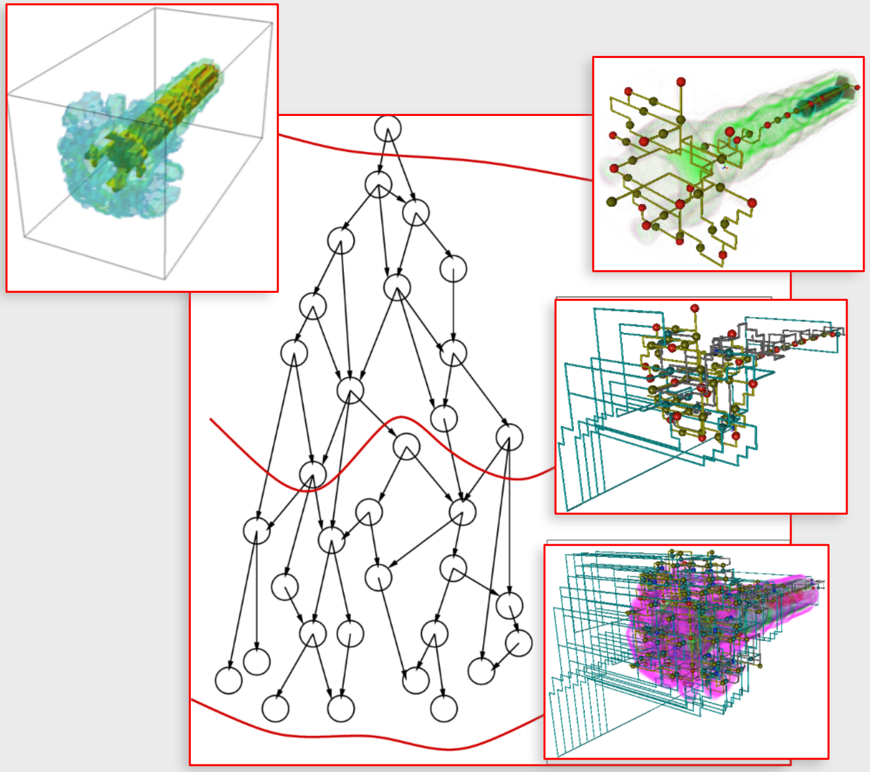


Image from [Gyulassy et al., 2010] network of the critical points on a volume data set at different resolutions

Applications: homology computation

- Homology computation
  - detection of holes in shapes
- 3D and higher-dimensional shapes
- Shapes discretized as simplicial complexes (generalization of triangle and tetrahedral meshes)

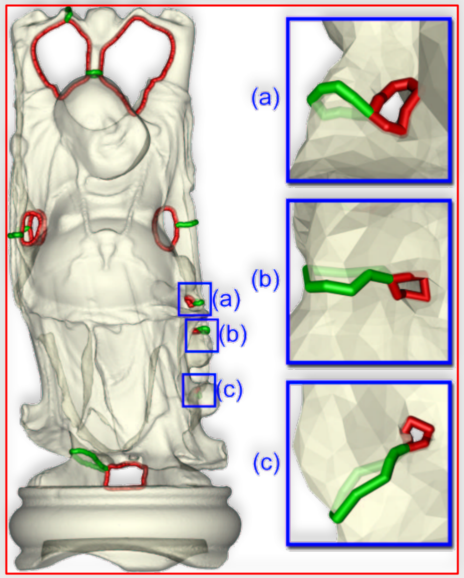


Image from
[Dey. et al., 2008]

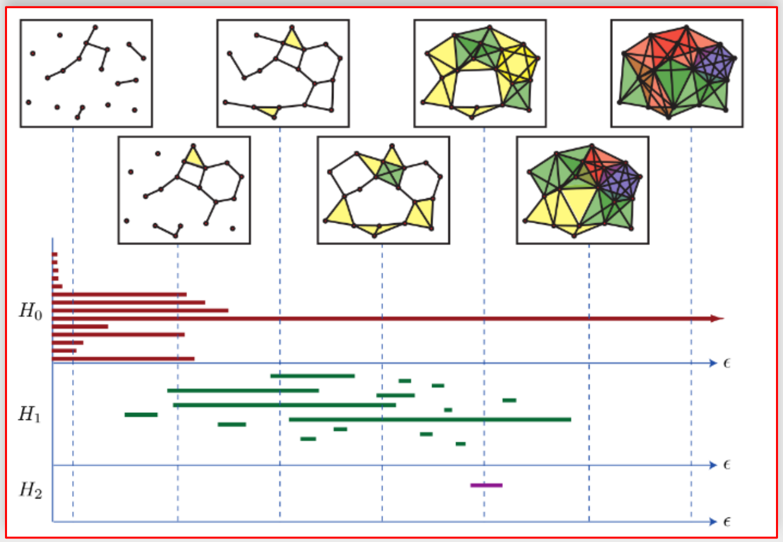
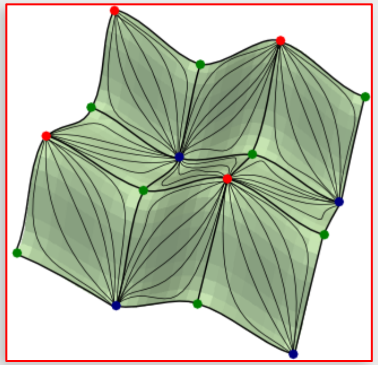
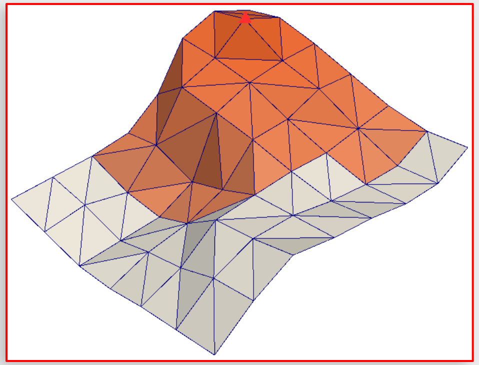


Image from [Ghrist, 2008]

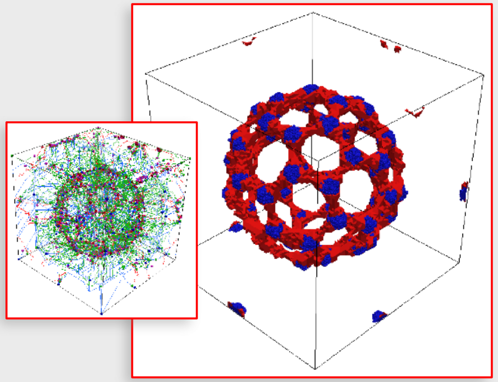
Outline

Morse theory in the smooth case → Morse theory in the discrete case

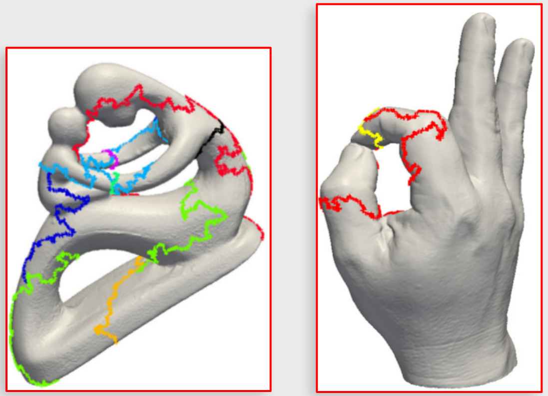




Morse theory for shape segmentation



Morse theory for homology computation



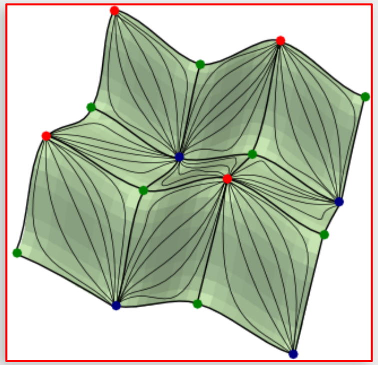
Outline

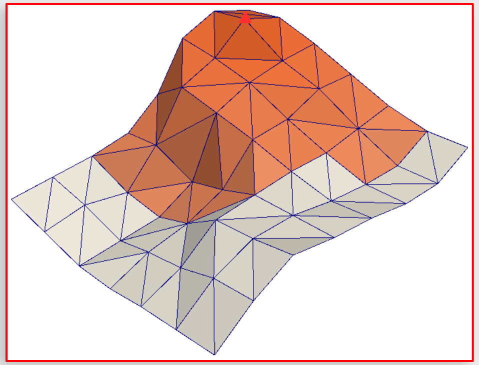
Morse theory in the smooth case

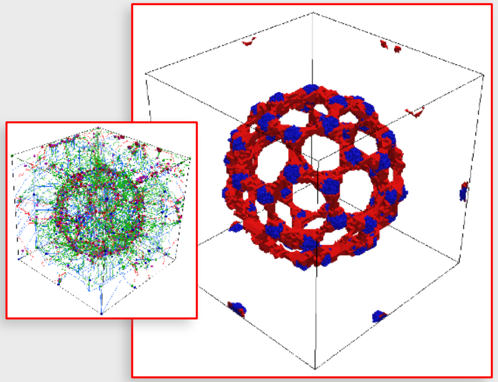
Morse theory in the discrete case

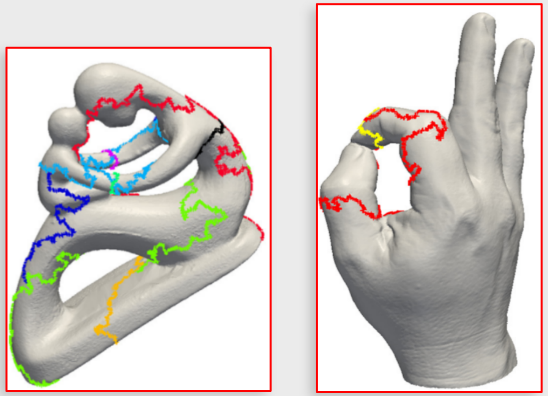
Morse theory for shape segmentation

Morse theory for homology computation





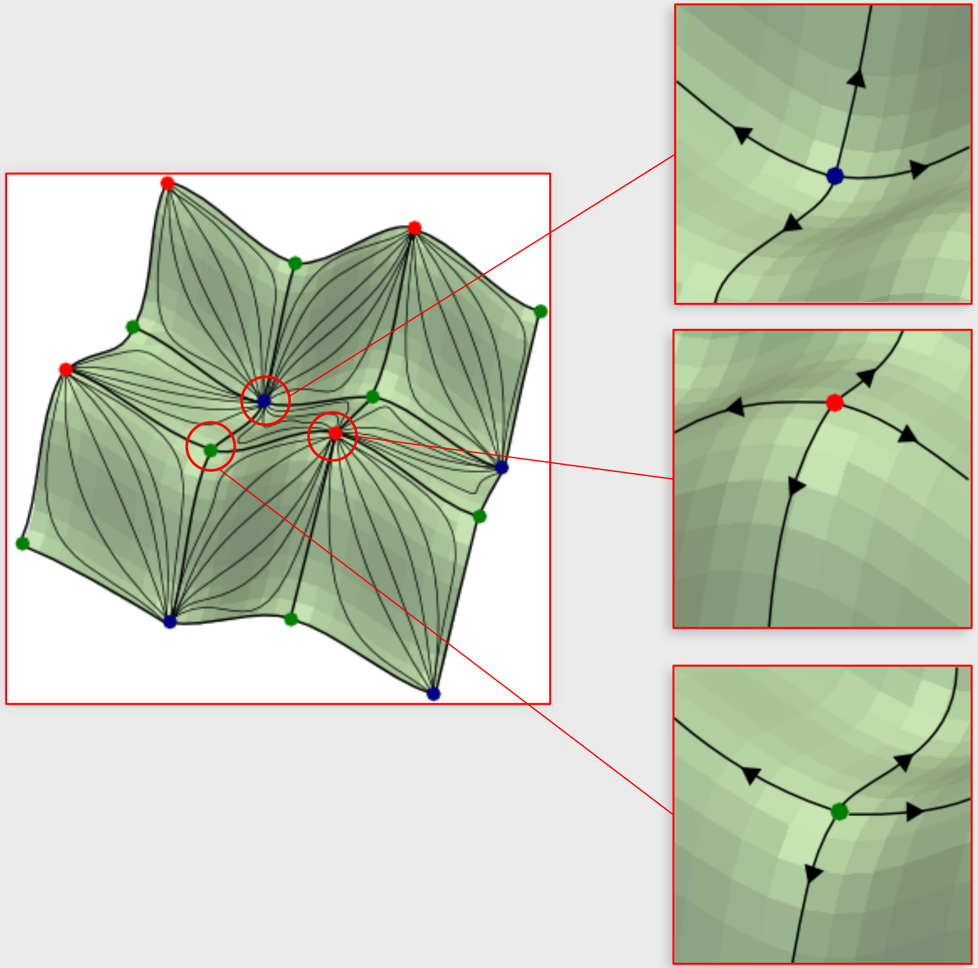




Morse Theory [Milnor J., 1963; Matsumoto Y., 2002]

- Relates the critical points of a smooth scalar function defined on a manifold shape to the topology of the shape
  - Manifold *M*: the neghborhood of each point of *M* is homeomorphic to the open unit ball in Euclidean space

- Analysis of a manifold shape endowed with a scalar function requires extracting morphological features (e.g., critical points, integral lines and surfaces)



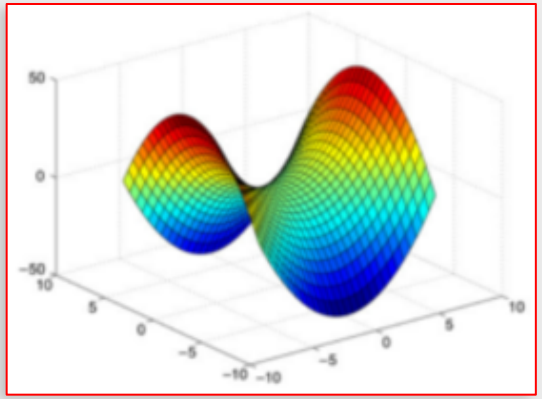
Morse Theory

Let f be a real-valued C^2-function defined on a d-dimensional manifold M

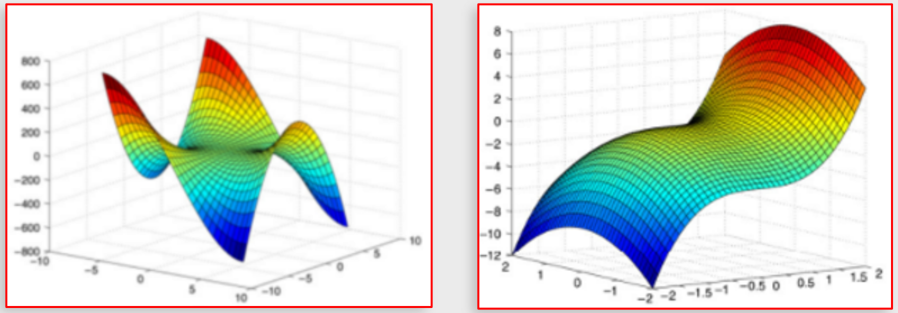
•Critical point of f: any point on M in which the gradient of f vanishes

•A critical point p is degenerate if and only if the determinant the Hessian matrix H of the second order derivatives of function f at p is null

•Function f is a Morse function if and only if all its critical points are non-degenerate



Non-degenerate critical point



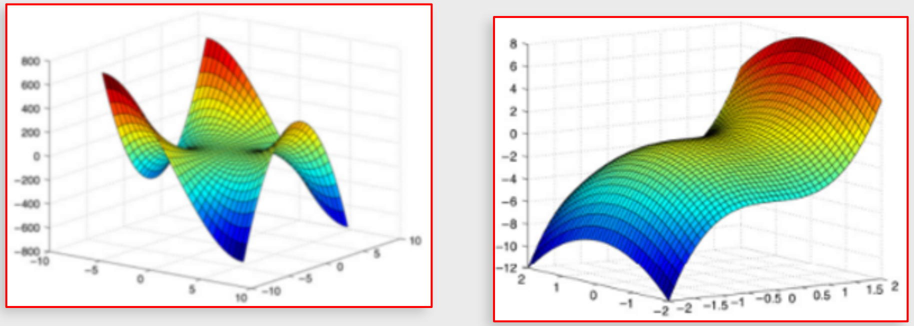
Degenerate critical points (monkey saddle and flat saddle)

Morse Theory

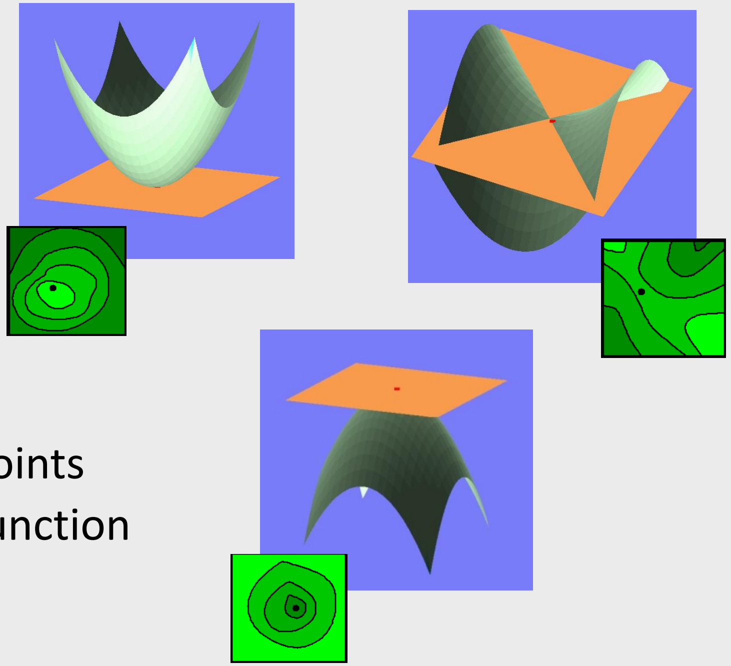
- The critical points of a Morse function defined on a compact manifold are isolated

- A *d*-dimensional Morse function *f* has *d+1* types of critical points
  - For d=2 : minima, saddles and maxima
  - For d=3: minima, 1-saddles, 2- saddles and maxima

- The index *i* of a non-degenerate critical point *p* is the number of negative eigenvalues of the Hessian of *f* at *p*

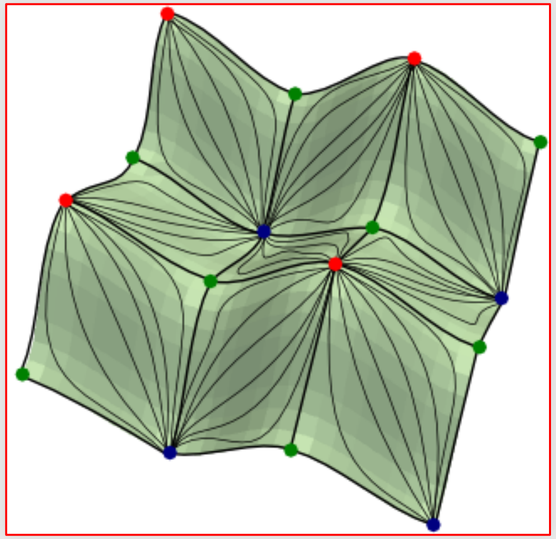


Examples of non-Morse functions



Critical points of a 2D function

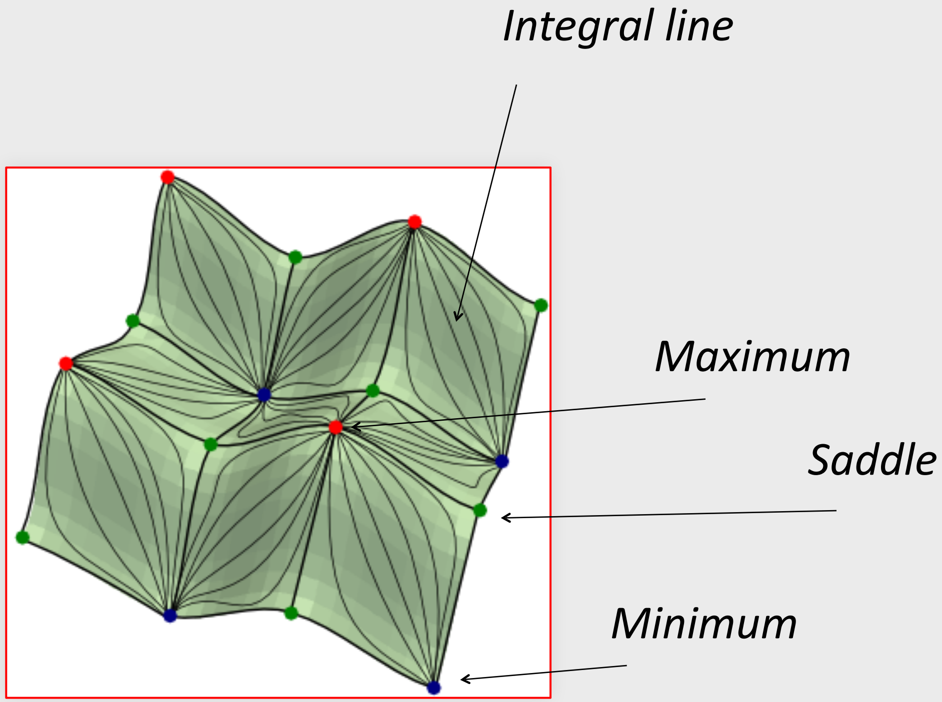
Morse Theory



Morse Theory

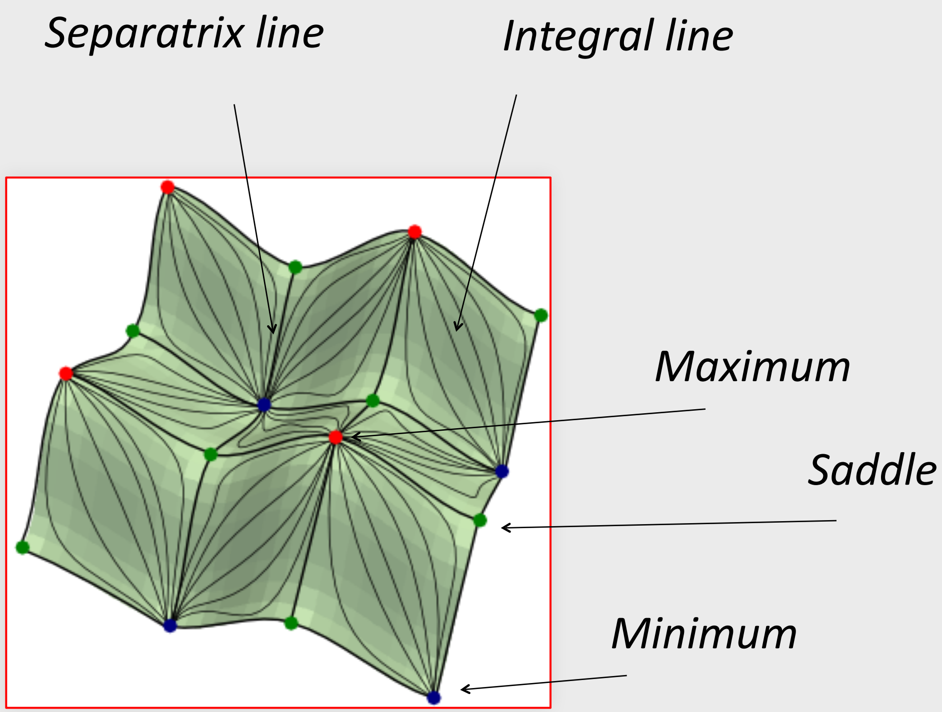
- An integral line of a smooth function f is a maximal path on M whose tangent vectors agree everywhere with the gradient of f

- Integral lines start and end at the critical points of f



Morse Theory

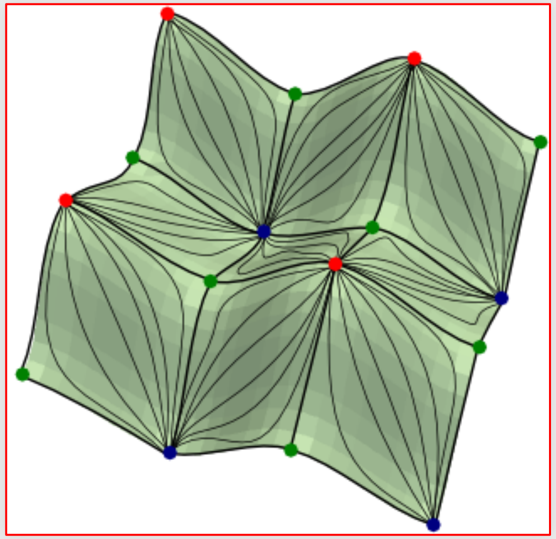
- An integral line of a smooth function f is a maximal path on M whose tangent vectors agree everywhere with the gradient of f
- Integral lines start and end at the critical points of f
- Integral lines that connect critical points of consecutive index are called separatrix lines



Descending Morse complexes

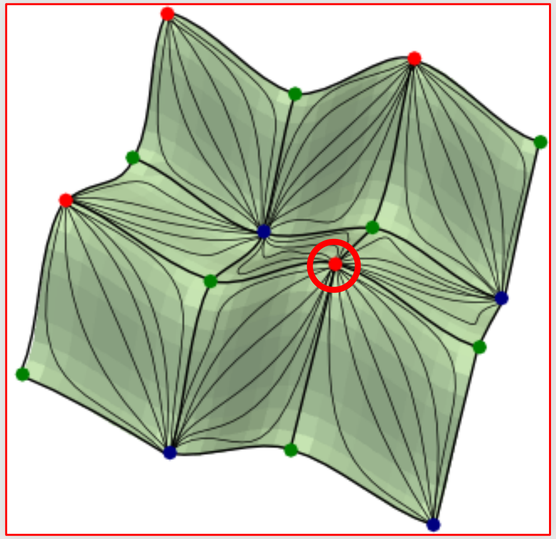
- Integral lines that converge toward a critical point *p* of index *i* form an i-cell called the descending (stable) cell of *p*
  - Descending cell of a maximum: 2-cell
  - Descending cell of a saddle: 1-cell
  - Descending cell of a minimum: 0-cell

- Descending Morse complex: collection of the descending cells of all critical points of function *f*



Descending Morse complexes

- Integral lines that converge toward a critical point *p* of index *i* form an i-cell called the descending (stable) cell of *p*
  - Descending cell of a maximum: 2-cell
  - Descending cell of a saddle: 1-cell
  - Descending cell of a minimum: 0-cell

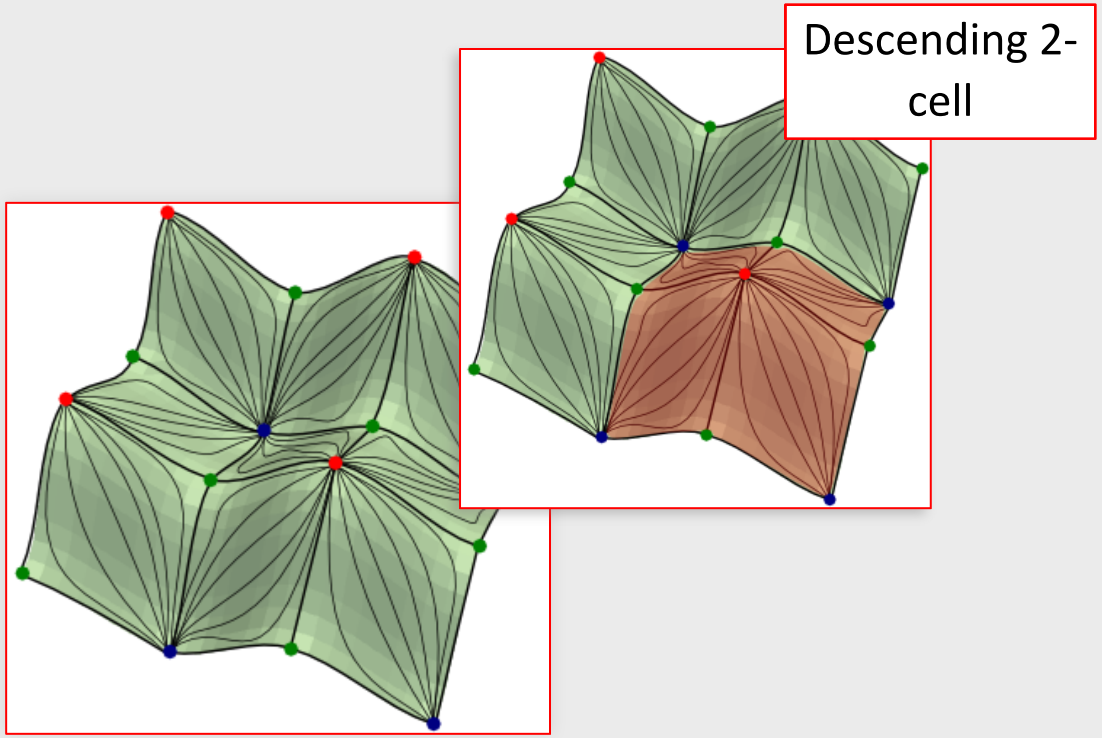


- Descending Morse complex: collection of the descending cells of all critical points of function *f*

Descending Morse complexes

- Integral lines that converge toward a critical point *p* of index *i* form an i-cell called the descending (stable) cell of *p*
  - Descending cell of a maximum: 2-cell
  - Descending cell of a saddle: 1-cell
  - Descending cell of a minimum: 0-cell

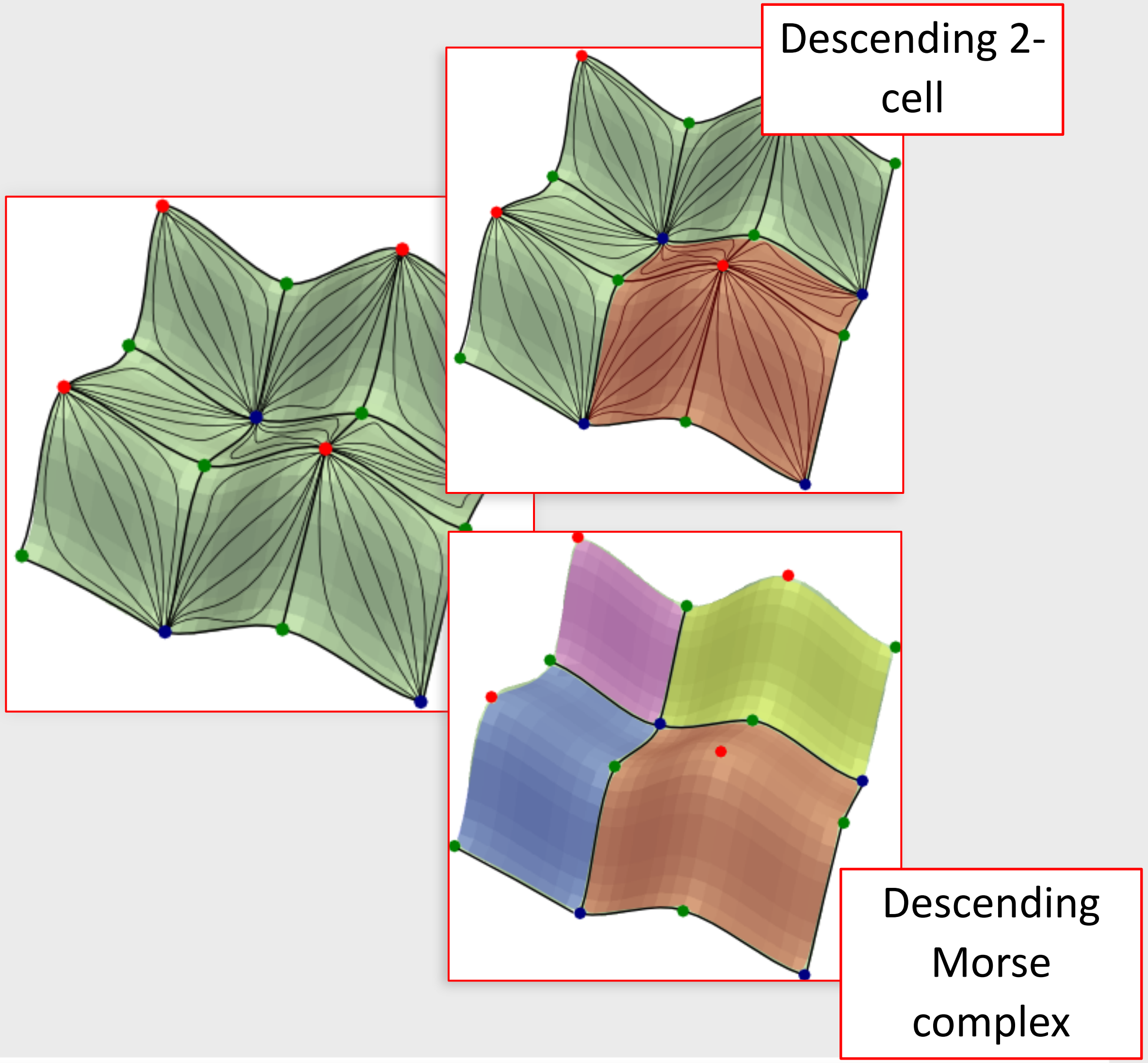
- Descending Morse complex: collection of the descending cells of all critical points of function *f*



Descending Morse complexes

- Integral lines that converge toward a critical point *p* of index *i* form an i-cell called the descending (stable) cell of *p*
  - Descending cell of a maximum: 2-cell
  - Descending cell of a saddle: 1-cell
  - Descending cell of a minimum: 0-cell

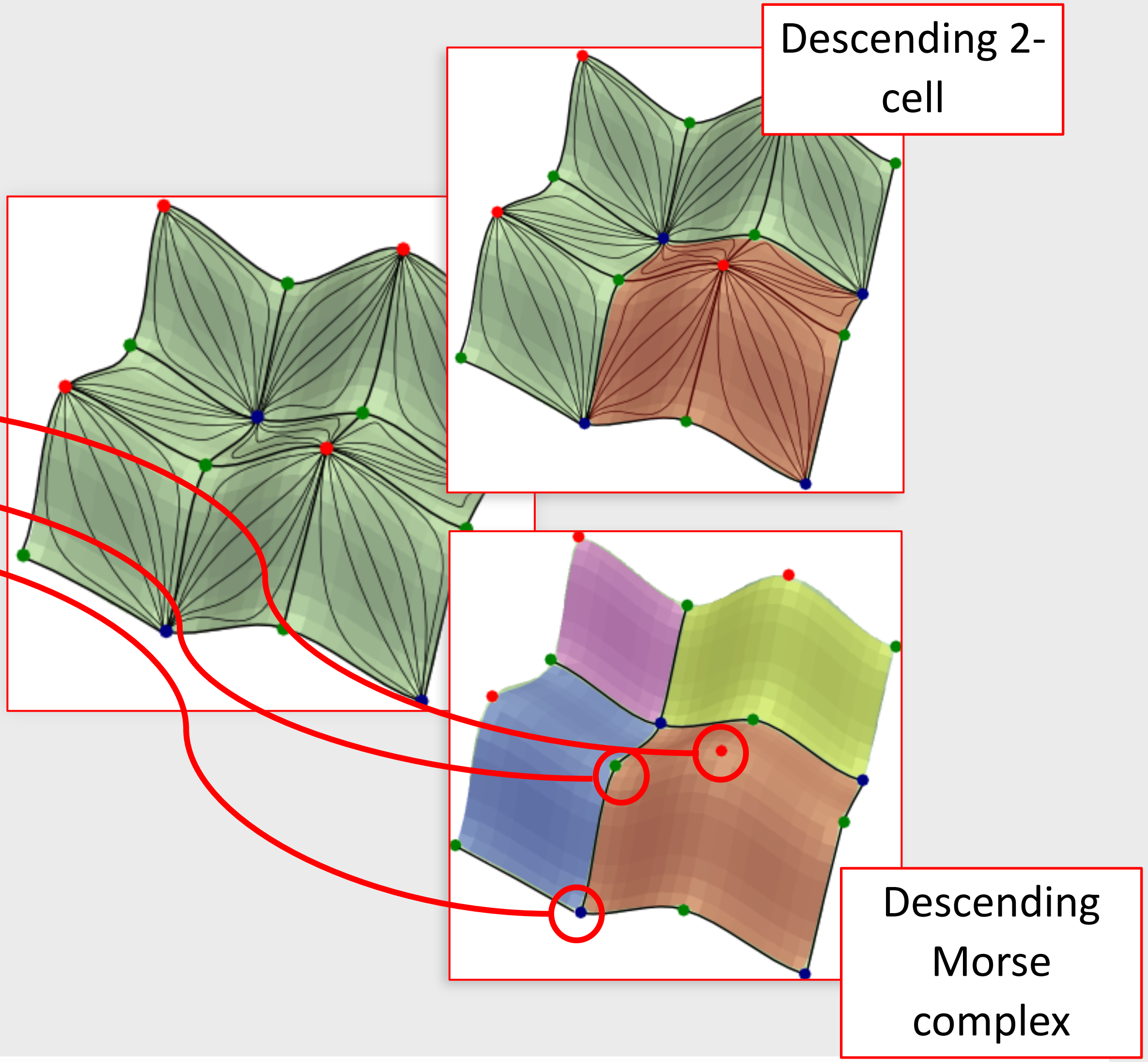
- Descending Morse complex: collection of the descending cells of all critical points of function *f*



Descending Morse complexes

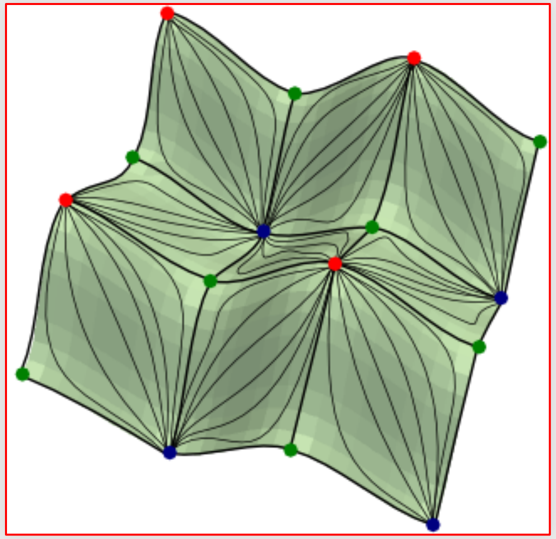
- Integral lines that converge toward a critical point *p* of index *i* form an i-cell called the descending (stable) cell of *p*
  - Descending cell of a maximum: 2-cell
  - Descending cell of a saddle: 1-cell
  - Descending cell of a minimum: 0-cell

- Descending Morse complex: collection of the descending cells of all critical points of function *f*



Ascending Morse complexes

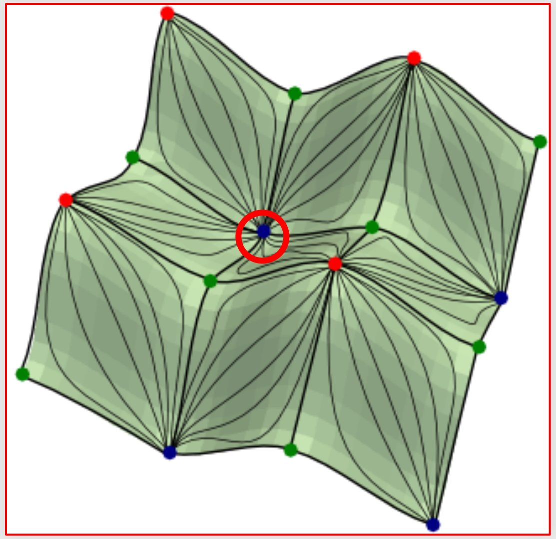
- Integral lines that originate at a critical point *p* of index *i* form a *(d-i)*-cell called the ascending (unstable) cell of p
  - Ascending cell of a maximum: 0-cell
  - Ascending cell of saddle: 1-cell
  - Ascending cell of minimum: 2-cell



- Ascending Morse complex: collection of the ascending cells of all critical points of function *f*

Ascending Morse complexes

- Integral lines that originate at a critical point *p* of index *i* form a *(d-i)*-cell called the ascending (unstable) cell of p
  - Ascending cell of a maximum: 0-cell
  - Ascending cell of saddle: 1-cell
  - Ascending cell of minimum: 2-cell

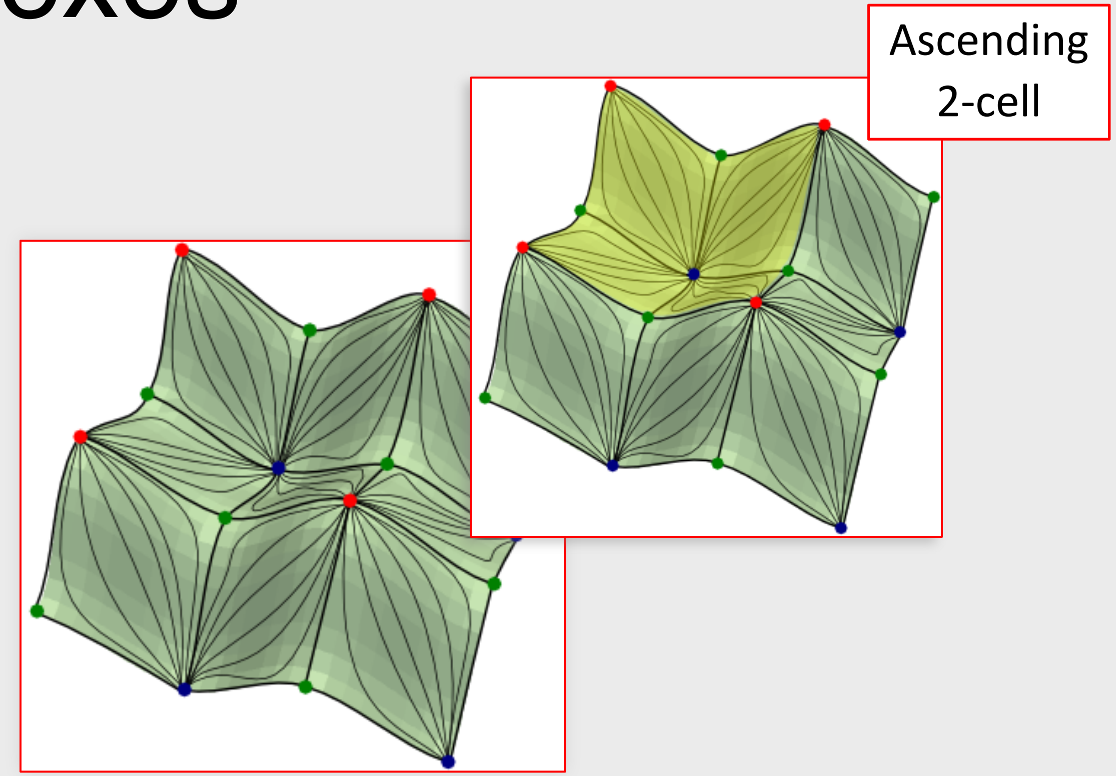


- Ascending Morse complex: collection of the ascending cells of all critical points of function *f*

Ascending Morse complexes

- Integral lines that originate at a critical point *p* of index *i* form a *(d-i)*-cell called the ascending (unstable) cell of p
  - Ascending cell of a maximum: 0-cell
  - Ascending cell of saddle: 1-cell
  - Ascending cell of minimum: 2-cell

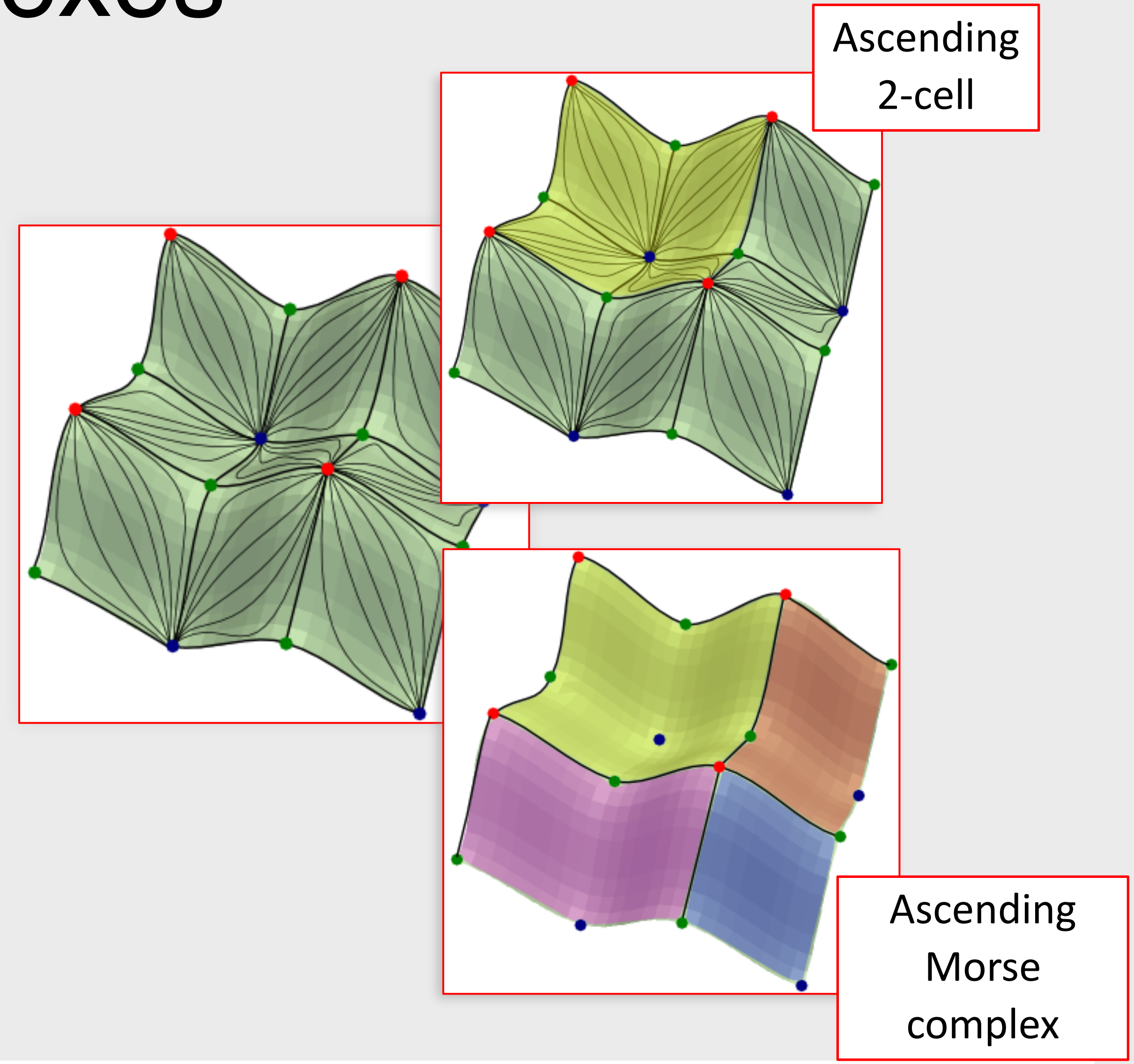
- Ascending Morse complex: collection of the ascending cells of all critical points of function *f*



Ascending Morse complexes

- Integral lines that originate at a critical point *p* of index *i* form a *(d-i)*-cell called the ascending (unstable) cell of p
  - Ascending cell of a maximum: 0-cell
  - Ascending cell of saddle: 1-cell
  - Ascending cell of minimum: 2-cell

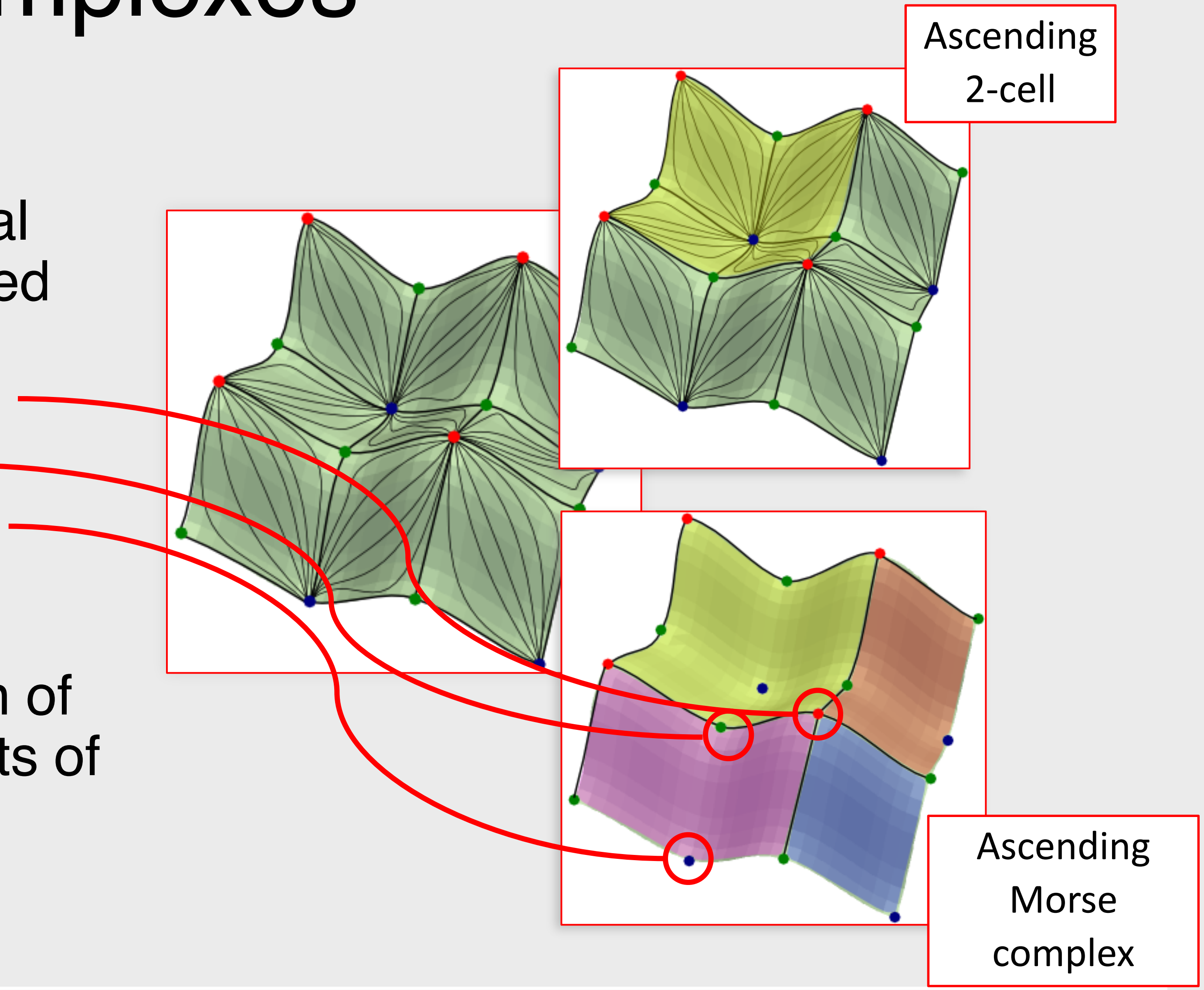
- Ascending Morse complex: collection of the ascending cells of all critical points of function *f*



Ascending Morse complexes

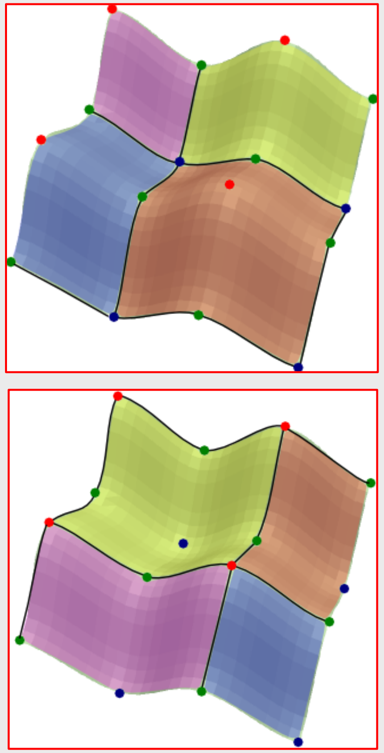
- Integral lines that originate at a critical point *p* of index *i* form a *(d-i)*-cell called the ascending (unstable) cell of p
  - Ascending cell of a maximum: 0-cell
  - Ascending cell of saddle: 1-cell
  - Ascending cell of minimum: 2-cell

- Ascending Morse complex: collection of the ascending cells of all critical points of function *f*



Morse-Smale complexes

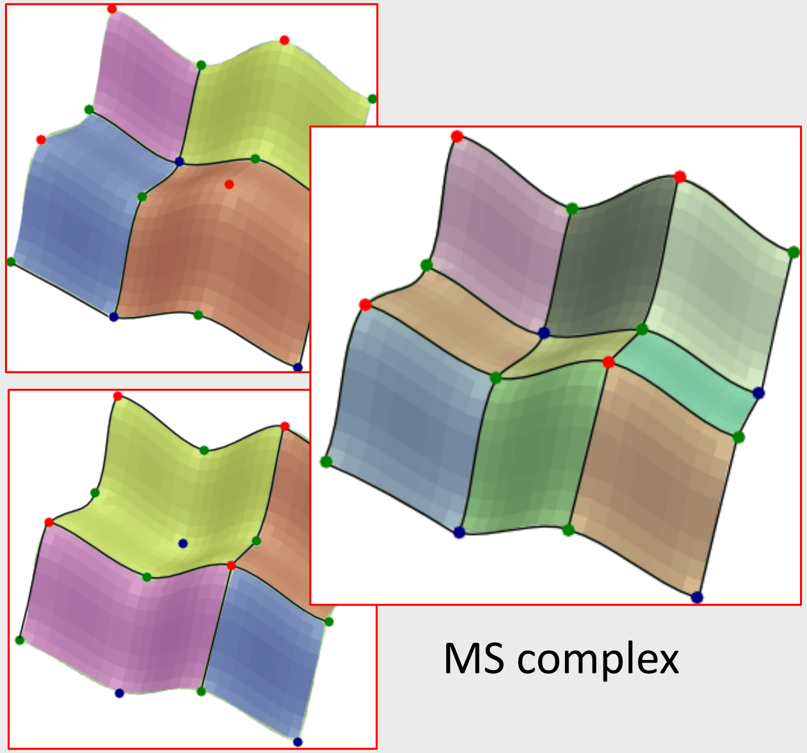
- Function *f* is a Morse-Smale function if its ascending and descending Morse cells intersect transversally
- Morse-Smale (MS) complex is the complex obtained by intersecting all the ascending and descending cells



Morse-Smale complexes

- Function *f* is a Morse-Smale function if its ascending and descending Morse cells intersect transversally

- Morse-Smale (MS) complex is the complex obtained by intersecting all the ascending and descending cells

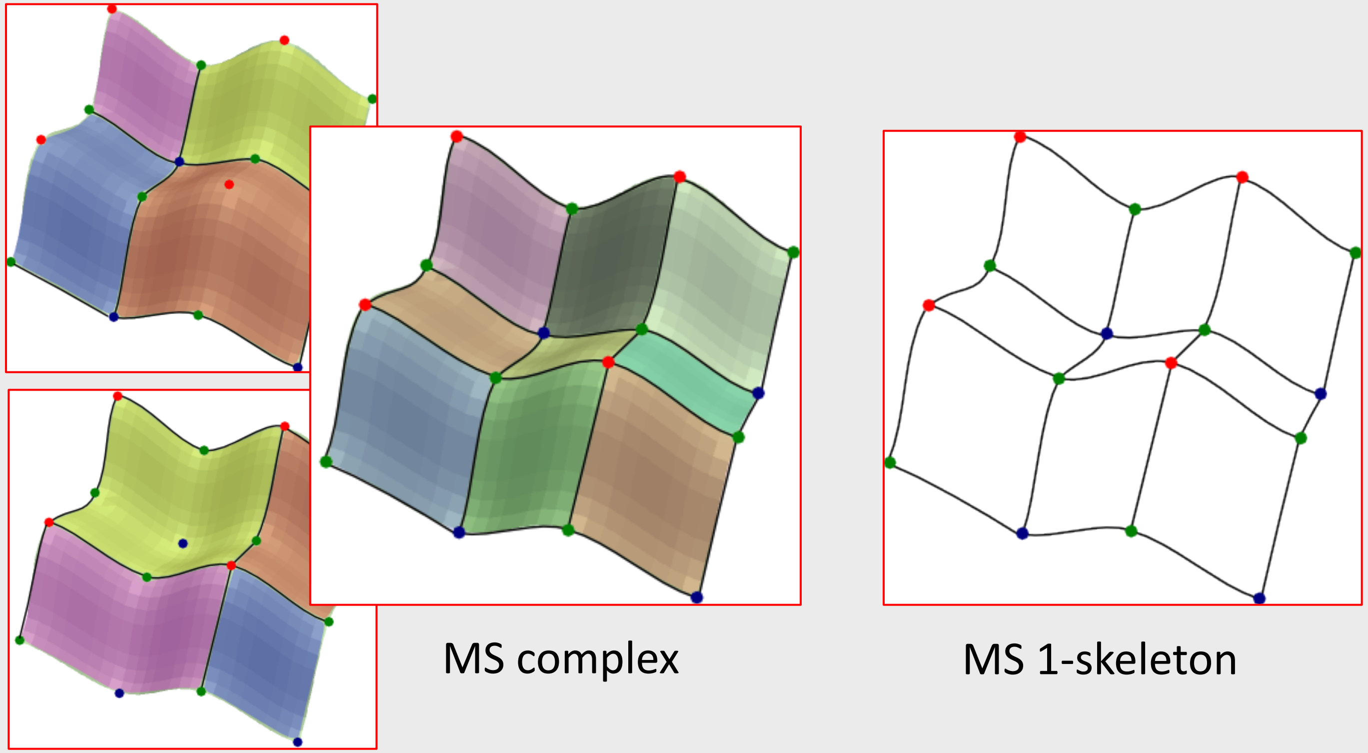


MS complex

Morse-Smale complexes

- Function *f* is a Morse-Smale function if its ascending and descending Morse cells intersect transversally

- Morse-Smale (MS) complex is the complex obtained by intersecting all the ascending and descending cells



MS complex

MS 1-skeleton

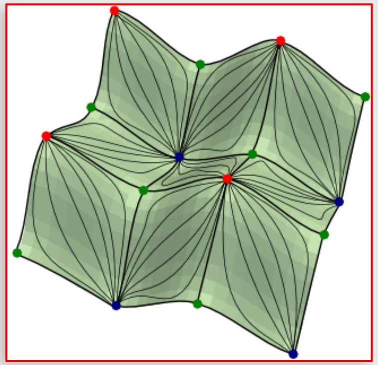
Outline

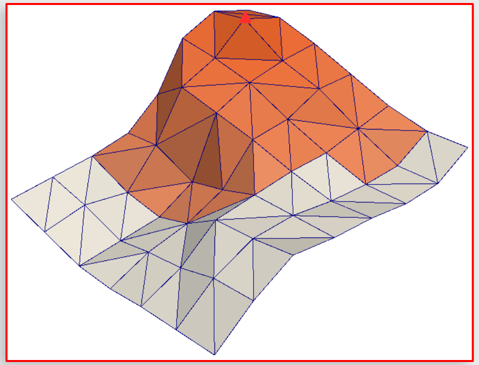
Morse theory in the **smooth case**

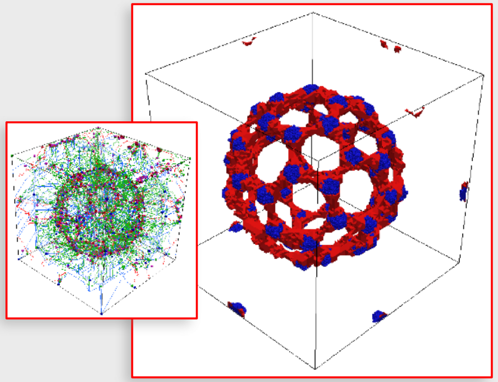
Morse theory in the **discrete case**

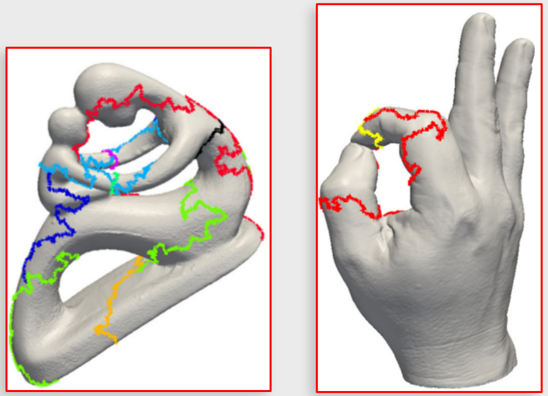
Morse theory for **shape segmentation**

Morse theory for **homology computation**

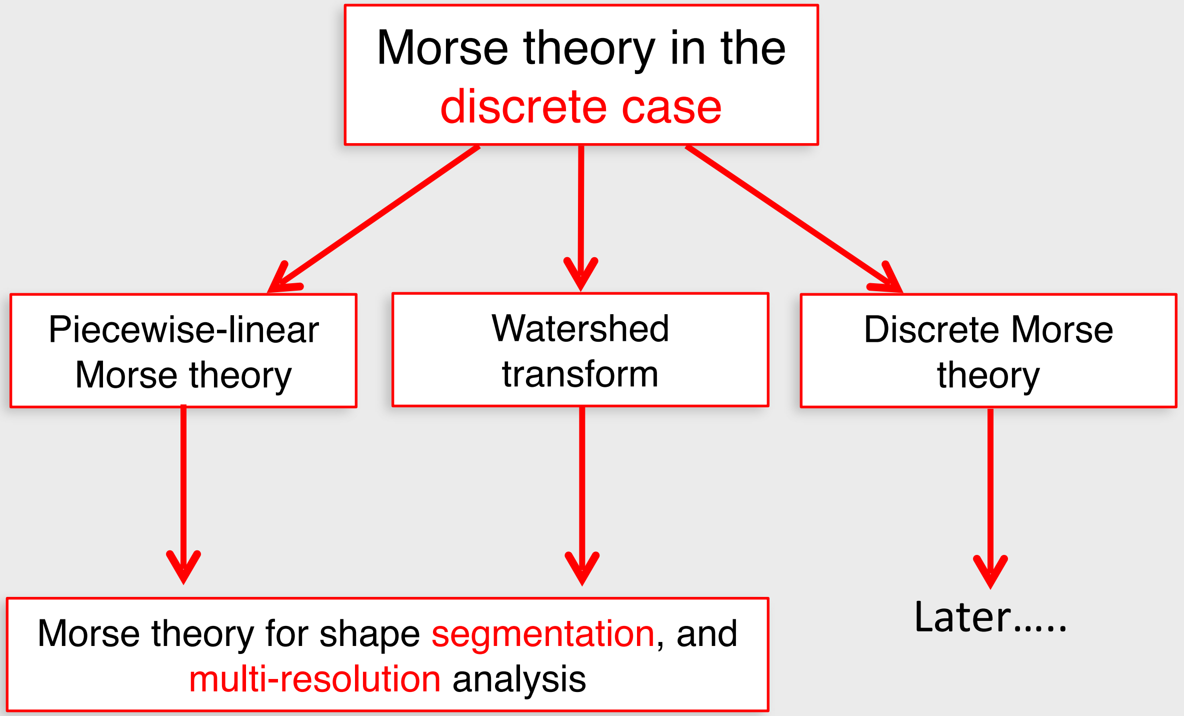






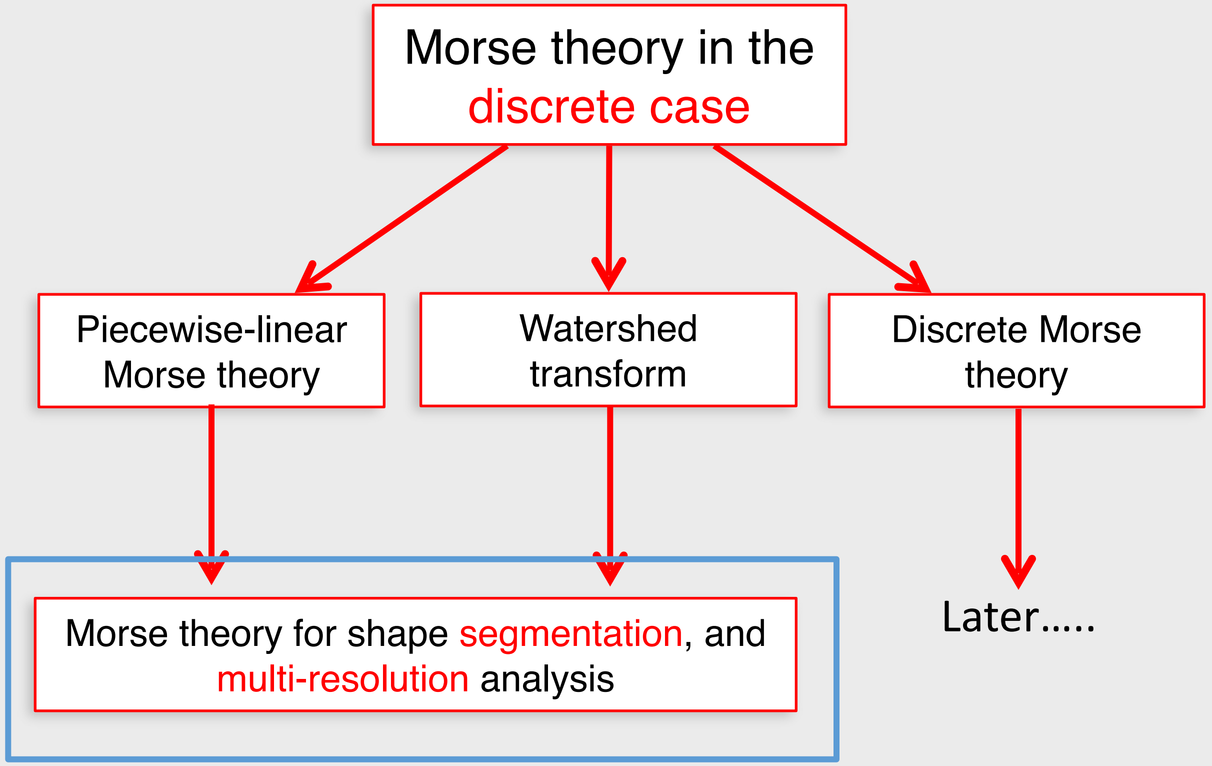


Morse theory in the discrete case



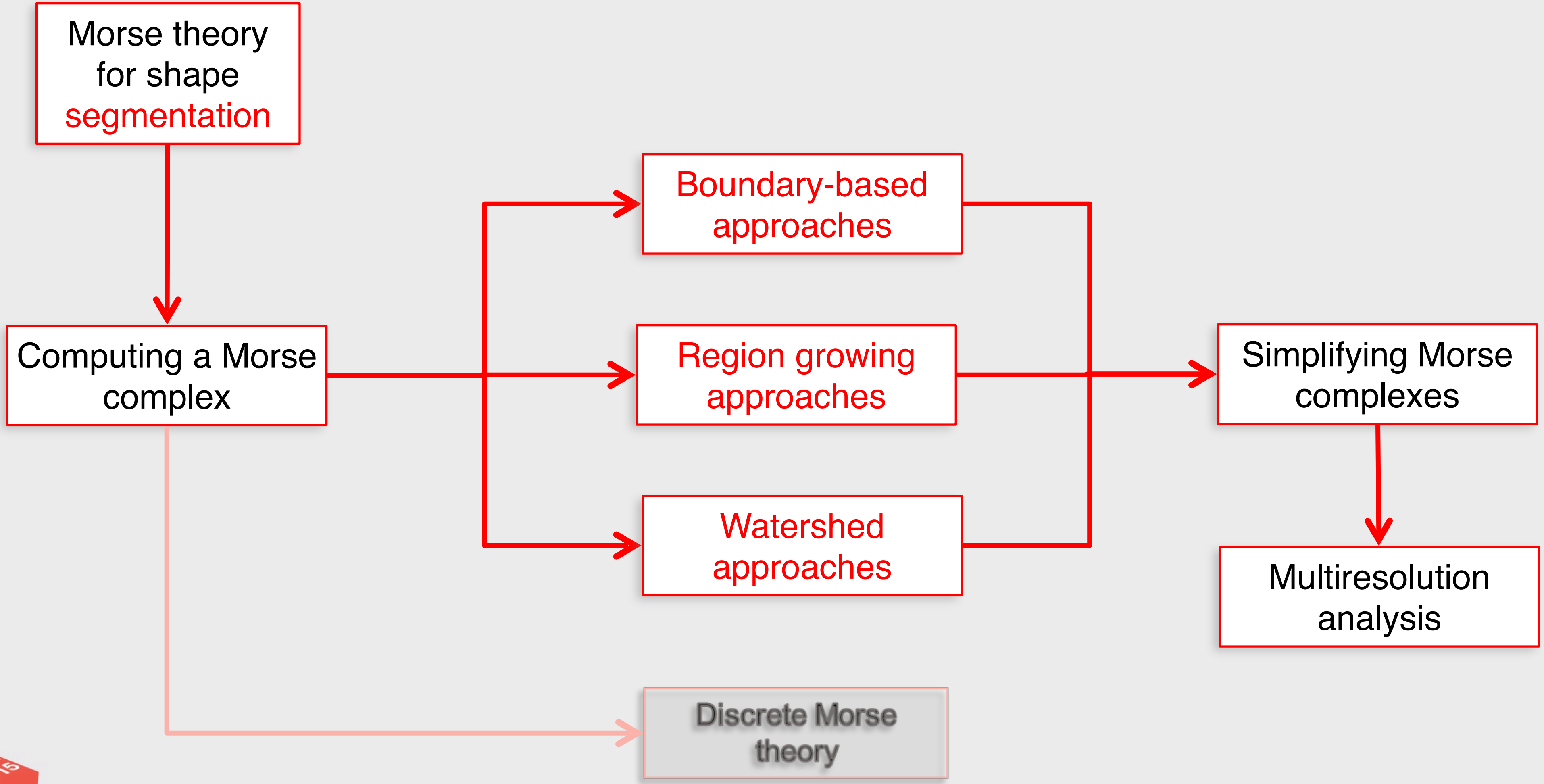
- Piecewise-linear Morse theory *[Banchoff 1967, 1970; Edelsbrunner et al.,2001, 2003]*
  - Characterization of the critical points for polyhedral surfaces in 2D and 3D
- Watershed transform *[F. Meyer 1994]*
  - For images and labeled graphs
  - Dimension-independent
- Discrete Morse theory *[R. Forman 1998, 2002]*
  - For cell complexes
  - Dimension-independent

Morse theory in the discrete case

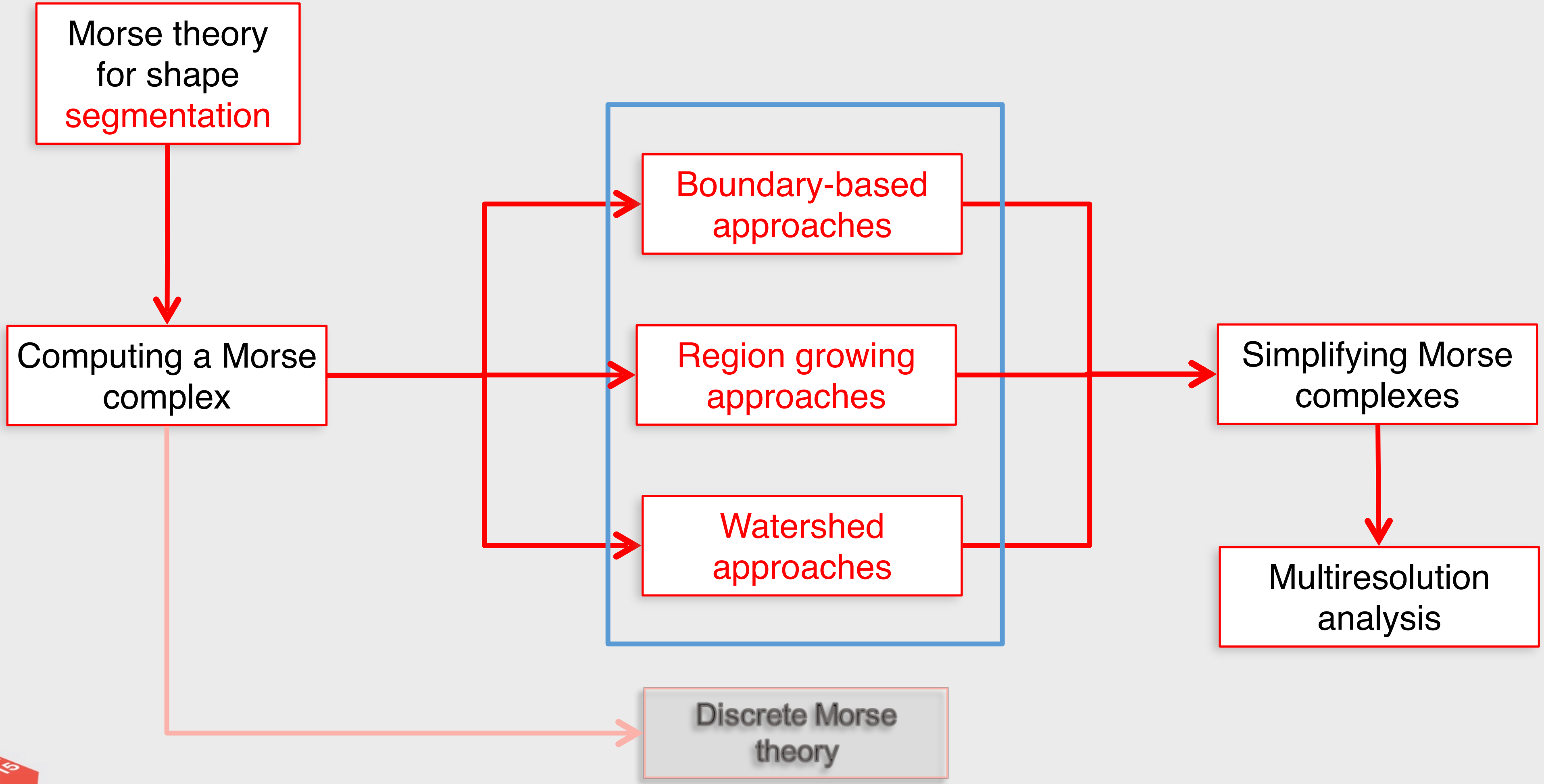


- Piecewise-linear Morse theory *[Banchoff 1967, 1970; Edelsbrunner et al.,2001, 2003]*
  - Characterization of the critical points for polyhedral surfaces in 2D and 3D
- Watershed transform *[F. Meyer 1994]*
  - For images and labeled graphs
  - Dimension-independent
- Discrete Morse theory *[R. Forman 1998, 2002]*
  - For cell complexes
  - Dimension-independent

Morse theory for shape segmentation

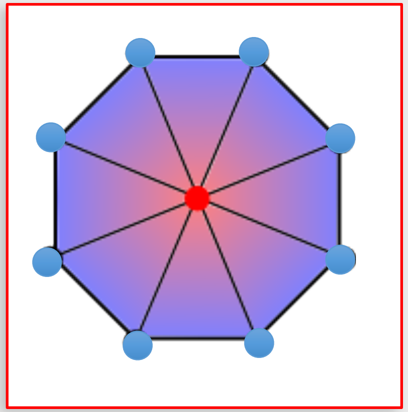


Morse theory for shape segmentation

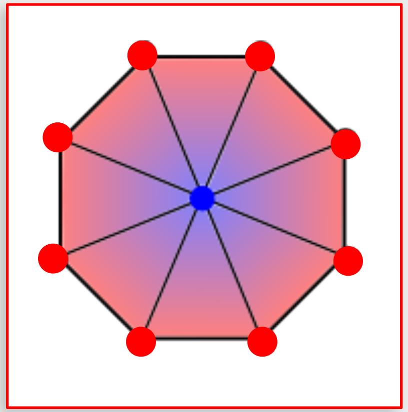


Characterization of critical points

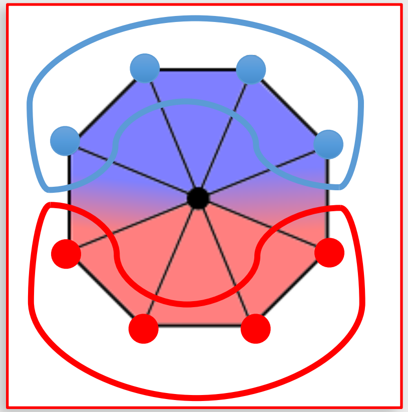
- Consider a triangulated surface endowed with a function *f* defined at its vertices
- Assumption: any pair of adjacent vertices have different function values
- A critical point *p* is defined as regular, maximum, minimum or saddle depending of the values of *f* at its vertices



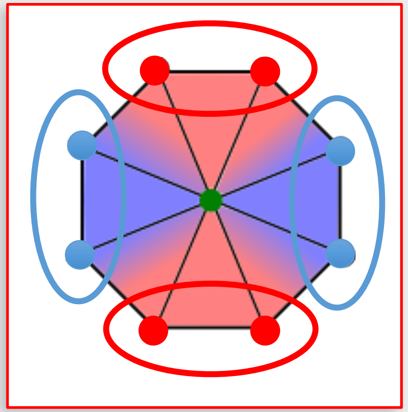
Maximum



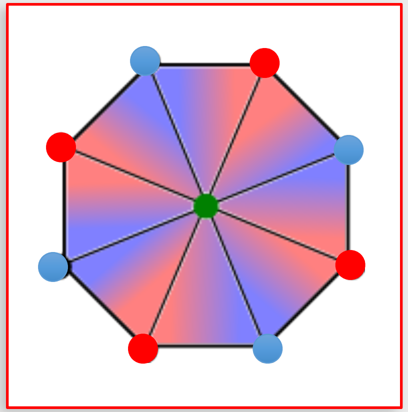
Minimum



Regular



Saddle



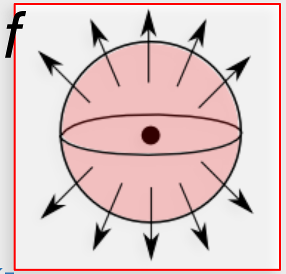
Multiple Saddle

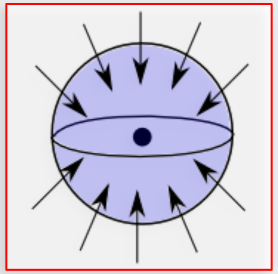
Characterization of critical points in 3D [Edelsbrunner et al., 2003]

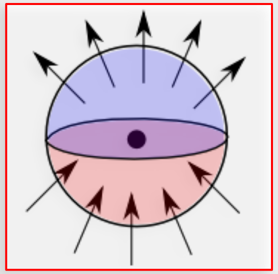
- In 3D: tetrahedral meshes endowed with a function *f* at its vertices
- A vertex *p* is classified based on:
  - number **m** of connected components in the lower link $Lk^-(p)$ of *p*
  - number **n** of connected components in the upper link $Lk^+(p)$ of *p*

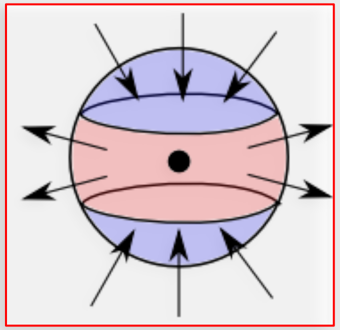
where

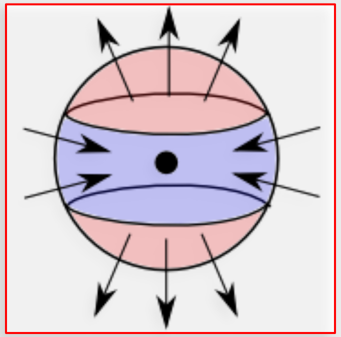
- lower link $Lk^-(p)$ of *p*: vertices *z* adjacent to *p* such that $f(z)<f(p)$ plus the edges of the mesh joining them
- upper link $Lk^+(p)$ of *p*: vertices *q* adjacent to *p* such that $f(q)>f(p)$ plus the edges of the mesh joining them


Minimum

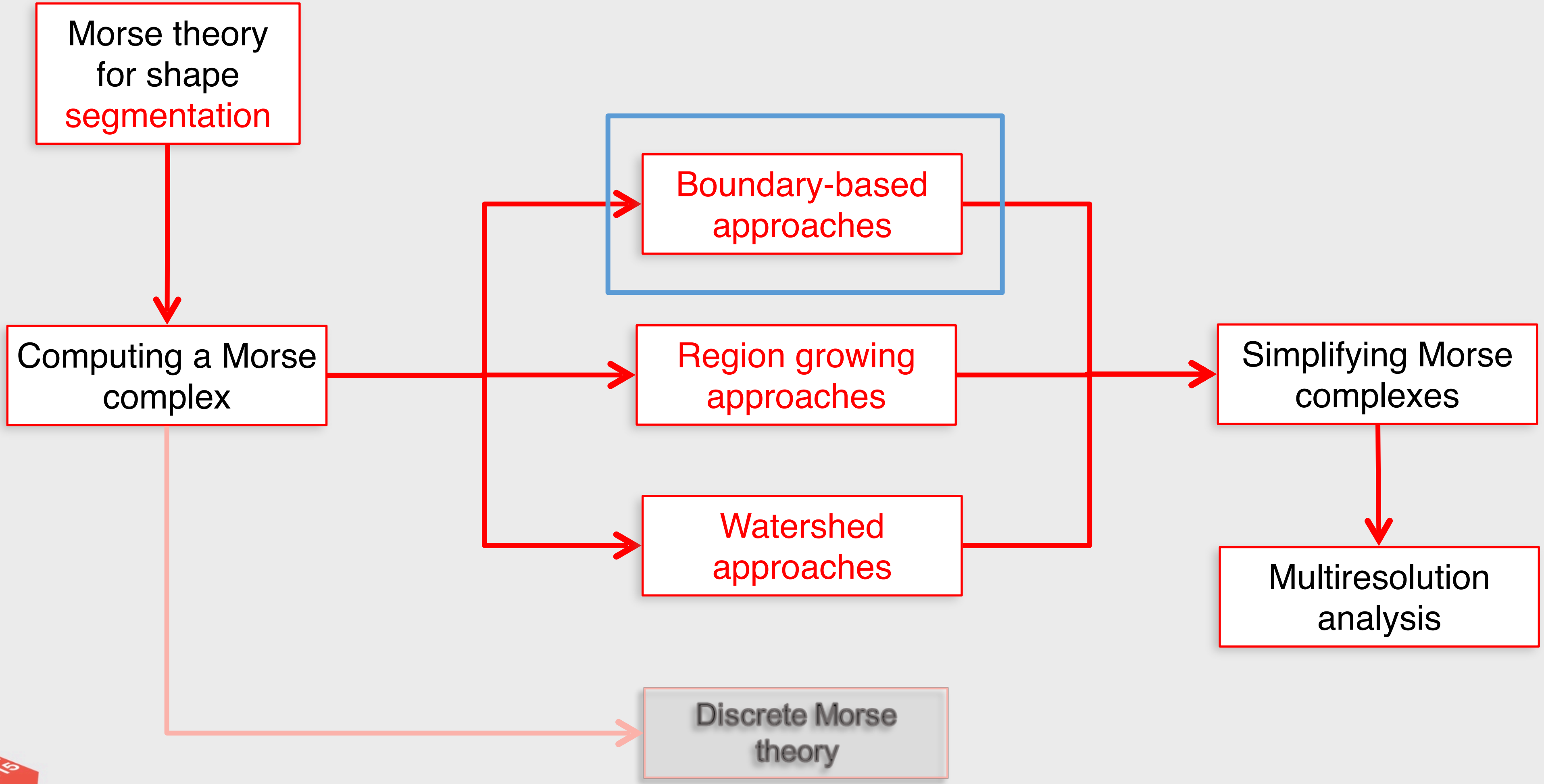

Maximum


Regular


1-saddle


2-saddle

Morse theory for shape segmentation



Boundary-based algorithms

- Widely used in terrain modeling and analysis
- Triangle (and tetrahedral) meshes: based on piecewise-linear Morse theory for critical point detection
- Regular square and cubic grids: based on computing C^0 or higher order interpolating functions over the grid
- Output:
    - 1-skeleton of the Morse-Smale complex in 2D (vertices and edges)
    - 2-skeleton of the Morse-Smale complex in 3D (vertices, edges and 2-cells)

Boundary based algorithms

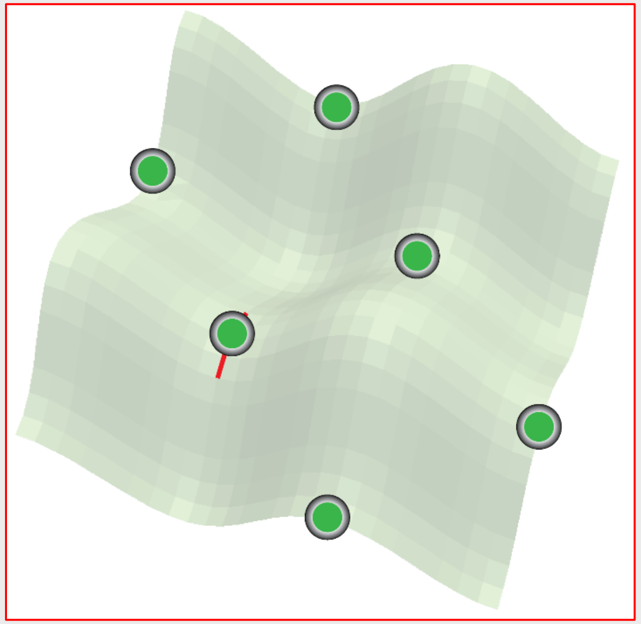
- General approach
  - Extraction of critical points
  - Computation of descending and ascending separatrix lines (from saddles to minima and maxima)
  - In 3D, also computation of separatrix surfaces

- On triangle meshes, separatrix lines are traced
  - along edges following the steepest descent/ascent *[Takahashi et al., 1995; [Edelsbrunner et al., 2001]*
  - along edges and inside triangles *[Bremer et al., 2004]*

- Just one algorithm for tetrahedral meshes *[Edelsbrunner et al., 2003]*
  - build descending Morse complex and then the ascending cells inside it
  - computational intensive.

Boundary based algorithms

- General approach
  - Extraction of critical points
  - Computation of descending and ascending separatrix lines (from saddles to minima and maxima)
  - In 3D, also computation of separatrix surfaces

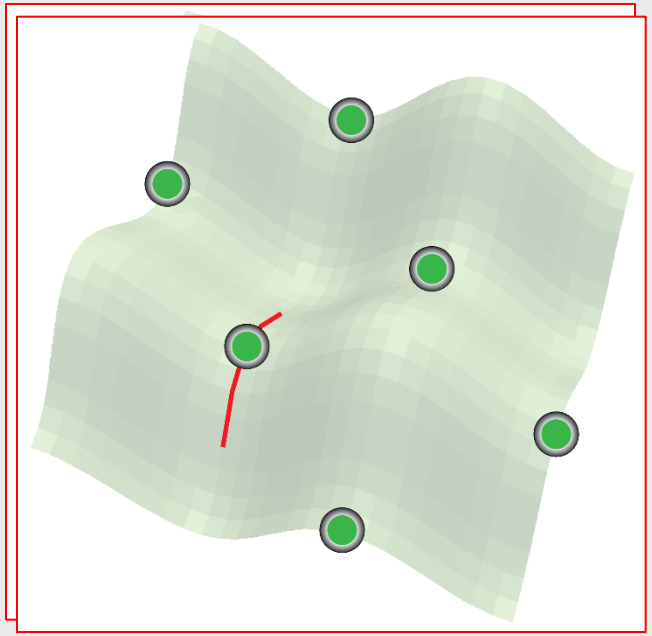


- On triangle meshes, separatrix lines are traced
  - along edges following the steepest descent/ascent *[Takahashi et al., 1995; [Edelsbrunner et al., 2001]*
  - along edges and inside triangles *[Bremer et al., 2004]*

- Just one algorithm for tetrahedral meshes *[Edelsbrunner et al., 2003]*
  - build descending Morse complex and then the ascending cells inside it
  - computational intensive.

Boundary based algorithms

- General approach
  - Extraction of critical points
  - Computation of descending and ascending separatrix lines (from saddles to minima and maxima)
  - In 3D, also computation of separatrix surfaces

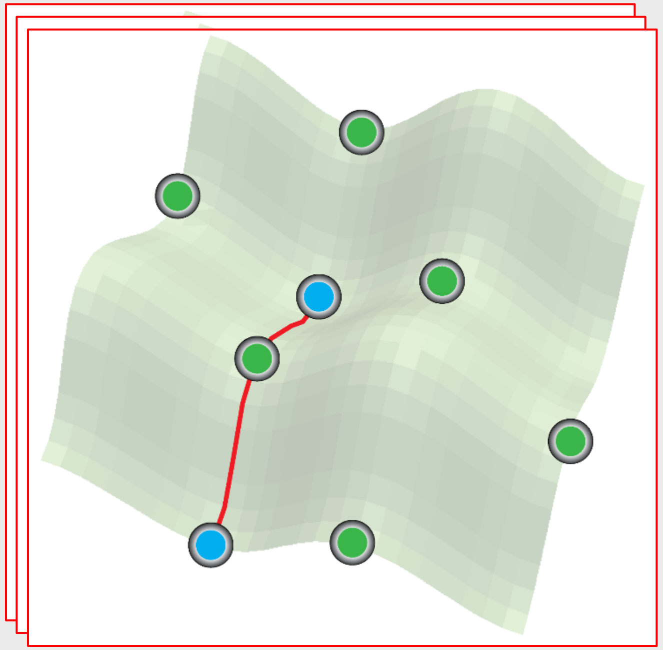


- On triangle meshes, separatrix lines are traced
  - along edges following the steepest descent/ascent *[Takahashi et al., 1995; [Edelsbrunner et al., 2001]*
  - along edges and inside triangles *[Bremer et al., 2004]*

- Just one algorithm for tetrahedral meshes *[Edelsbrunner et al., 2003]*
  - build descending Morse complex and then the ascending cells inside it
  - computational intensive.

Boundary based algorithms

- General approach
  - Extraction of critical points
  - Computation of descending and ascending separatrix lines (from saddles to minima and maxima)
  - In 3D, also computation of separatrix surfaces

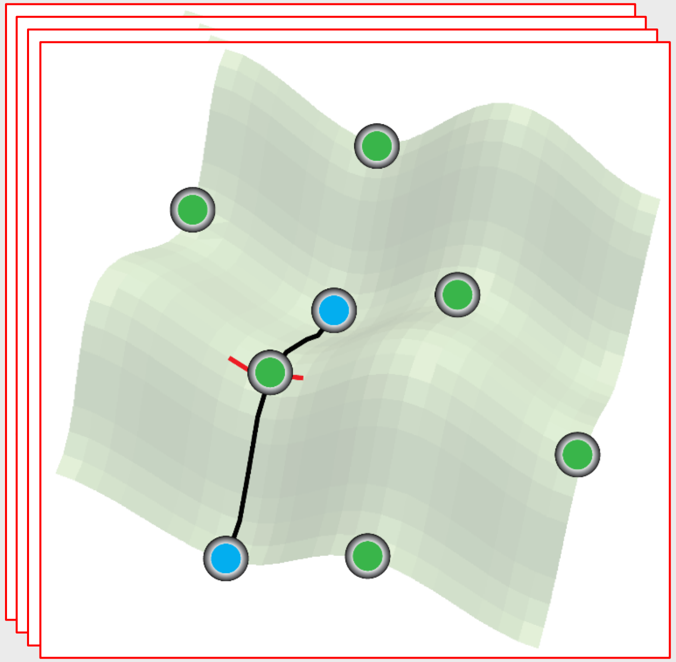


- On triangle meshes, separatrix lines are traced
  - along edges following the steepest descent/ascent *[Takahashi et al., 1995; [Edelsbrunner et al., 2001]*
  - along edges and inside triangles *[Bremer et al., 2004]*

- Just one algorithm for tetrahedral meshes *[Edelsbrunner et al., 2003]*
  - build descending Morse complex and then the ascending cells inside it
  - computational intensive.

Boundary based algorithms

- General approach
  - Extraction of critical points
  - Computation of descending and ascending separatrix lines (from saddles to minima and maxima)
  - In 3D, also computation of separatrix surfaces

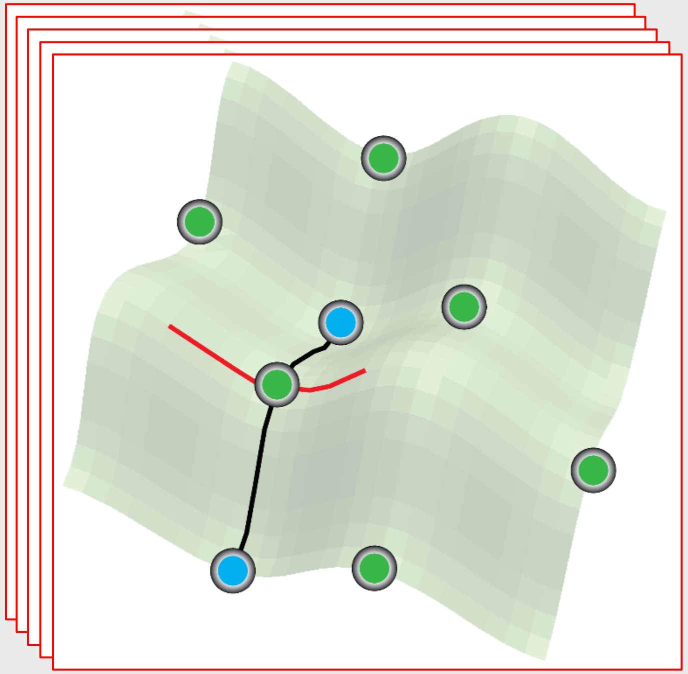


- On triangle meshes, separatrix lines are traced
  - along edges following the steepest descent/ascent *[Takahashi et al., 1995; [Edelsbrunner et al., 2001]*
  - along edges and inside triangles *[Bremer et al., 2004]*

- Just one algorithm for tetrahedral meshes *[Edelsbrunner et al., 2003]*
  - build descending Morse complex and then the ascending cells inside it
  - computational intensive.

Boundary based algorithms

- General approach
  - Extraction of critical points
  - Computation of descending and ascending separatrix lines (from saddles to minima and maxima)
  - In 3D, also computation of separatrix surfaces

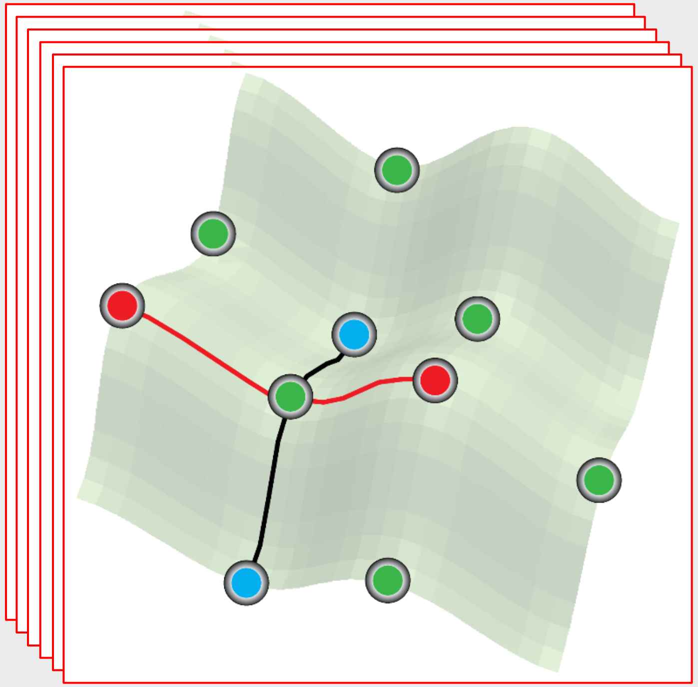


- On triangle meshes, separatrix lines are traced
  - along edges following the steepest descent/ascent *[Takahashi et al., 1995; [Edelsbrunner et al., 2001]*
  - along edges and inside triangles *[Bremer et al., 2004]*

- Just one algorithm for tetrahedral meshes *[Edelsbrunner et al., 2003]*
  - build descending Morse complex and then the ascending cells inside it
  - computational intensive.

Boundary based algorithms

- General approach
  - Extraction of critical points
  - Computation of descending and ascending separatrix lines (from saddles to minima and maxima)
  - In 3D, also computation of separatrix surfaces



- On triangle meshes, separatrix lines are traced
  - along edges following the steepest descent/ascent *[Takahashi et al., 1995; [Edelsbrunner et al., 2001]*
  - along edges and inside triangles *[Bremer et al., 2004]*

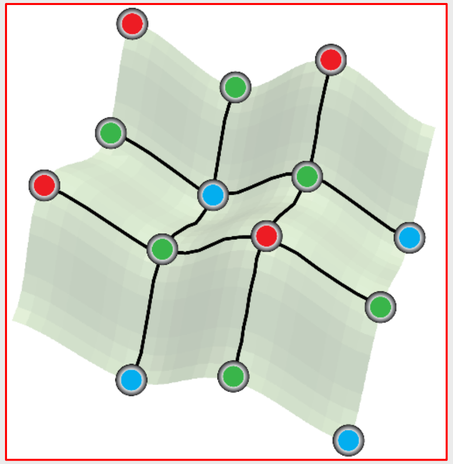
- Just one algorithm for tetrahedral meshes *[Edelsbrunner et al., 2003]*
  - build descending Morse complex and then the ascending cells inside it
  - computational intensive.

Boundary based algorithms

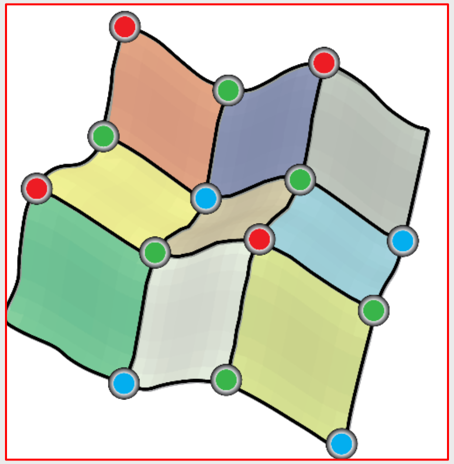
- General approach
  - Extraction of critical points
  - Computation of descending and ascending separatrix lines (from saddles to minima and maxima)
  - In 3D, also computation of separatrix surfaces

- On triangle meshes, separatrix lines are traced
  - along edges following the steepest descent/ascent *[Takahashi et al., 1995; [Edelsbrunner et al., 2001]*
  - along edges and inside triangles *[Bremer et al., 2004]*

- Just one algorithm for tetrahedral meshes *[Edelsbrunner et al., 2003]*
  - build descending Morse complex and then the ascending cells inside it
  - computational intensive.



Morse-Smale 1-skeleton



Morse-Smale complex

Boundary-based algorithms

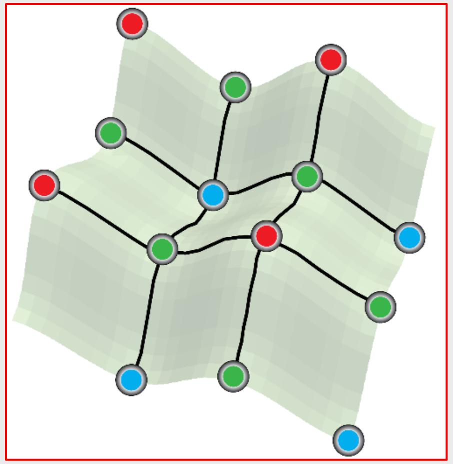
- General approach
  - Extraction of critical points
  - Computation of descending and ascending separatrix lines (from saddles to minima and maxima)
  - In 3D, also computation of separatrix surfaces

- On regular grids, different interpolants
  - C^1-differentiable Bernstein-Bezier bi-cubic (for 2D grids) or tri-cubic (for 3D grids) function *[Bajaj et al. 1998]*
  - Bi-linear C^0 function *[Schneider and Wood 2004]*
  - Bi-quadratic function with no overall continuity *[Schneider and Wood, 2005]*
  - Drawback: generation of additional critical points
  - Separatrix lines computed through numerical integration

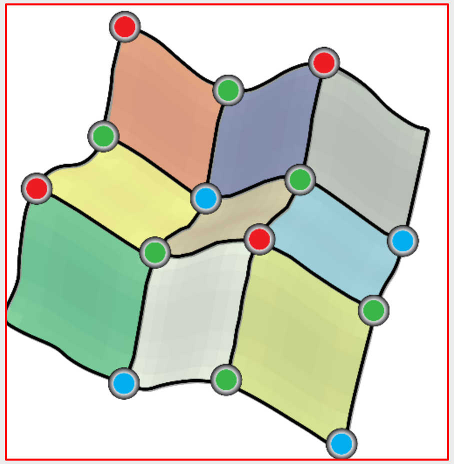
Boundary-based algorithms

- General approach
  - Extraction of critical points
  - Computation of descending and ascending separatrix lines (from saddles to minima and maxima)
  - In 3D, also computation of separatrix surfaces

- On regular grids, different interpolants
  - C^1-differentiable Bernstein-Bezier bi-cubic (for 2D grids) or tri-cubic (for 3D grids) function *[Bajaj et al. 1998]*
  - Bi-linear C^0 function *[Schneider and Wood 2004]*
  - Bi-quadratic function with no overall continuity *[Schneider and Wood, 2005]*
  - Drawback: generation of additional critical points
  - Separatrix lines computed through numerical integration

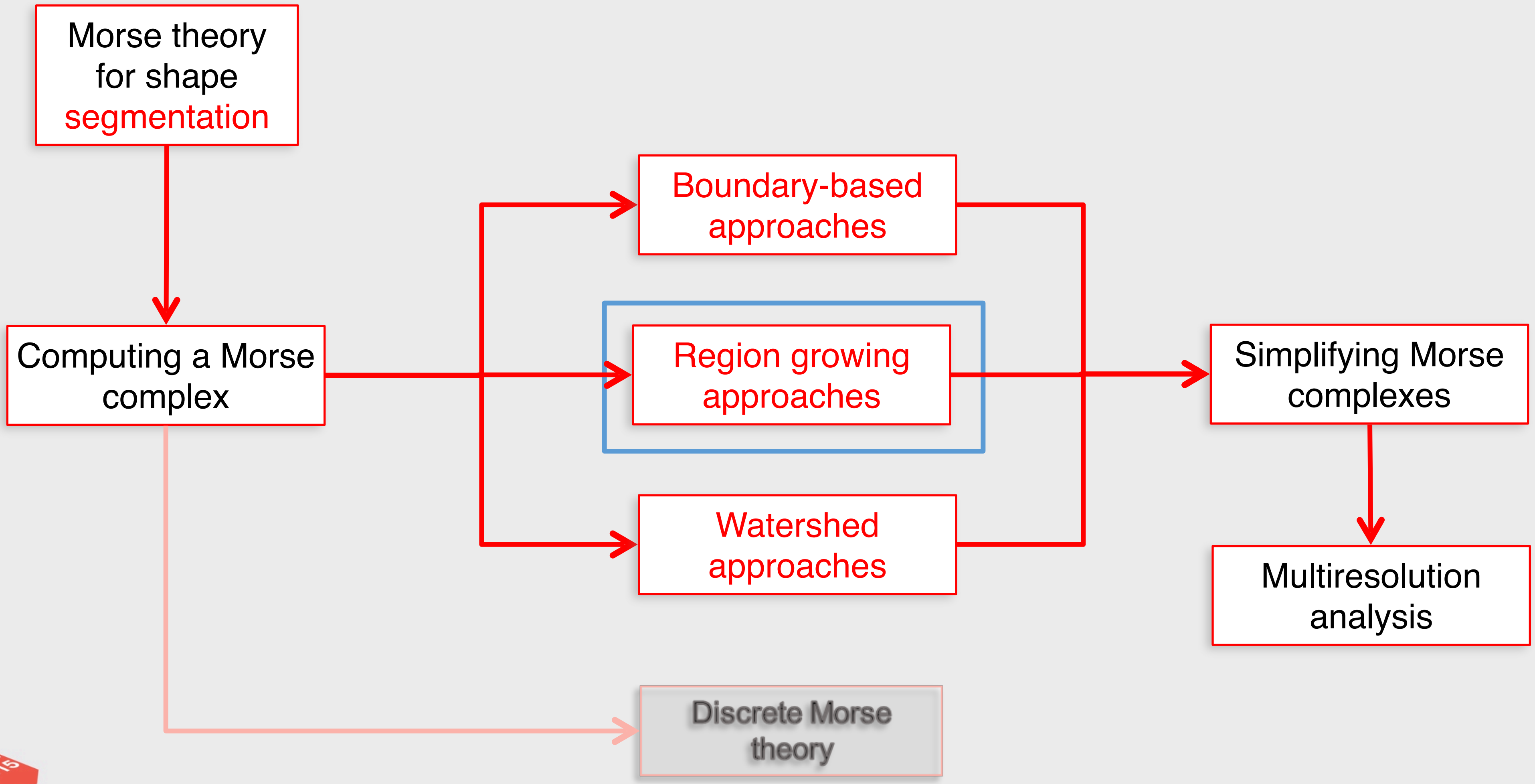


Morse-Smale 1-skeleton



Morse-Smale complex

Morse theory for shape segmentation



Region-growing algorithms

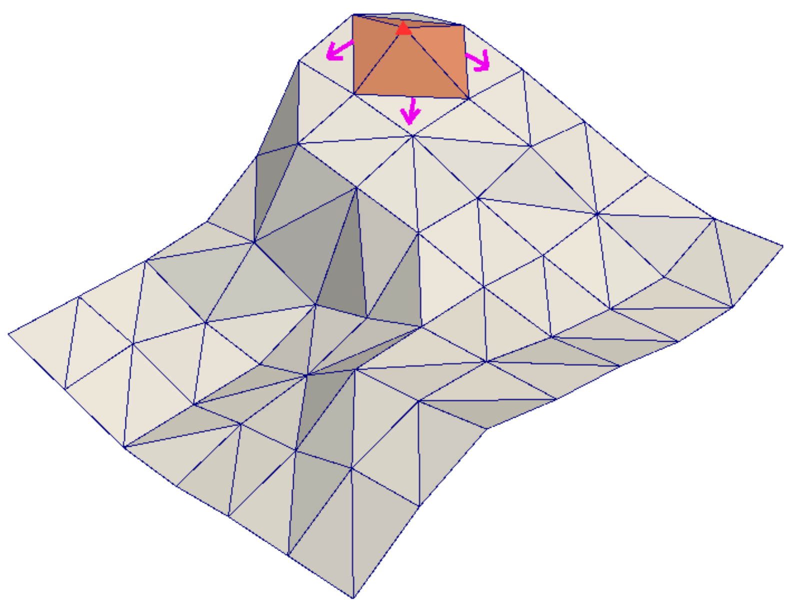
- General Approach:
  - Extract seed vertices (minima or maxima)
  - Grow regions from seeds by adding triangles/tetrahedra/vertices

- Adding triangles/tetrahedra *[Magillo et al, 1999; [Danovaro et al, 2003; Dey et al., 2003]*
  - on triangle/tetrahedral meshes

- Adding vertices *[Gyulassy et. al, 2007]*
  - on regular cubic grids

Region-growing algorithms

- General Approach:
  - Extract seed vertices (minima or maxima)
  - Grow regions from seeds by adding triangles/tetrahedra/vertices

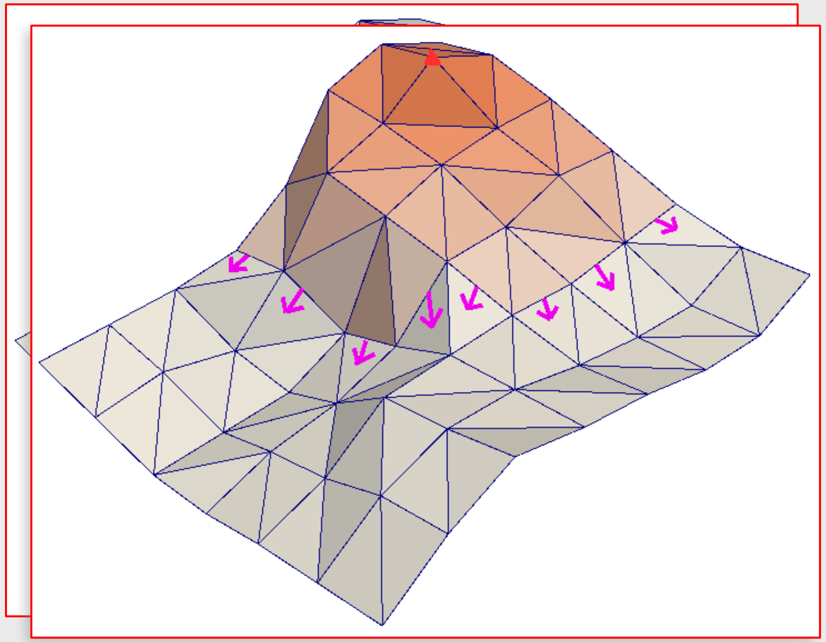


- Adding triangles/tetrahedra *[Magillo et al, 1999; [Danovaro et al, 2003; Dey et al., 2003]*
  - on triangle/tetrahedral meshes

- Adding vertices *[Gyulassy et. al, 2007]*
  - on regular cubic grids

Region-growing algorithms

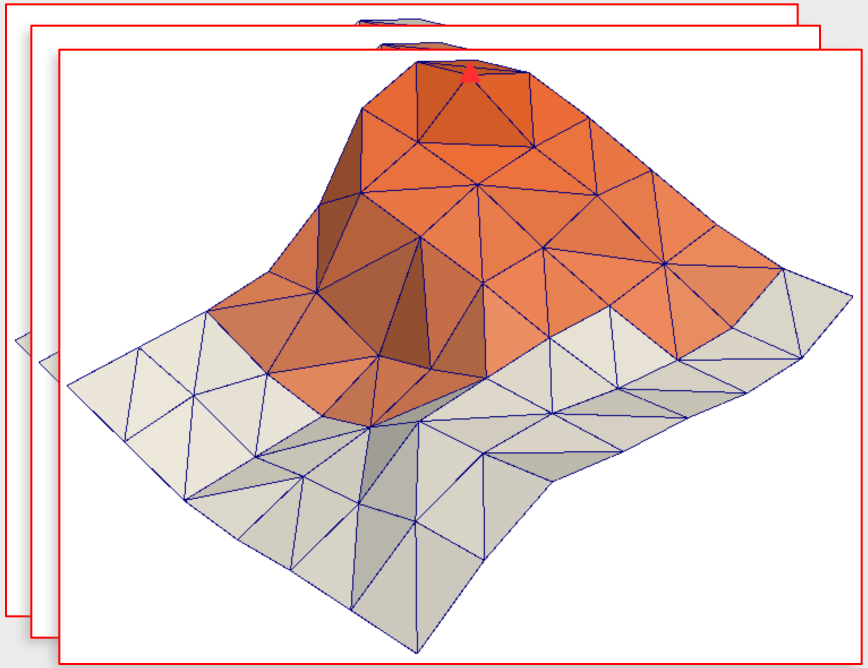
- General Approach:
  - Extract seed vertices (minima or maxima)
  - Grow regions from seeds by adding triangles/tetrahedra/vertices



- Adding triangles/tetrahedra *[Magillo et al, 1999; [Danovaro et al, 2003; Dey et al., 2003]*
  - on triangle/tetrahedral meshes
- Adding vertices *[Gyulassy et. al, 2007]*
  - on regular cubic grids

Region-growing algorithms

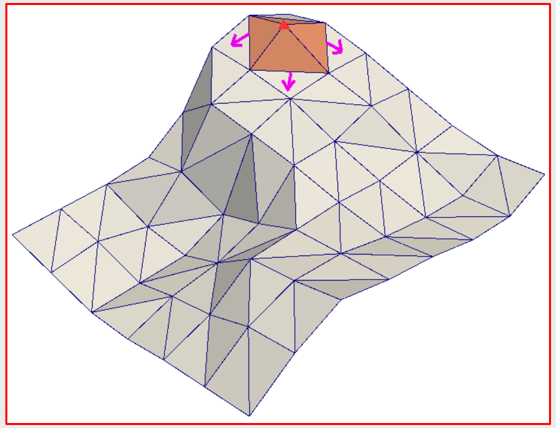
- General Approach:
  - Extract seed vertices (minima or maxima)
  - Grow regions from seeds by adding triangles/tetrahedra/vertices

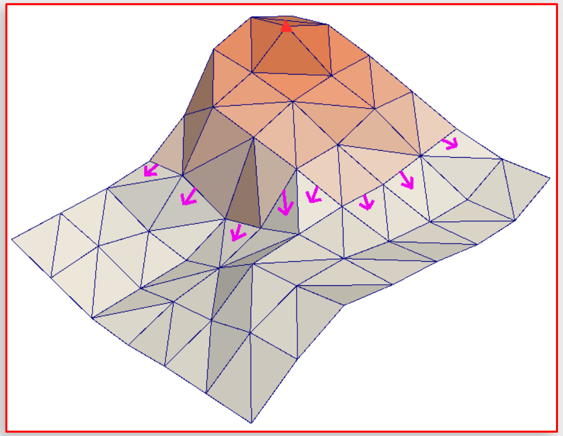


- Adding triangles/tetrahedra *[Magillo et al, 1999; [Danovaro et al, 2003; Dey et al., 2003]*
  - on triangle/tetrahedral meshes
- Adding vertices *[Gyulassy et. al, 2007]*
  - on regular cubic grids

Region-growing algorithms

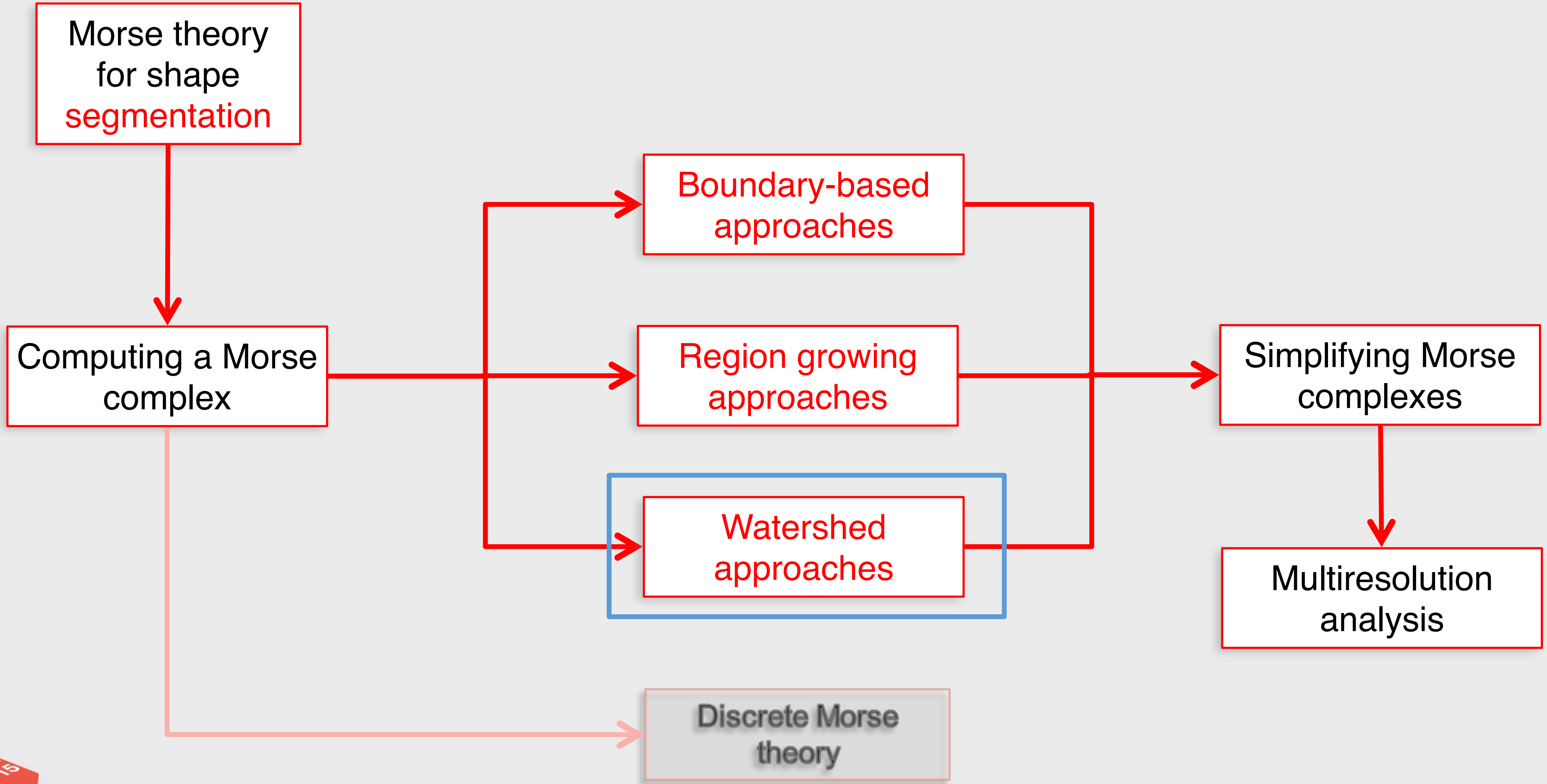
- General Approach:
  - Extract seed vertices (minima or maxima)
  - Grow regions from seeds by adding triangles/tetrahedra/vertices





- Critical point detection based on piecewise-linear Morse theory
- Output:
  - ascending/ descending 2-cells (3-cells) as collections of triangles (tetrahedra)
  - cells of the Morse-Smale complex as collections of vertices *[Gyulassy et. al, 2007]*

Morse theory for shape segmentation



The watershed transform

- Basic definitions:
  - Catchment basin of *p* – set of points in *M* closer to *p* than to any other critical point according to the topographic distance
  - Watershed lines – points of *M* which do not belong to any catchment basin

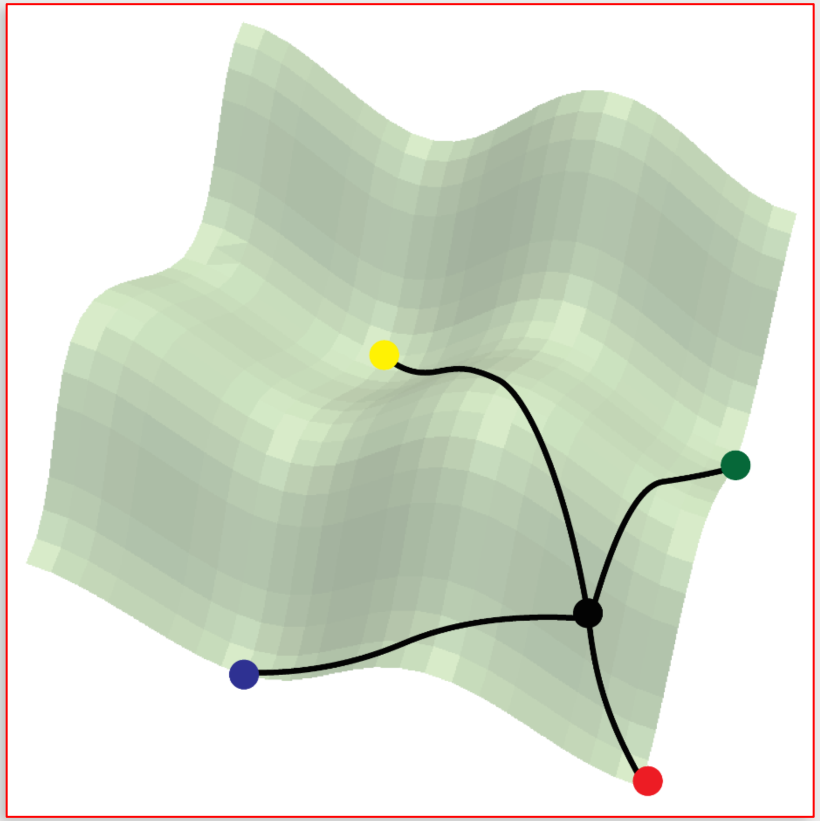
- Watershed and Morse theory
  - closure of the catchment basins correspond to closure of the ascending maximal Morse cells

- Topographic distance between two points p and q:

$$T_D(p,q) = \inf \int \| \nabla f(P(s)) \| \, ds$$

The watershed transform

- Basic definitions:
  - Catchment basin of *p* – set of points in *M* closer to *p* than to any other critical point according to the topographic distance
  - Watershed lines – points of *M* which do not belong to any catchment basin



- Watershed and Morse theory
  - closure of the catchment basins correspond to closure of the ascending maximal Morse cells
- Topographic distance between two points p and q:

$$T_D(p,q) = \inf \int \| \nabla f(P(s)) \| \, ds$$

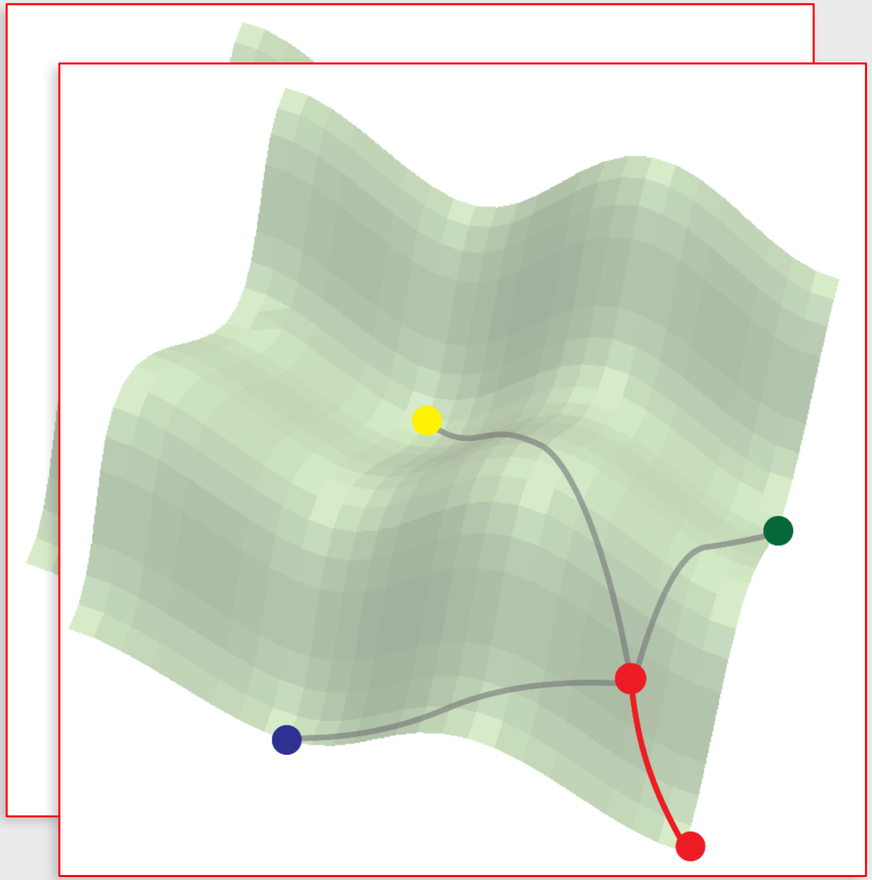
The watershed transform

- Basic definitions:
  - Catchment basin of *p* – set of points in *M* closer to *p* than to any other critical point according to the topographic distance
  - Watershed lines – points of *M* which do not belong to any catchment basin

- Watershed and Morse theory
  - closure of the catchment basins correspond to closure of the ascending maximal Morse cells

- Topographic distance between two points p and q:

$$T_D(p,q) = \inf \int \| \nabla f(P(s)) \| \, ds$$



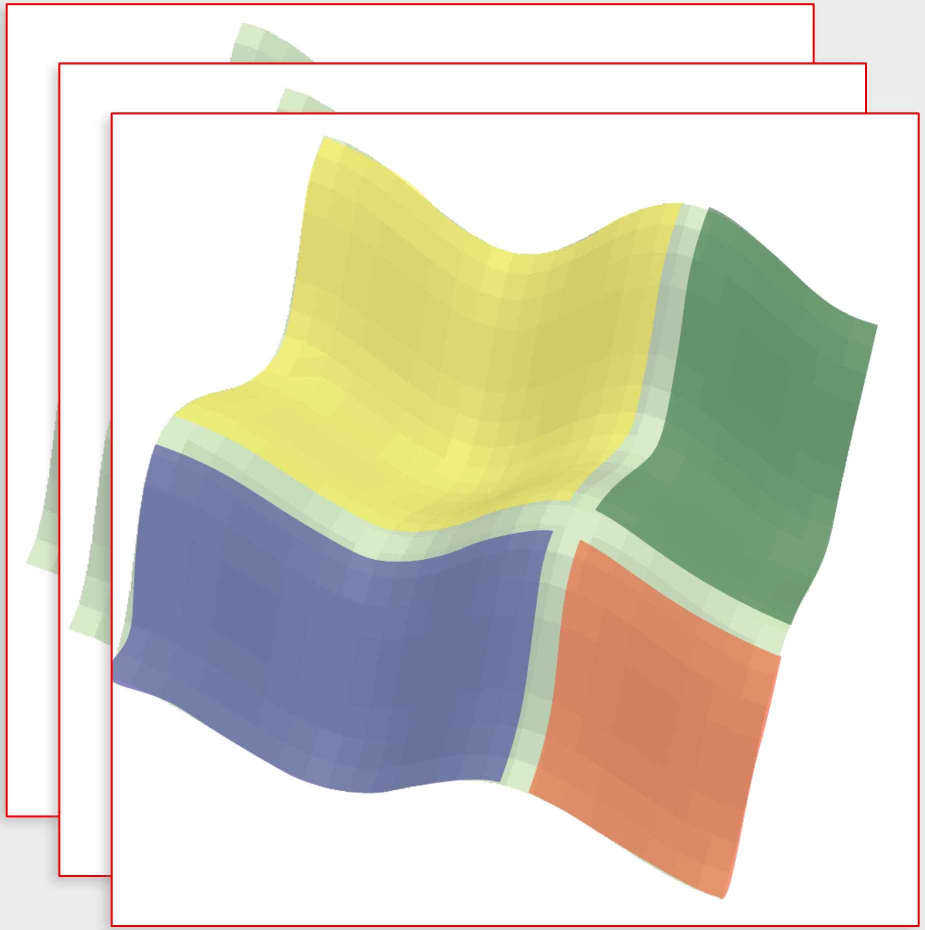
The watershed transform

- Basic definitions:
  - Catchment basin of *p* – set of points in *M* closer to *p* than to any other critical point according to the topographic distance
  - Watershed lines – points of *M* which do not belong to any catchment basin

- Watershed and Morse theory
  - closure of the catchment basins correspond to closure of the ascending maximal Morse cells

- Topographic distance between two points p and q:

$$T_D(p,q) = \inf \int \| \nabla f(P(s)) \| \, ds$$



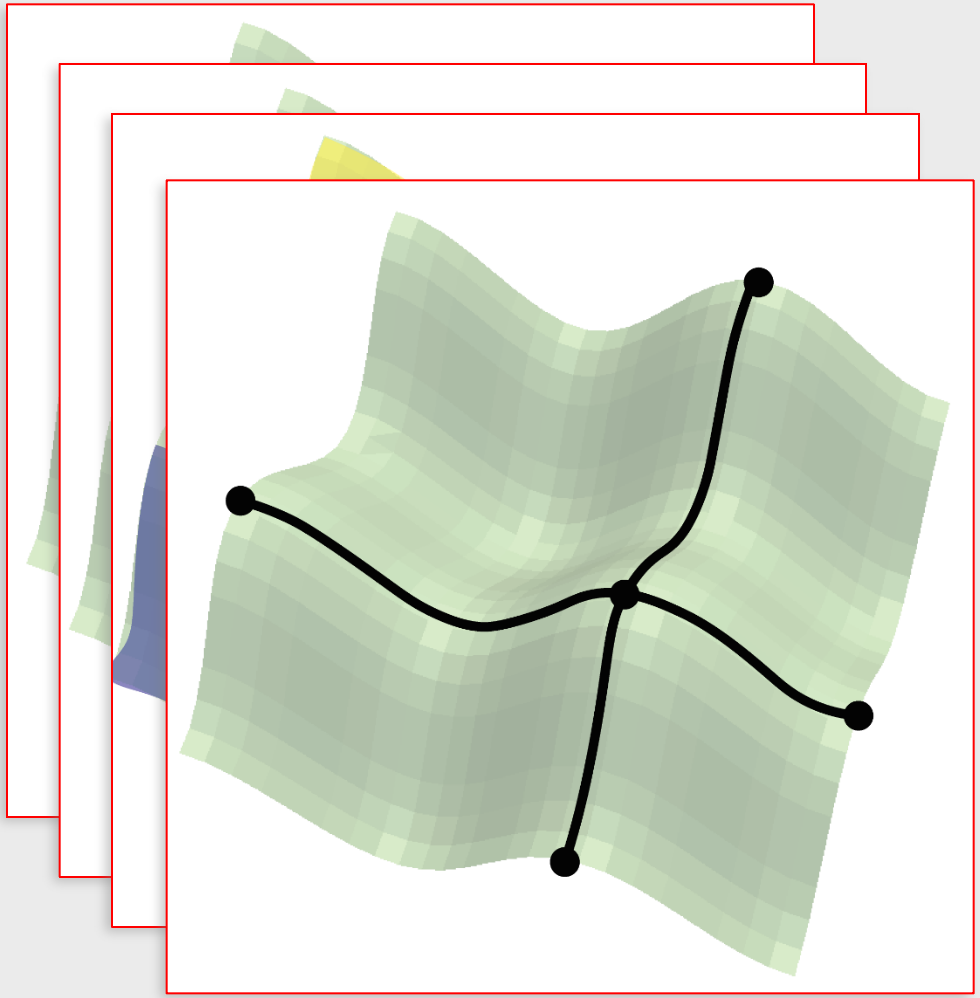
The watershed transform

- Basic definitions:
  - Catchment basin of p – set of points in M closer to p than to any other critical point according to the topographic distance
  - Watershed lines – points of M which do not belong to any catchment basin

- Watershed and Morse theory
  - closure of the catchment basins correspond to closure of the ascending maximal Morse cells

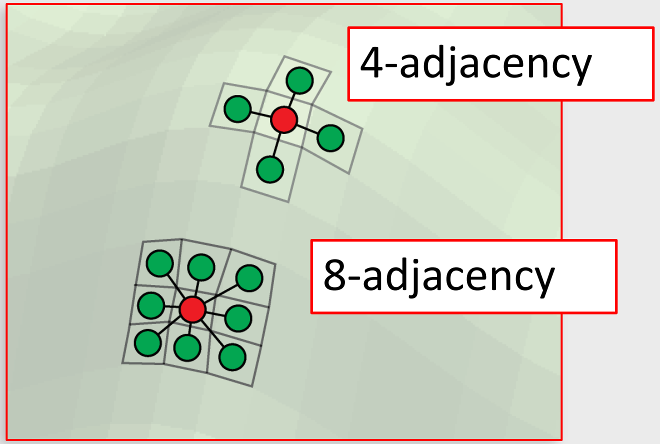
- Topographic distance between two points p and q:

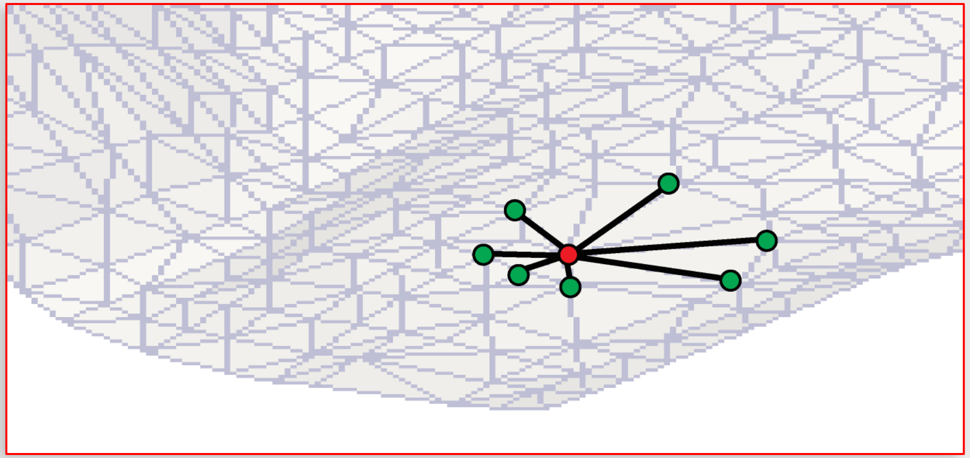
$$T_D(p,q) = \inf \int \| \nabla f(P(s)) \| \, ds$$



The watershed transform – discrete definition

- Defined on labeled graph G=(N,A) with a field value associated with each node in N
  - Regular grids:
    - Nodes in N are pixels/voxels
    - Arcs in A define the adjacency relation between pixels/voxels
  - Triangle/tetrahedral meshes:
    - Nodes in N are the vertices
    - Arcs in A are edges between adjacent vertices



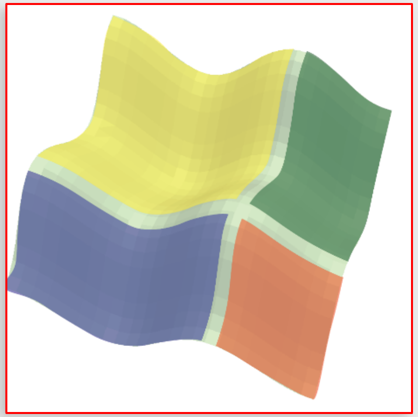


- Discrete topographic distance

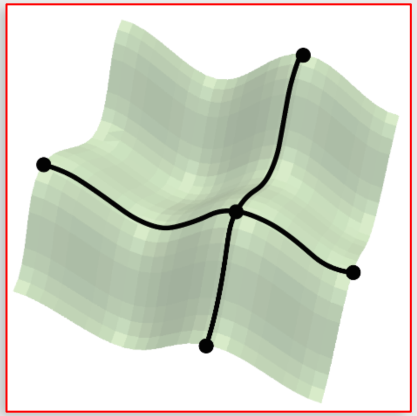
  T(p,q) = *min* {*cost* (γ) | γ path from *p* to *q* in G}

The watershed transform – algorithms

- General approach:
  - Works on labeled graph G
  - Produces catchment basins as a classification of the nodes of G



- Algorithms based on:
  - Topographic distance *[Meyer and Beucher 1990, Meyer 1994]*
    - discrete topographic distance as a path in graph G
    - application of Dijkstra's algorithm
  - Simulated Immersion *[Vincent and Soille 1991, Soille 2004]*
  - Rain falling simulation *[Mangan and Whitaker 1999, Stoev and Strasser 2000]*
- Survey *[Roerdink and Meijster, 2000]*



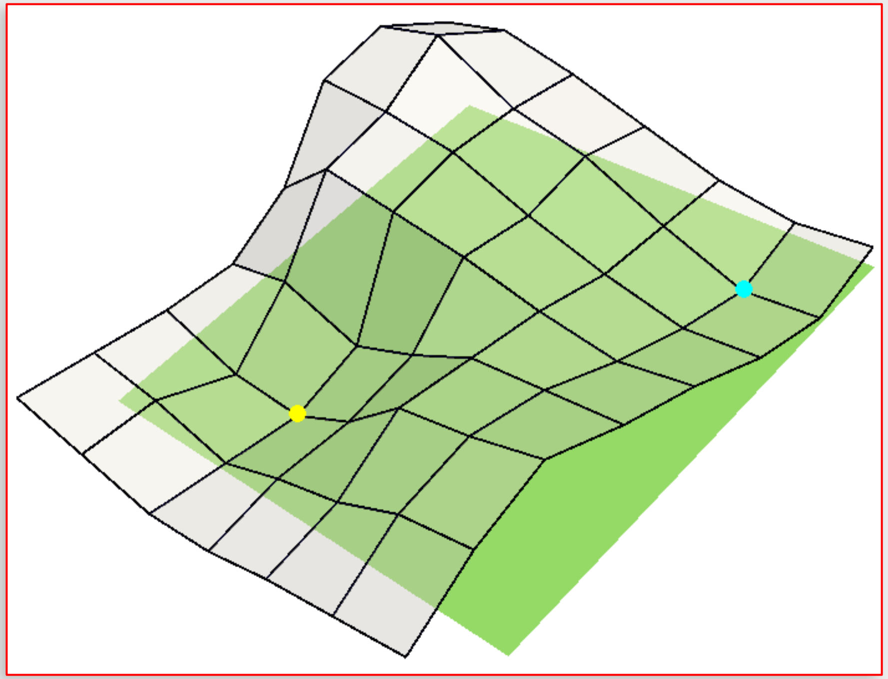
The watershed transform - algorithms

- General approach:
  - Works on labeled graph G
  - Produces catchment basins as a classification of the nodes of G

- Methods based on:
  - Topographic distance
    - Image integration [Meyer and Beucher 1990, Meyer 1994]
    - Hill climbing [Meyer 1994]
  - Simulated Immersion *[Vincent and Soille 1991, Soille 2004]*
  - Rain falling simulation [Mangan and Whitaker 1999, Stoev and Strasser 2000]

The watershed transform - algorithms

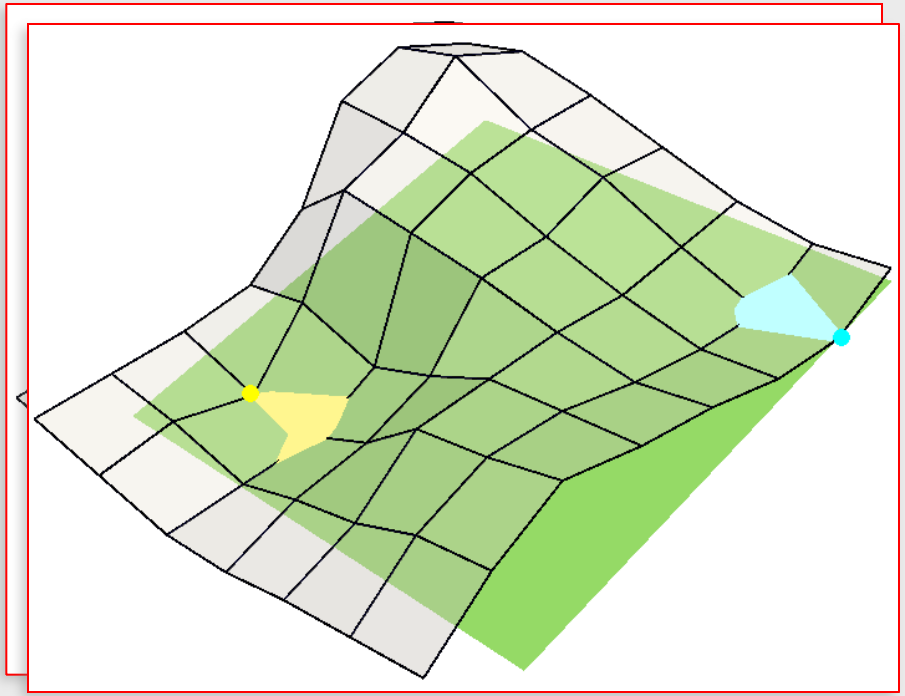
- General approach:
  - Works on labeled graph G
  - Produces catchment basins as a classification of the nodes of G



- Methods based on:
  - Topographic distance
    - Image integration [Meyer and Beucher 1990, Meyer 1994]
    - Hill climbing [Meyer 1994]
  - Simulated Immersion *[Vincent and Soille 1991, Soille 2004]*
  - Rain falling simulation [Mangan and Whitaker 1999, Stoev and Strasser 2000]

The watershed transform - algorithms

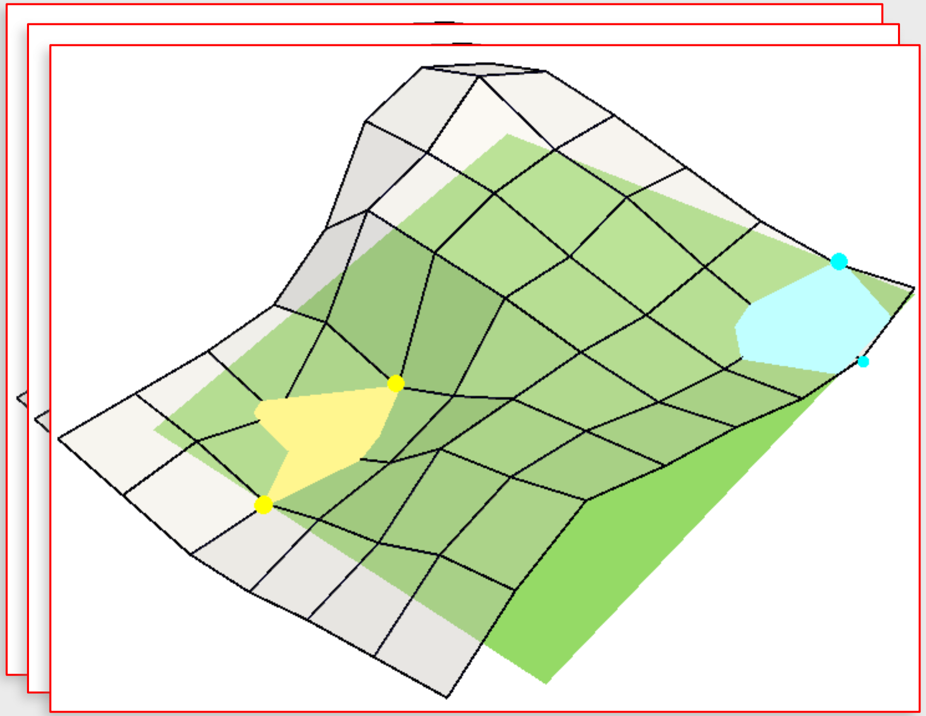
- General approach:
  - Works on labeled graph G
  - Produces catchment basins as a classification of the nodes of G



- Methods based on:
  - Topographic distance
    - Image integration [Meyer and Beucher 1990, Meyer 1994]
    - Hill climbing [Meyer 1994]
  - Simulated Immersion *[Vincent and Soille 1991, Soille 2004]*
  - Rain falling simulation [Mangan and Whitaker 1999, Stoev and Strasser 2000]

The watershed transform - algorithms

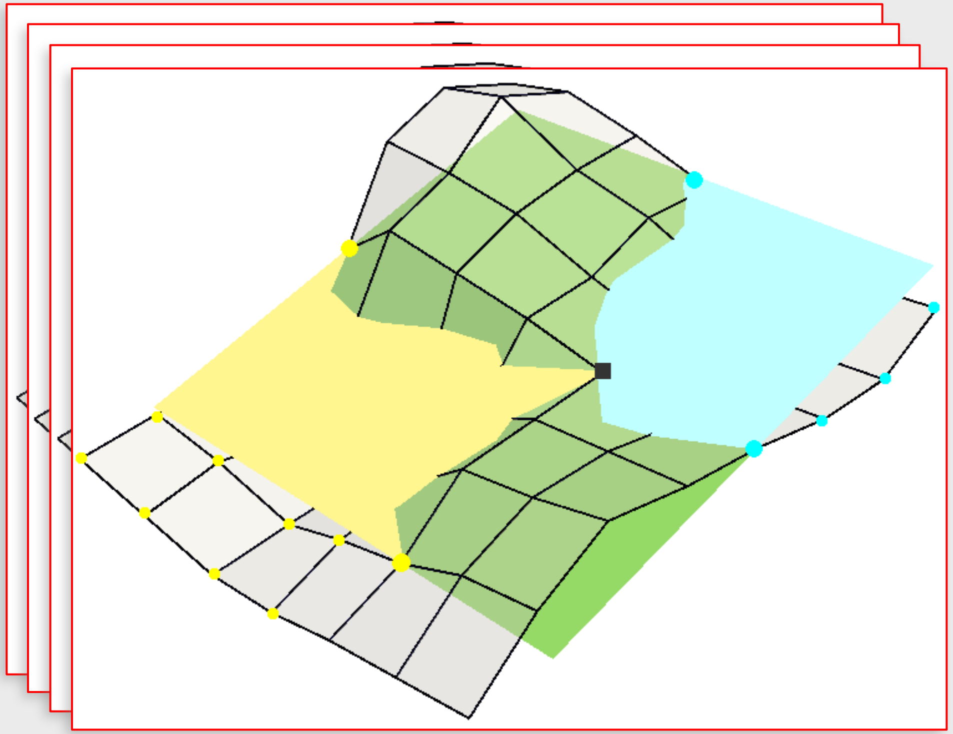
- General approach:
  - Works on labeled graph G
  - Produces catchment basins as a classification of the nodes of G



- Methods based on:
  - Topographic distance
    - Image integration [Meyer and Beucher 1990, Meyer 1994]
    - Hill climbing [Meyer 1994]
  - Simulated Immersion *[Vincent and Soille 1991, Soille 2004]*
  - Rain falling simulation [Mangan and Whitaker 1999, Stoev and Strasser 2000]

The watershed transform - algorithms

- General approach:
  - Works on labeled graph G
  - Produces catchment basins as a classification of the nodes of G



- Methods based on:
  - Topographic distance
    - Image integration [Meyer and Beucher 1990, Meyer 1994]
    - Hill climbing [Meyer 1994]
  - Simulated Immersion *[Vincent and Soille 1991, Soille 2004]*
  - Rain falling simulation [Mangan and Whitaker 1999, Stoev and Strasser 2000]

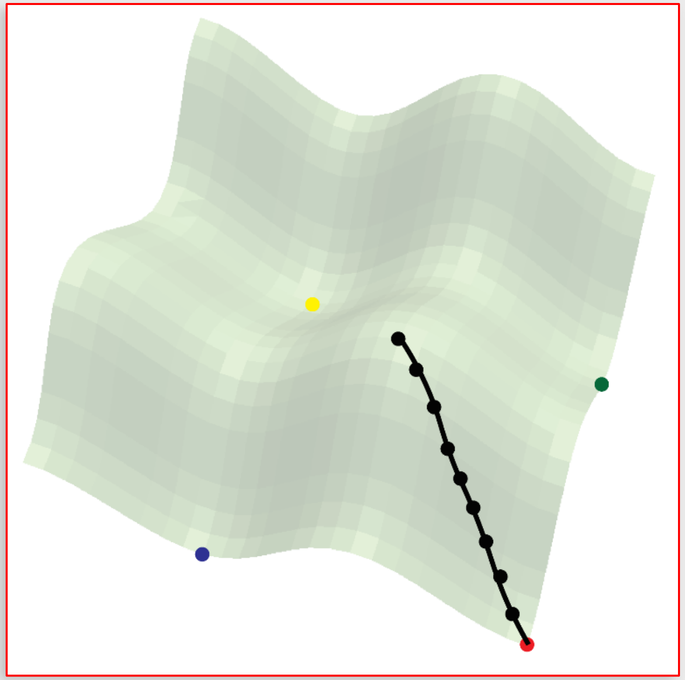
The watershed transform - algorithms

- General approach:
  - Works on labeled graph G
  - Produces catchment basins as a classification of the nodes of G

- Methods based on:
  - Topographic distance
    - Image integration [Meyer and Beucher 1990, Meyer 1994]
    - Hill climbing [Meyer 1994]
  - Simulated Immersion [Vincent and Soille 1991, Soille 2004]
  - Rain falling simulation *[Mangan and Whitaker 1999, Stoev and Strasser 2000]*

The watershed transform - algorithms

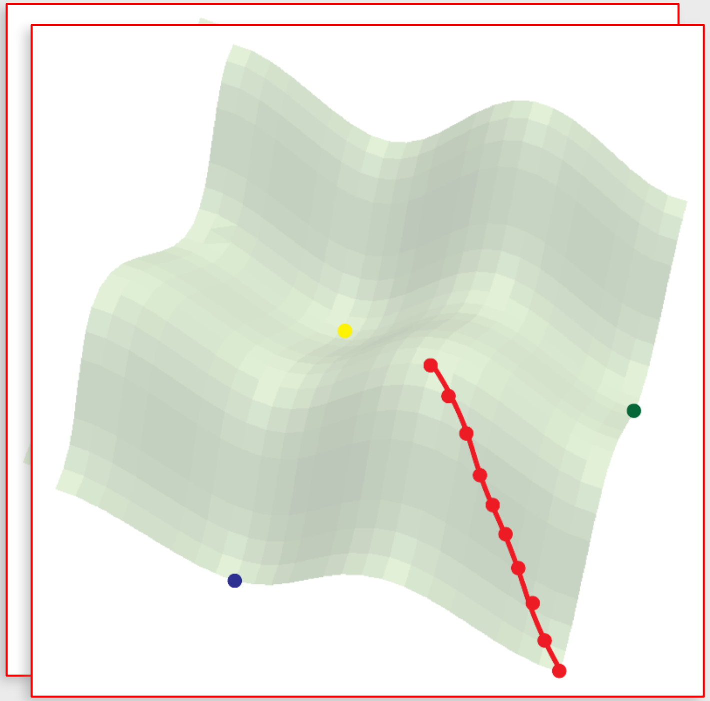
- General approach:
  - Works on labeled graph G
  - Produces catchment basins as a classification of the nodes of G



- Methods based on:
  - Topographic distance
    - Image integration [Meyer and Beucher 1990, Meyer 1994]
    - Hill climbing [Meyer 1994]
  - Simulated Immersion [Vincent and Soille 1991, Soille 2004]
  - Rain falling simulation *[Mangan and Whitaker 1999, Stoev and Strasser 2000]*

The watershed transform - algorithms

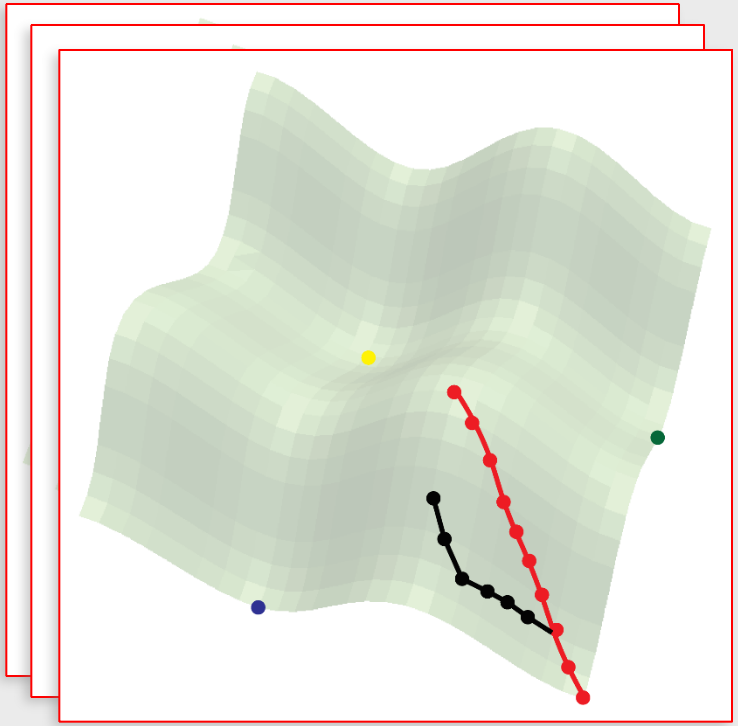
- General approach:
  - Works on labeled graph G
  - Produces catchment basins as a classification of the nodes of G



- Methods based on:
  - Topographic distance
    - Image integration [Meyer and Beucher 1990, Meyer 1994]
    - Hill climbing [Meyer 1994]
  - Simulated Immersion [Vincent and Soille 1991, Soille 2004]
  - Rain falling simulation *[Mangan and Whitaker 1999, Stoev and Strasser 2000]*

The watershed transform - algorithms

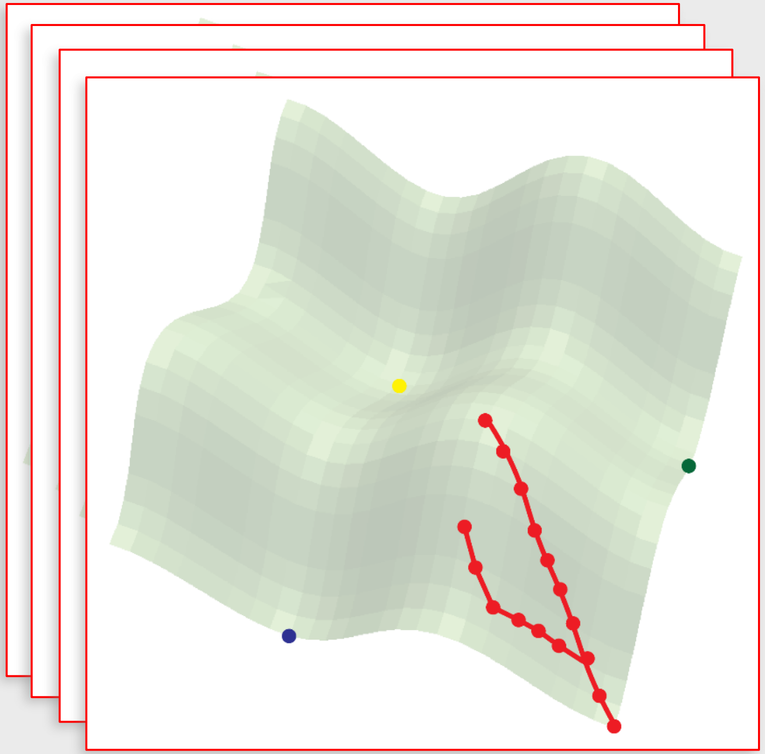
- General approach:
    - Works on labeled graph G
    - Produces catchment basins as a classification of the nodes of G



- Methods based on:
    - Topographic distance
        - Image integration [Meyer and Beucher 1990, Meyer 1994]
        - Hill climbing [Meyer 1994]
    - Simulated Immersion [Vincent and Soille 1991, Soille 2004]
    - Rain falling simulation *[Mangan and Whitaker 1999, Stoev and Strasser 2000]*

The watershed transform - algorithms

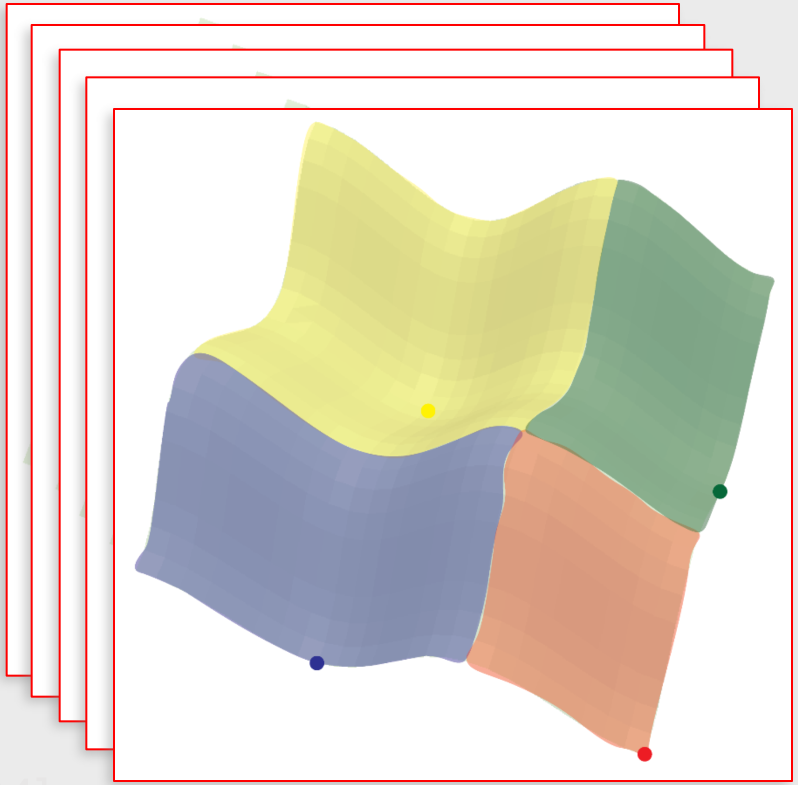
- General approach:
  - Works on labeled graph G
  - Produces catchment basins as a classification of the nodes of G



- Methods based on:
  - Topographic distance
    - Image integration [Meyer and Beucher 1990, Meyer 1994]
    - Hill climbing [Meyer 1994]
  - Simulated Immersion [Vincent and Soille 1991, Soille 2004]
  - Rain falling simulation *[Mangan and Whitaker 1999, Stoev and Strasser 2000]*

The watershed transform - algorithms

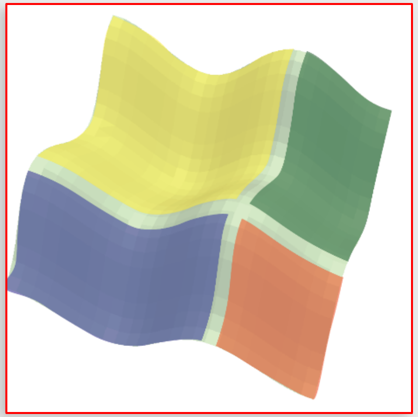
- General approach:
  - Works on labeled graph G
  - Produces catchment basins as a classification of the nodes of G



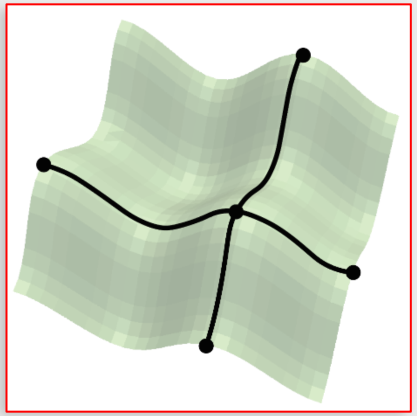
- Methods based on:
  - Topographic distance
    - Image integration [Meyer and Beucher 1990, Meyer 1994]
    - Hill climbing [Meyer 1994]
  - Simulated Immersion [Vincent and Soille 1991, Soille 2004]
  - Rain falling simulation *[Mangan and Whitaker 1999, Stoev and Strasser 2000]*

The watershed transform - algorithms

- General approach:
  - Works on labeled graph G
  - Produces catchment basins as a classification of the nodes of G



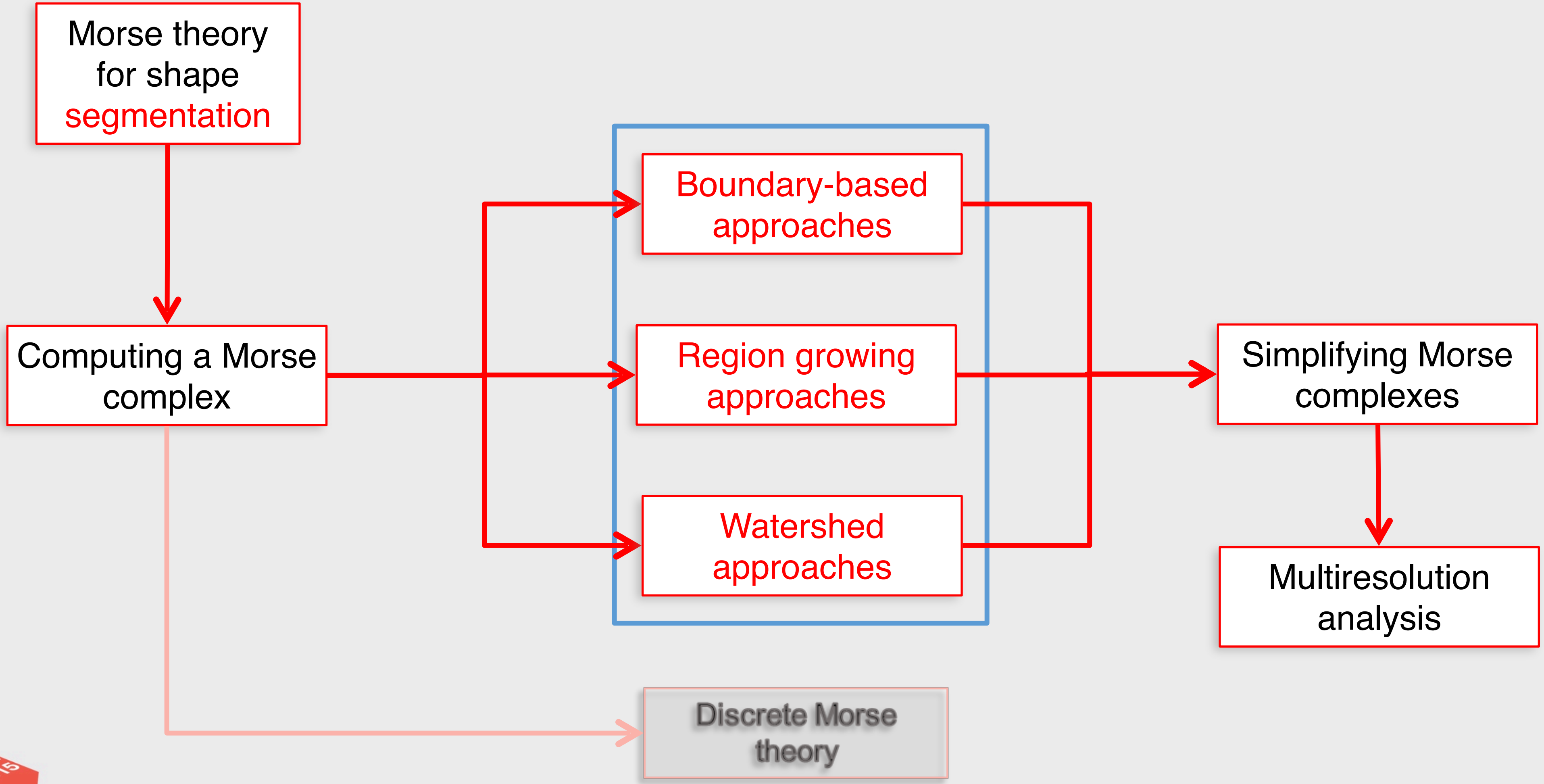
- Dimension-independent
- All algorithms produce comparable results
- For meshes, labeling of the nodes of graph G extended to triangles and tetrahedra
- Output:
- descending or ascending Morse maximal cells as collections of maximal cells of the input simplicial mesh or regular grid



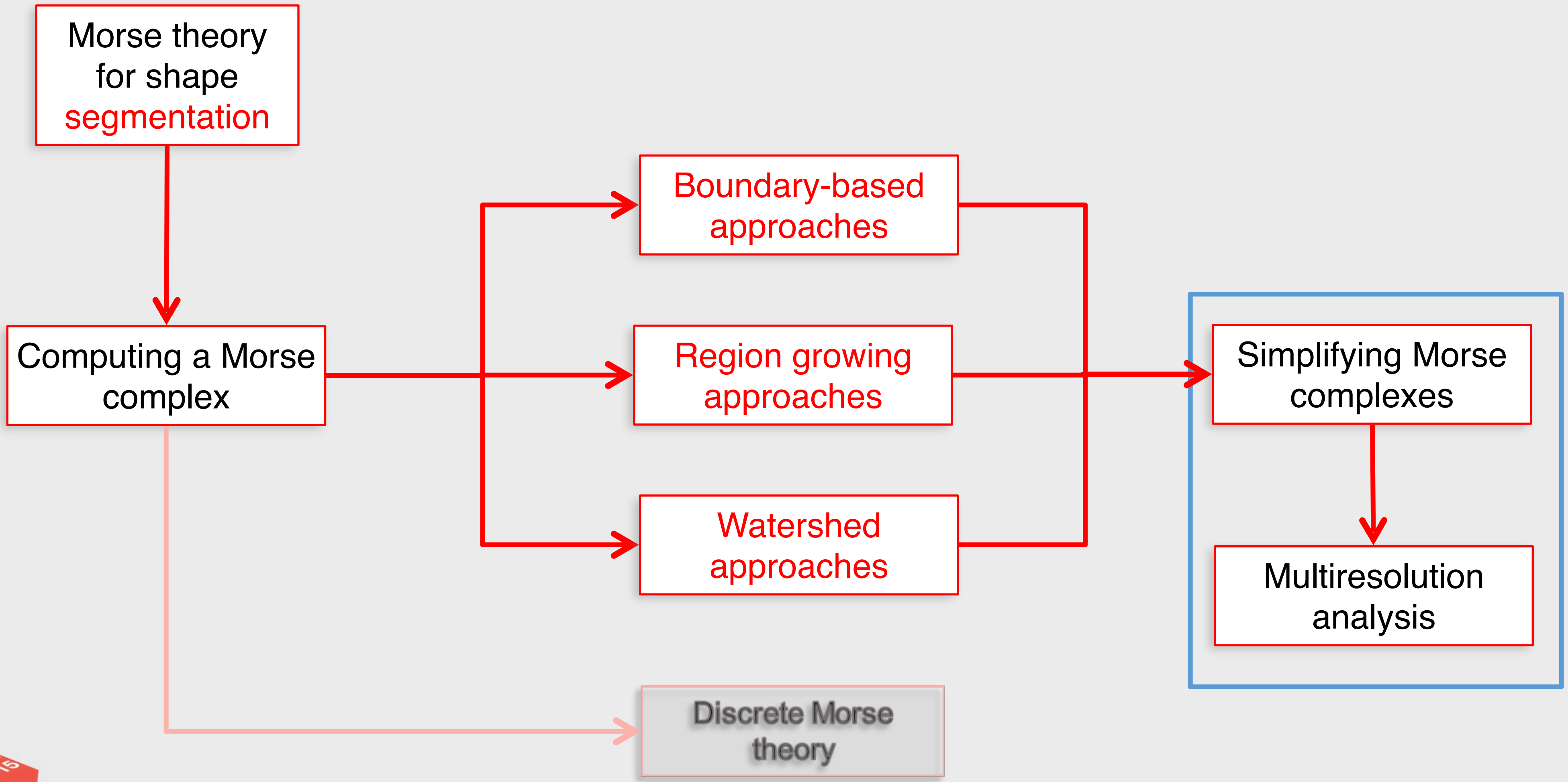
Morse theory for shape segmentation

Approach	Input	Output	Algorithm
Boundary-based	Triangle mesh	Morse-Smale	*Takahashi et al. 1995*
			Edelsbrunner et al. 2001
			Bremer et al. 2004
	Tetrahedral mesh	Morse-Smale	*Edelsbrunner et al. 2003*
	2D/3D grid	Morse-Smale	*Bajaj et al. 1998*
	2D grid	Morse-Smale	*Schneider and Wood 2004, 2005*
Region-based	Triangle mesh	Morse	*Magillo et al., 1999*
			Danovaro et al., 2003
	Tetrahedral mesh	Morse-Smale	*Gyulassy et al., 2007*
Watershed	any	Morse	*(topographic distance) Meyer and Beucher 1990*
	Grid	Morse	(topographic distance) *Meyer 1994*
	Any	Morse	(immersion) *Vincent and Soille 1991, Soille 2004*
	Triangle mesh	Morse	(rain) *Mangan and Whitaker 1999*
	Grid	Morse	(rain) *Stoev and Strasser 2000*

Morse theory for shape segmentation

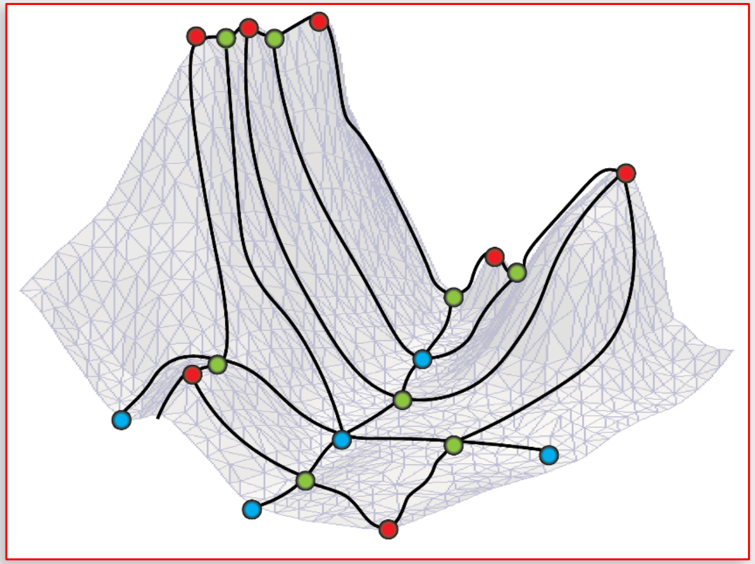


Morse theory for shape segmentation

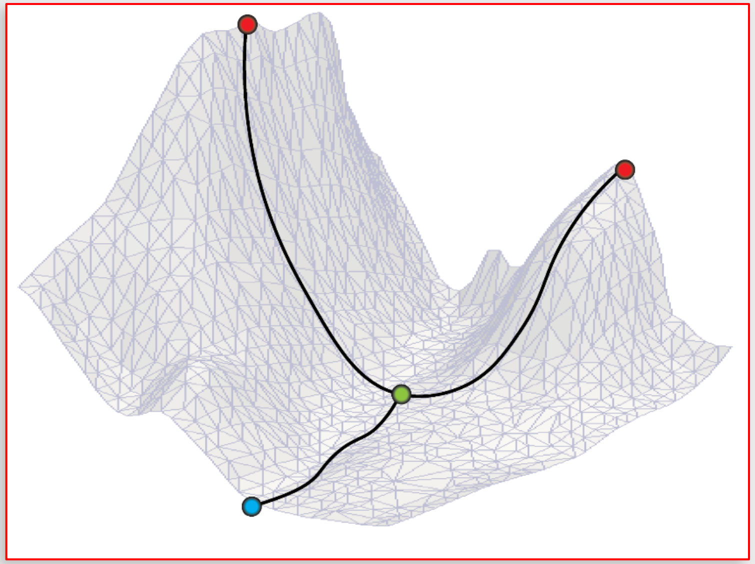


Topological Simplification

- Topological simplification is a fundamental tool for eliminating noise and irrelevant features in a topological description of a shape



From a noisy representation to a simplified representation focusing on relevant features

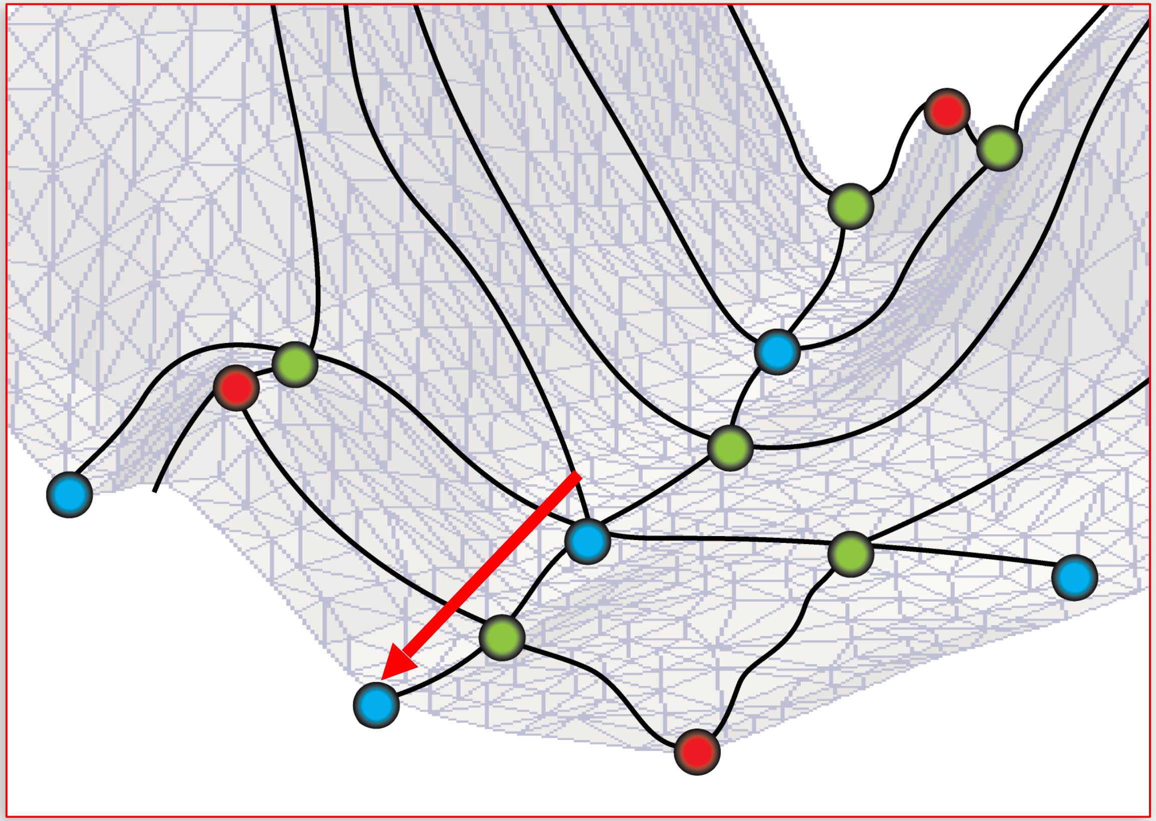


Topological Simplification

- Simplifications organized in a sequence:
  - importance value assigned to each simplification [Edelsbrunner et al., 2002]
- From a sequence we can build progressive models

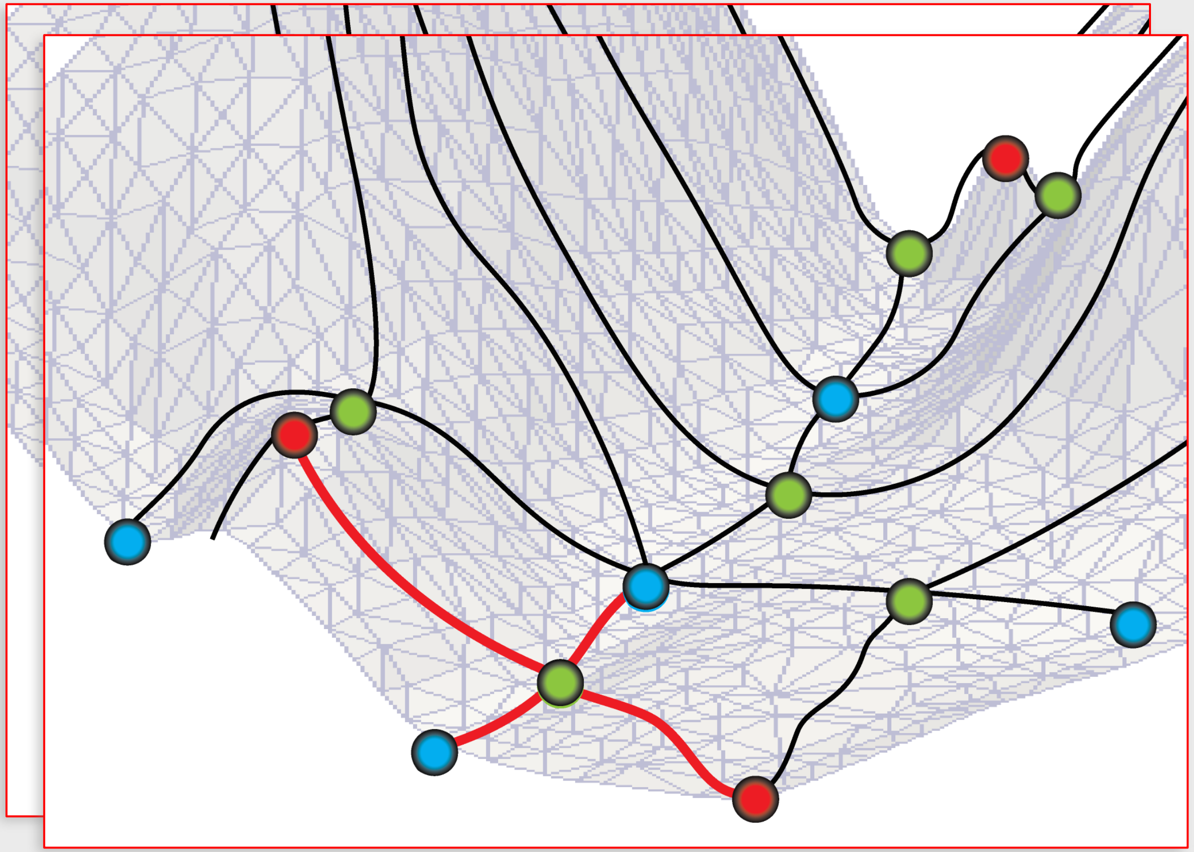
Topological Simplification

- Simplifications organized in a sequence:
  - importance value assigned to each simplification [Edelsbrunner et al., 2002]
- From a sequence we can build progressive models



Topological Simplification

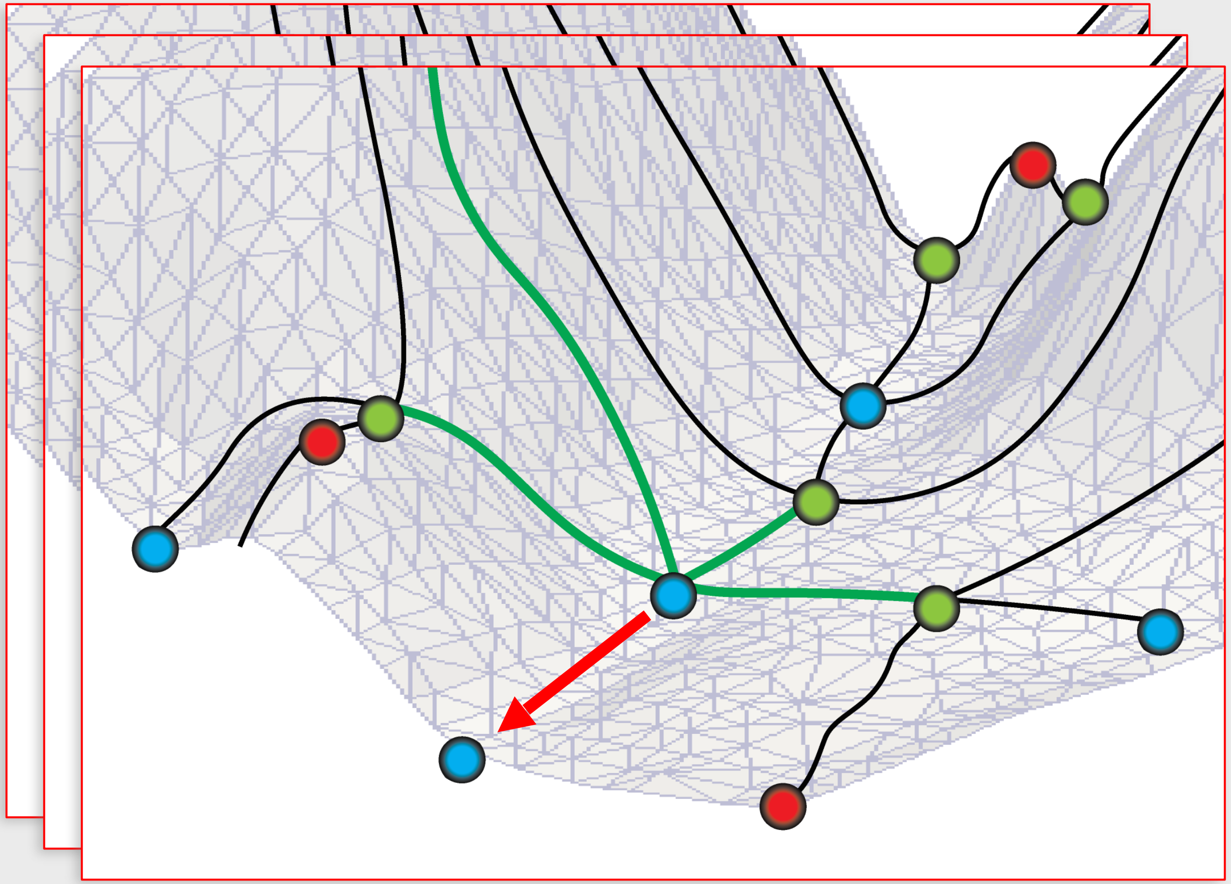
- Simplifications organized in a sequence:
  - importance value assigned to each simplification [Edelsbrunner et al., 2002]
- From a sequence we can build progressive models



Topological Simplification

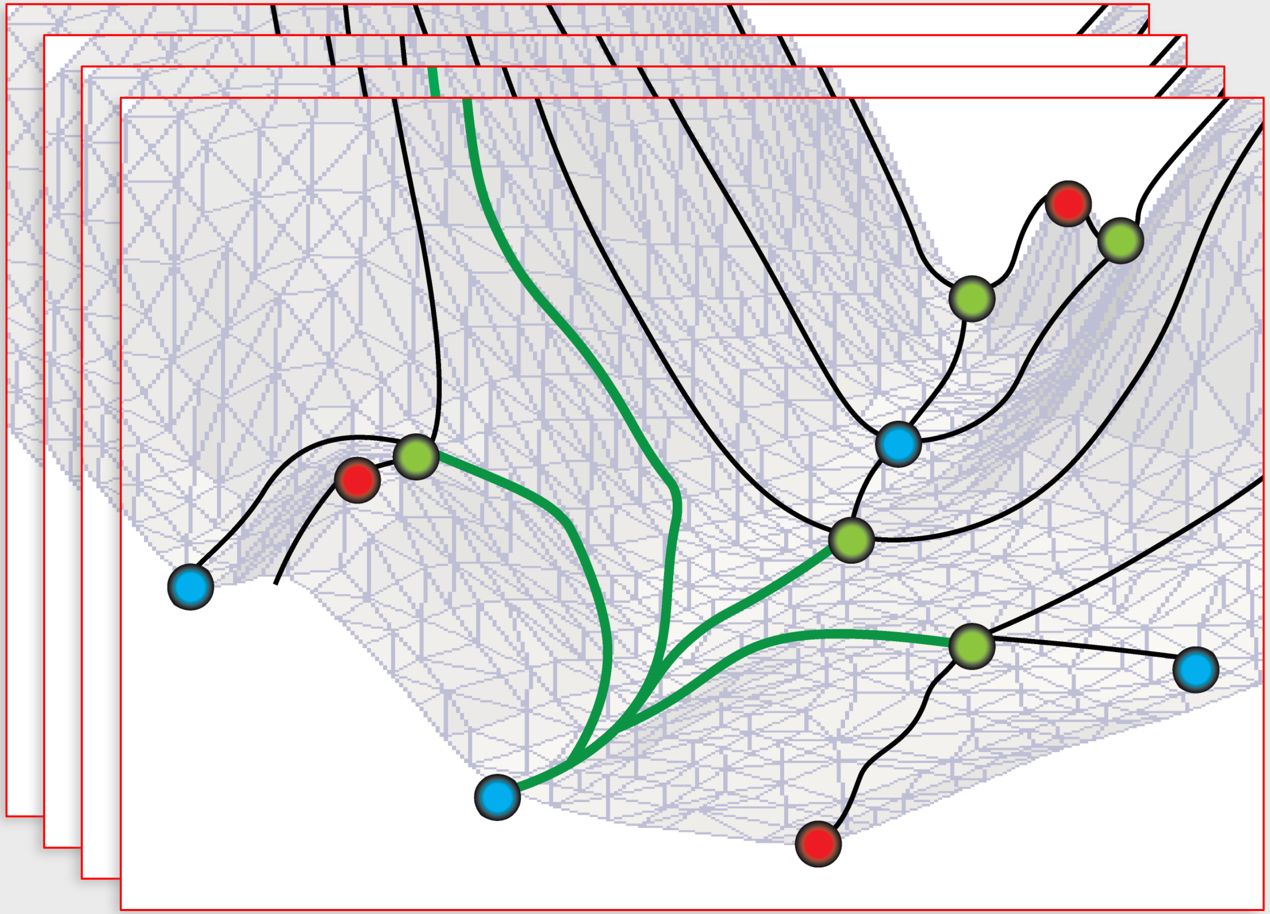
- Simplifications organized in a sequence:
    - importance value assigned to each simplification [Edelsbrunner et al., 2002]

- From a sequence we can build progressive models



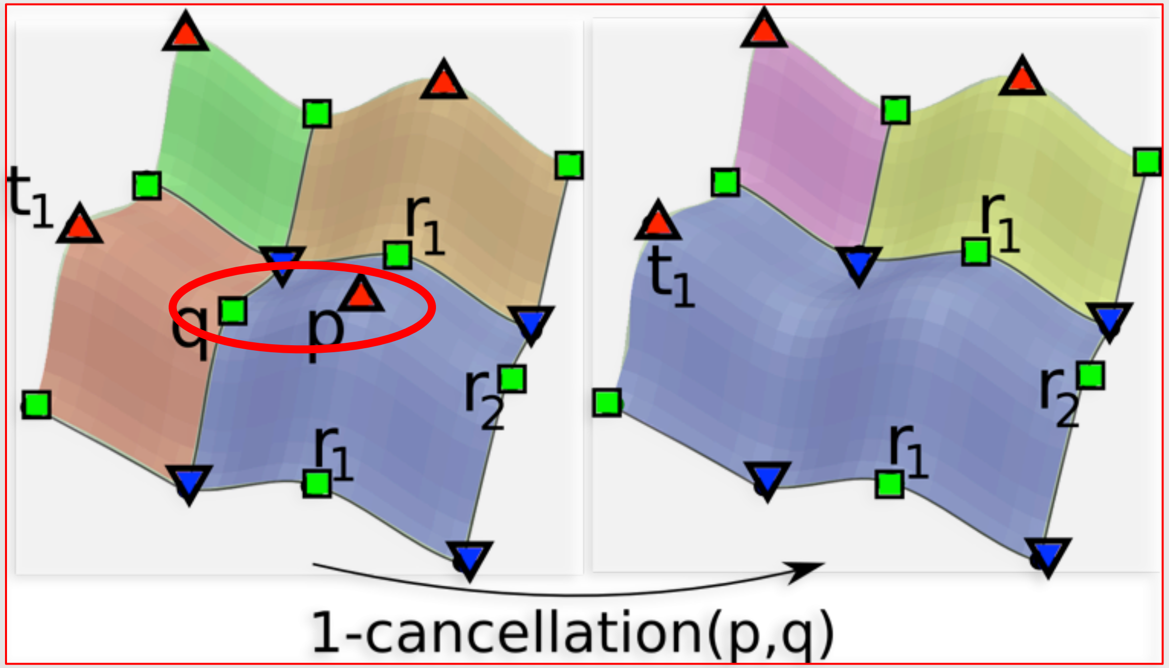
Topological Simplification

- Simplifications organized in a sequence:
  - importance value assigned to each simplification [Edelsbrunner et al., 2002]
- From a sequence we can build progressive models

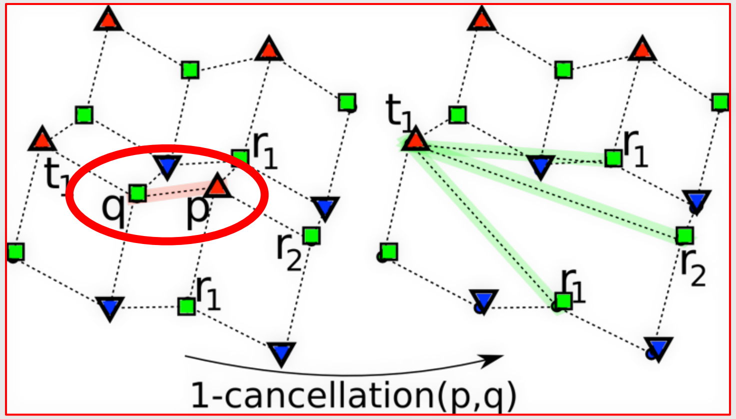


Simplifying Morse complexes

- Simplification operator defined in Morse theory: cancellation *[Milnor, 1963]*
  - removes a pair of critical points connected through a unique integral line



On the descending Morse complex



On the 1-skeleton of the Morse-Smale complex

Cancellation of a maximum *p* and a saddle point *q*

Simplification in 2D

- Based on:
  - Persistence [Edelsbrunner et al, 2002]
    - Absolute difference of two critical points scalar values *[Bremer et al., 2004] [Comic et al., 2013] [Fellegara et al., 2014]*
  - Separatrix persistence *[Gunther et al. 2009]*
    - Computed on the separatrix line between two critical points
  - Topological saliency *[Doraiswamy* et al., 2013*]*
    - Computed based on the two critical points and the critical points in the neighborhood

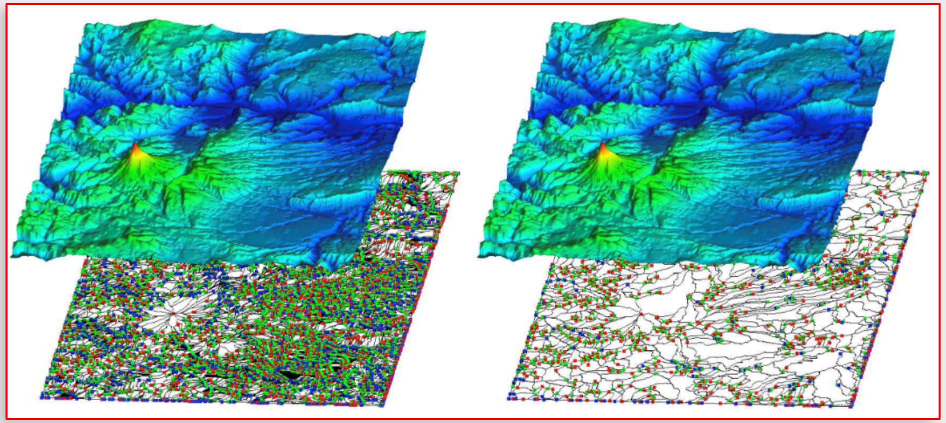
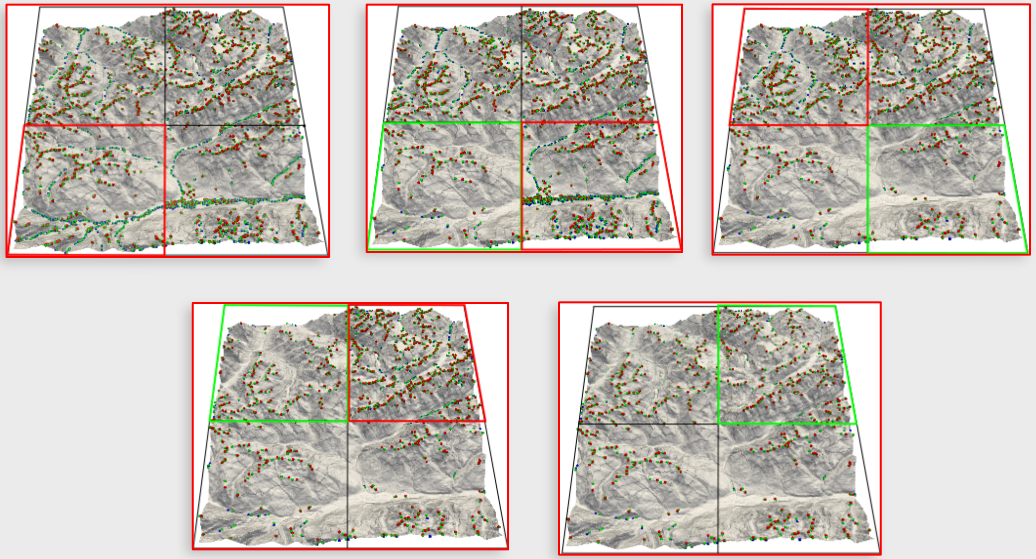


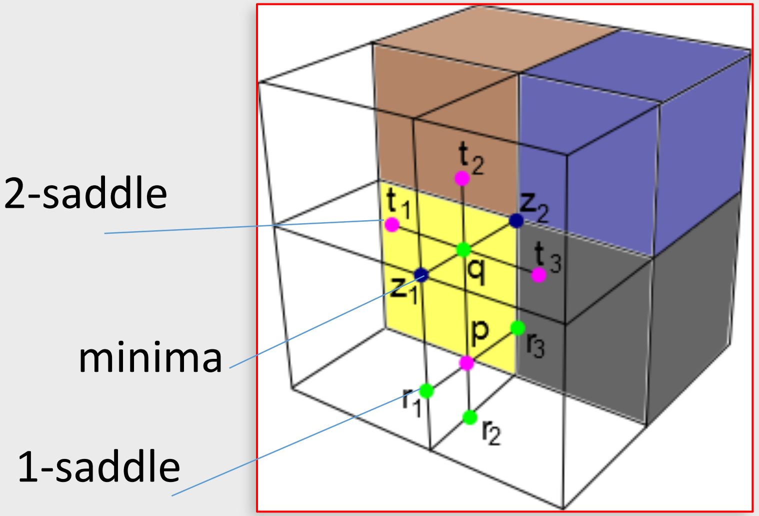
Image from [Bremer et al., 2004]



Images from [Fellegara et al., 2014]

Simplifying Morse complexes in higher dimensions

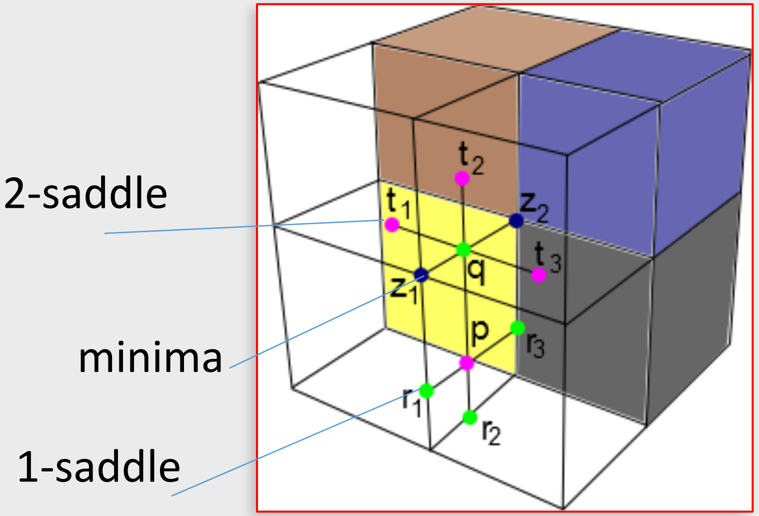
- In 2D every saddle as a regular connectivity
  - Each saddle is connected to at most two maxima and two minima
- In 3D: no restriction for connections between 1-saddles and 2-saddles
- Given a cancellation involving a 1-saddle q and 2-saddle p
  - let m = separatrix lines of p
  - let k = separatrix lines of q
  - Cancellation deletes *m+k+1* arcs and inserts *m*k* arcs *[Čomič and De Floriani, 2011]*

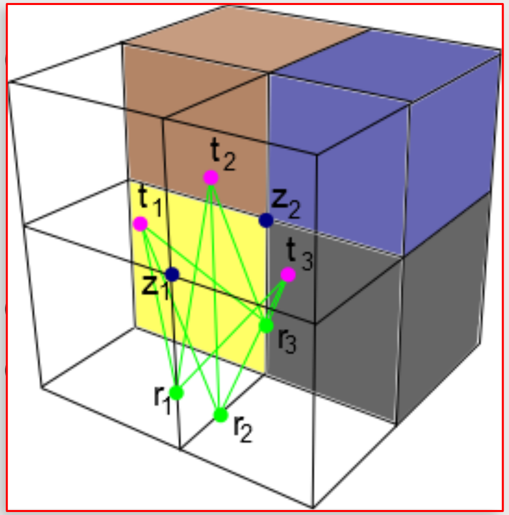


Combinatorial representation of the Morse-Smale complex. Each arc represents a 1-cell of the MS complex connecting two critical points

Simplifying Morse complexes in higher dimensions

- In 2D every saddle as a regular connectivity
  - Each saddle is connected to at most two maxima and two minima
- In 3D: no restriction for connections between 1-saddles and 2-saddles
- Given a cancellation involving a 1-saddle q and 2-saddle p
  - let m = separatrix lines of p
  - let k = separatrix lines of q
  - Cancellation deletes *m+k+1* arcs and inserts *m*k* arcs *[Čomič and De Floriani, 2011]*





Simplification operators

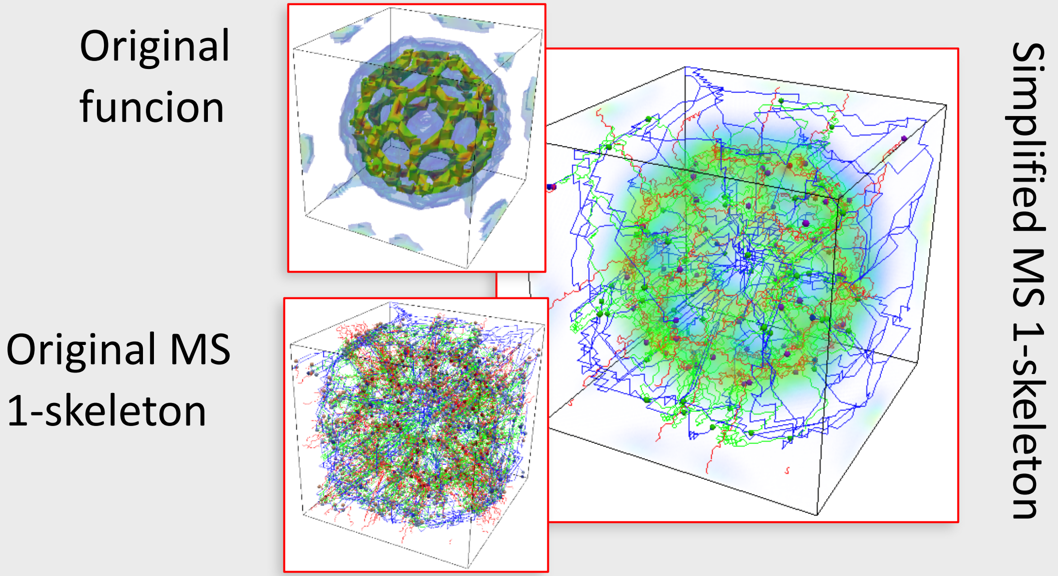
- Different simplification strategies based on cancellation have been studied for effectively simplifying a Morse-Smale complex *[Gyulassy et al., 2011]*
  - Perform all the maxima-2-saddle and minima-1-saddle firsts
  - Postpone cancellations introducing too many cells

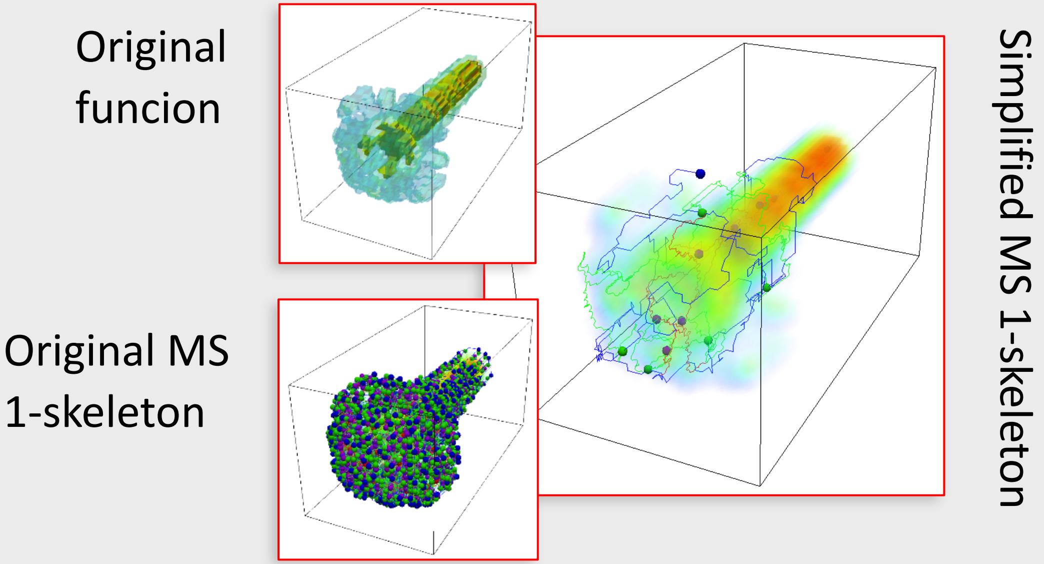
- Dimension-independent simplification operators, called ***remove*** *[Čomič and De Floriani, 2011]*
  - Deletes an *i-saddle q* and an *(i+1)-saddle p* connected to *q* only iff exactly one *(i+1)-saddle p*' is connected to *q* or exactly one *i-saddle p'* is connected to *p*
  - Can be seen as a special case of cancellation
- Remove operators form minimally complete basis of operators for simplifying Morse-Smale complexes

Simplification in 3D

- All the simplification algorithm defined for volumetric data are based on persistence *[Gyulassy et al., 2006] [Comic et al., 2013]*

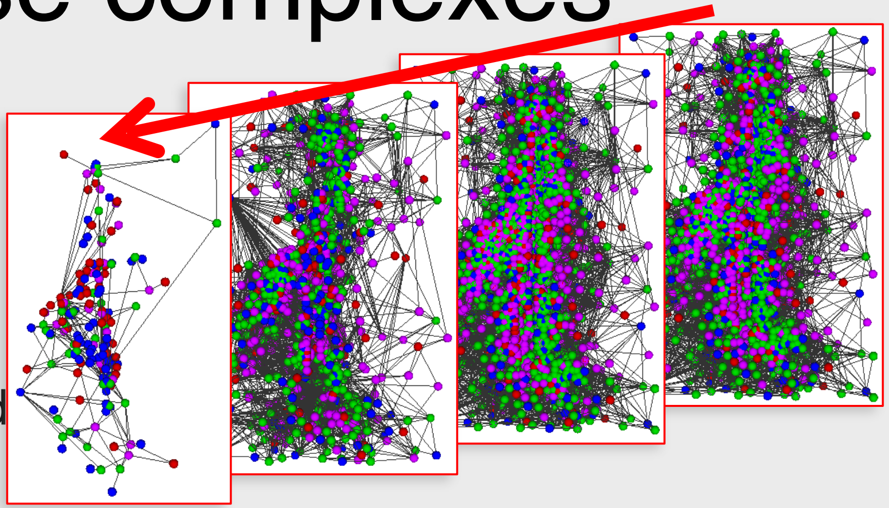
- Using remove operators results in 20% more compact Morse-Smale complexes in about half the time *[Comic et al., 2013]*



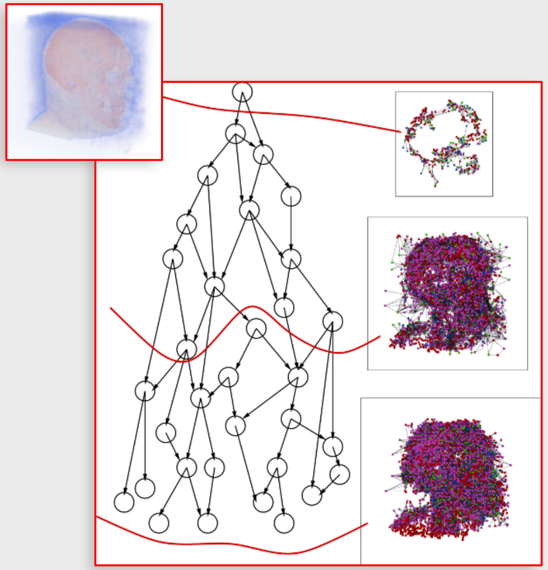


Multi-resolution models for Morse complexes

- Generated through a sequence of cancellations (or remove) applied to the original Morse or Morse-Smale complex



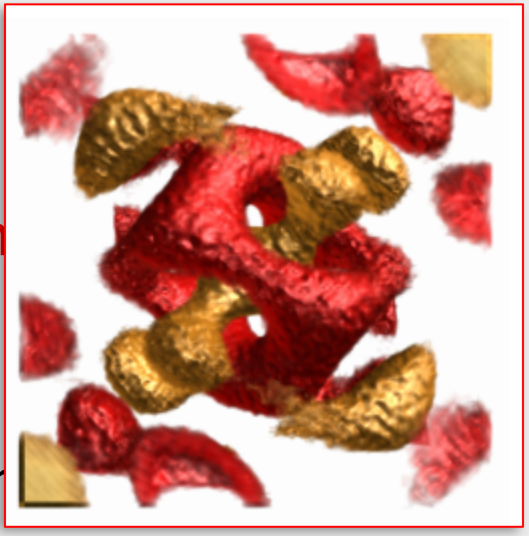
- Multi-resolution model:
  - A collection of refinements reversing the cancellations performed in simplification
  - A direct dependency relation between pairs of refinements
- Combinatorial representation of a family of Morse or Morse-Smale complexes



- Multi-resolution models for terrain data *[Edelsbrunner et al., 2001; Bremer et al., 2005; Danovaro et al., 2007]*
- Multi-resolution models for volumetric data *[Gyulassy et al., 2012; Comic et al., 2012]*

Modifying the scalar function

Images from *[Gunther et al., 2014]*

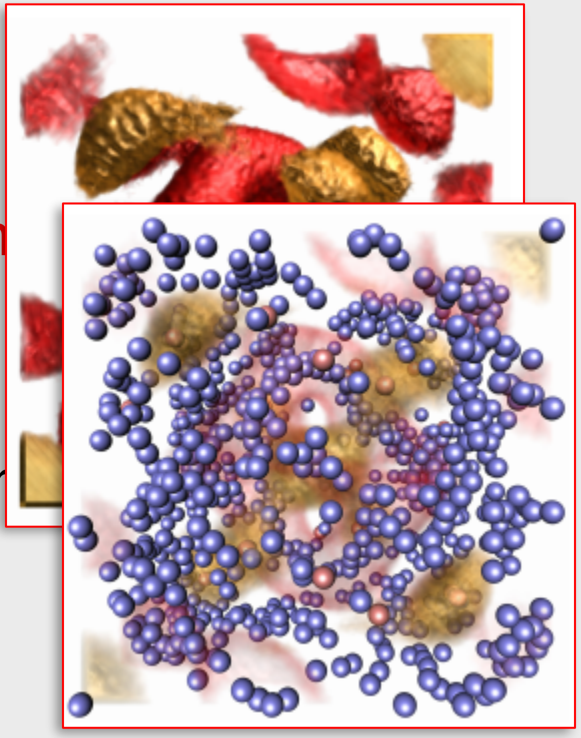


- Algorithms have been defined for modifying the underlying scalar function while modifying the topological representation

- *For terrains defined on regular grids*
  - *[Bremer et al., 2004]* function modified using Laplacian smoothing after each cancellation
  - *[Weinkauf et al., 2010]* function modified at the end of the sequence of cancellations to improve performances
  - *[Allemand et al., 2015]* function modified using piecewise-polynomial lines and surfaces.
- *For volumetric data defined on cubical grids*
  - *[Gunther et al., 2014]*

Modifying the scalar function

Images from *[Gunther et al., 2014]*

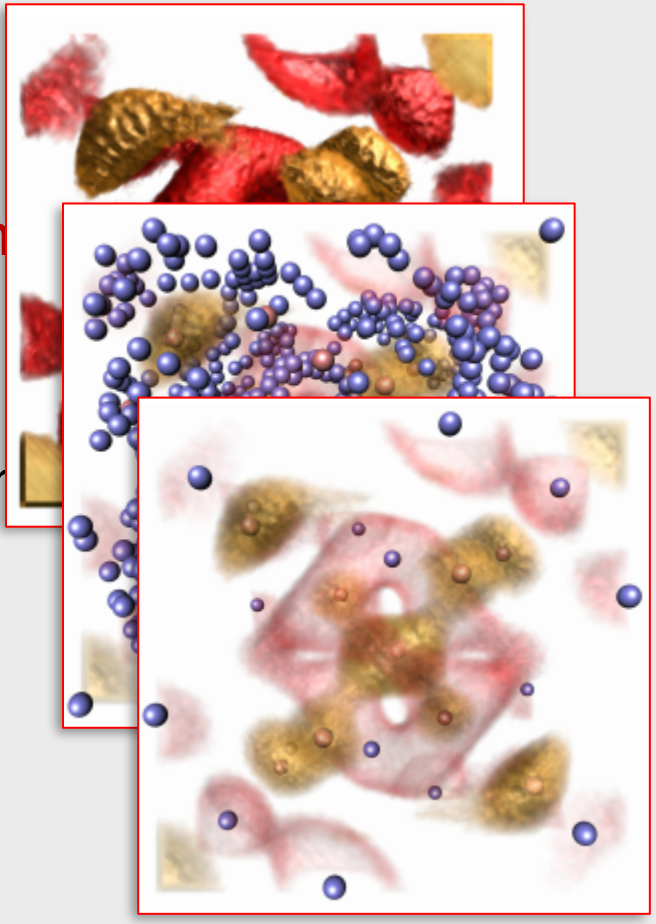


- Algorithms have been defined for modifying the underlying scalar function while modifying the topological representation

- *For terrains defined on regular grids*
  - *[Bremer et al., 2004]* function modified using Laplacian smoothing after each cancellation
  - *[Weinkauf et al., 2010]* function modified at the end of the sequence of cancellations to improve performances
  - *[Allemand et al., 2015]* function modified using piecewise-polynomial lines and surfaces.
- *For volumetric data defined on cubical grids*
  - *[Gunther et al., 2014]*

Modifying the scalar function

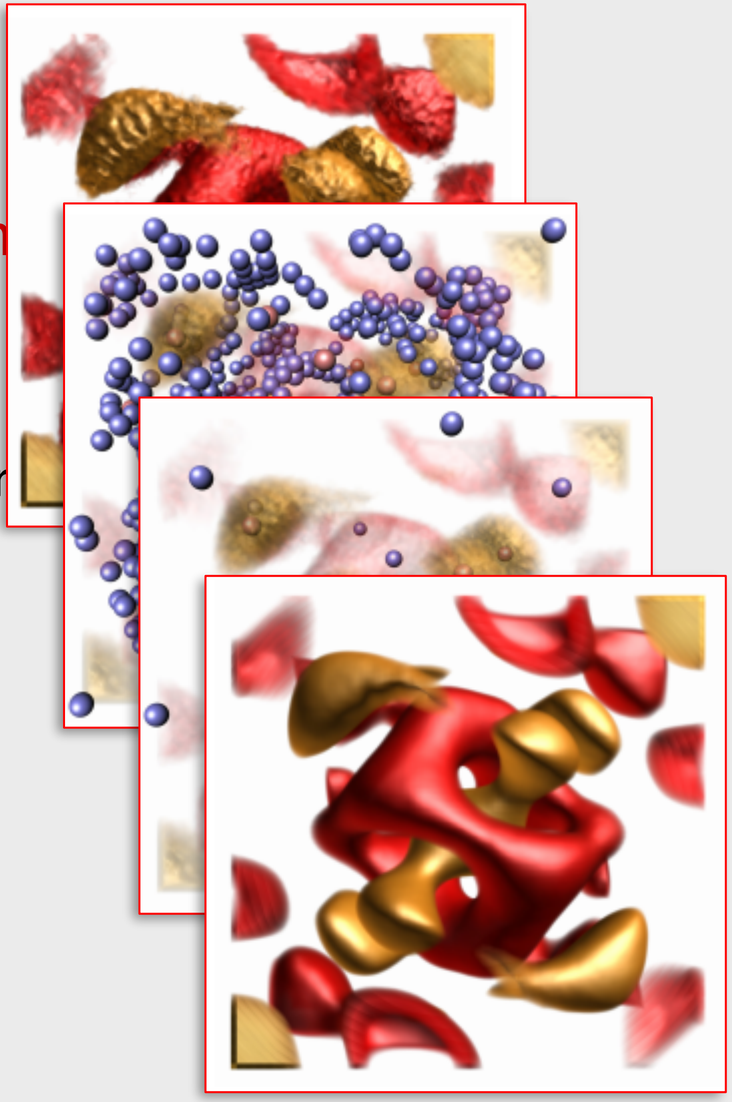
Images from *[Gunther et al., 2014]*



- Algorithms have been defined for modifying the underlying scalar function while modifying the topological representation
- *For terrains defined on regular grids*
  - *[Bremer et al., 2004]* function modified using Laplacian smoothing after each cancellation
  - *[Weinkauf et al., 2010]* function modified at the end of the sequence of cancellations to improve performances
  - *[Allemand et al., 2015]* function modified using piecewise-polynomial lines and surfaces.
- *For volumetric data defined on cubical grids*
  - *[Gunther et al., 2014]*

Modifying the scalar function

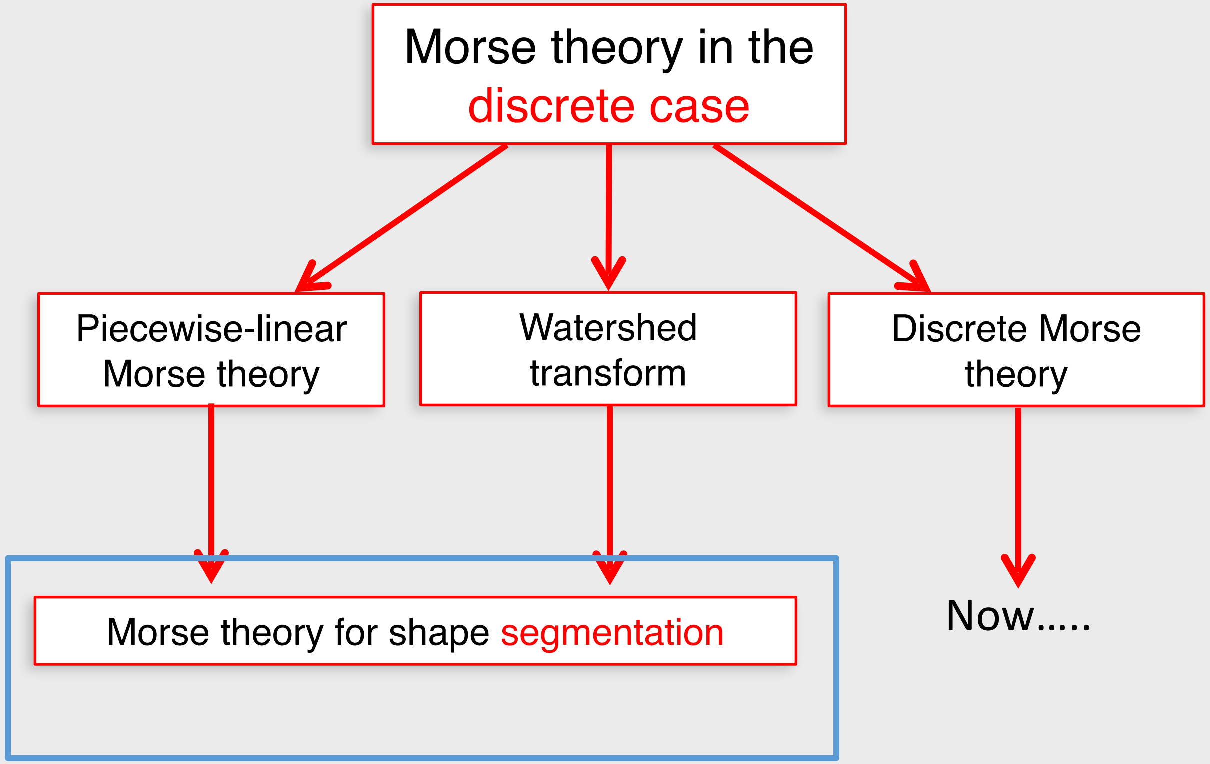
Images from *[Gunther et al., 2014]*



- Algorithms have been defined for modifying the underlying scalar function while modifying the topological representation

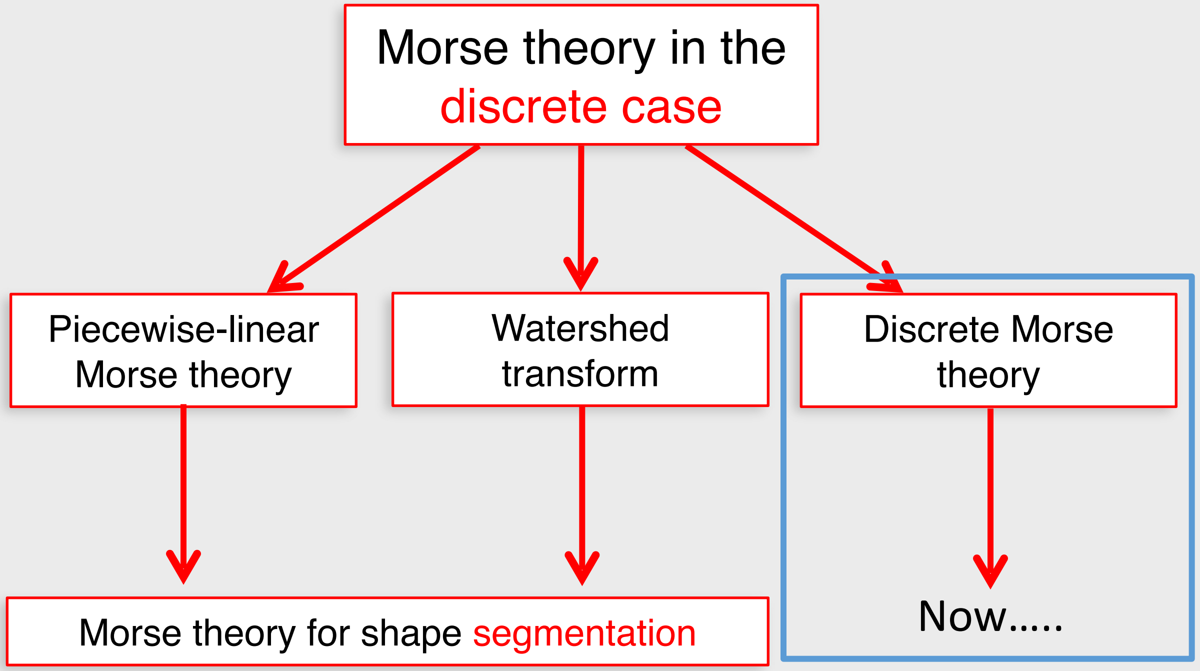
- *For terrains defined on regular grids*
  - *[Bremer et al., 2004]* function modified using Laplacian smoothing after each cancellation
  - *[Weinkauf et al., 2010]* function modified at the end of the sequence of cancellations to improve performances
  - *[Allemand et al., 2015]* function modified using piecewise-polynomial lines and surfaces.
- *For volumetric data defined on cubical grids*
  - *[Gunther et al., 2014]*

Morse theory in the discrete case



- Piecewise-linear Morse theory *[T. Banchoff 1967, 1970]*
  - For polyhedral surfaces
  - Defined for the 2D case and extended to 3D
- Watershed transform *[F. Meyer 1994]*
  - For cell complexes
  - Dimension-independent
- Discrete Morse theory *[R. Forman 1998, 2002]*
  - For cell complexes
  - Dimensions-independent

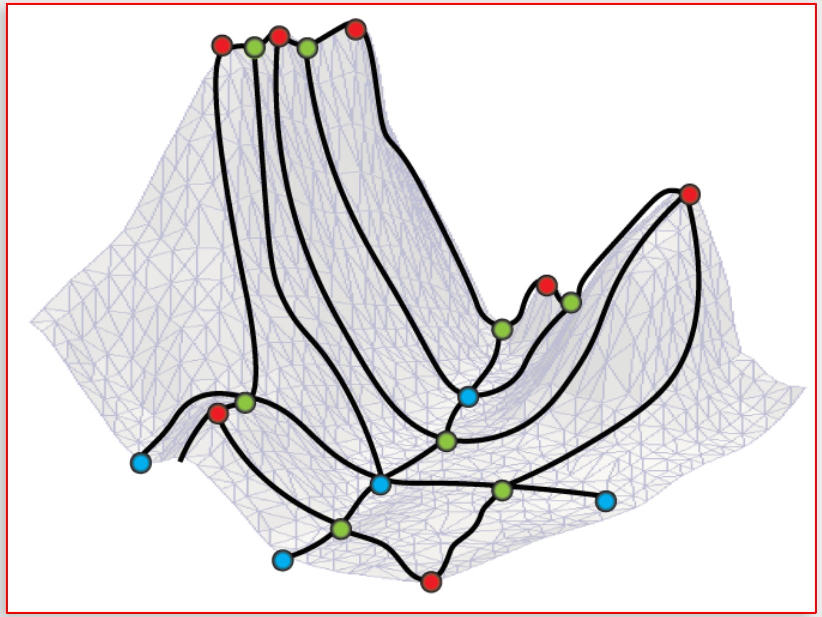
Morse theory in the discrete case



- Piecewise-linear Morse theory *[T. Banchoff 1967, 1970]*
  - For polyhedral surfaces
  - Defined for the 2D case and extended to 3D
- Watershed transform *[F. Meyer 1994]*
  - For cell complexes
  - Dimension-independent
- Discrete Morse theory *[R. Forman 1998, 2002]*
  - For cell complexes
  - Dimensions-independent

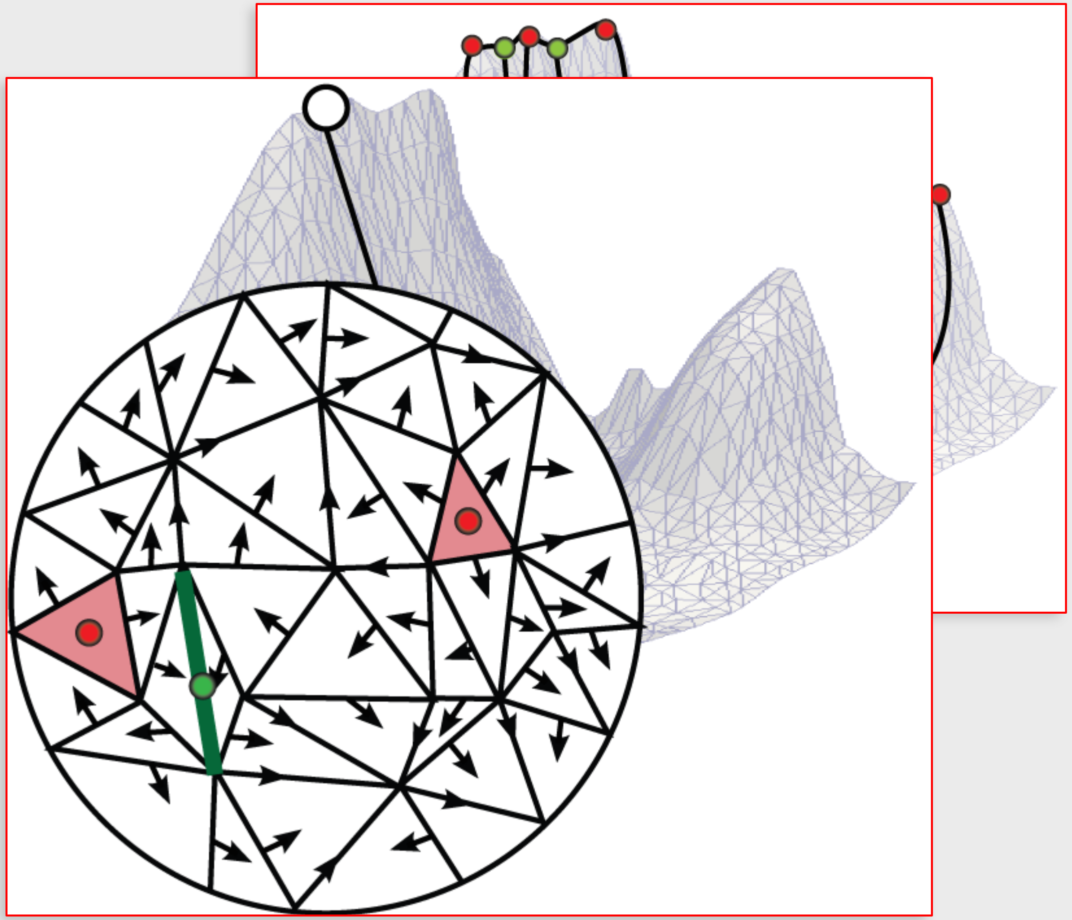
Discrete Morse Theory [Forman 1998]

- Combinatorial counterpart of Morse theory
  - Introduced for cell complexes
  - Gives a compact homology-equivalent model for a shape
  - Derivative free tool for computing segmentations of shapes



Discrete Morse Theory [Forman 1998]

- Combinatorial counterpart of Morse theory
  - Introduced for cell complexes
  - Gives a compact homology-equivalent model for a shape
  - Derivative free tool for computing segmentations of shapes



Discrete Morse theory

Let Σ be a simplicial complex

•Function F: Σ R, defined on every simplex σ of Σ

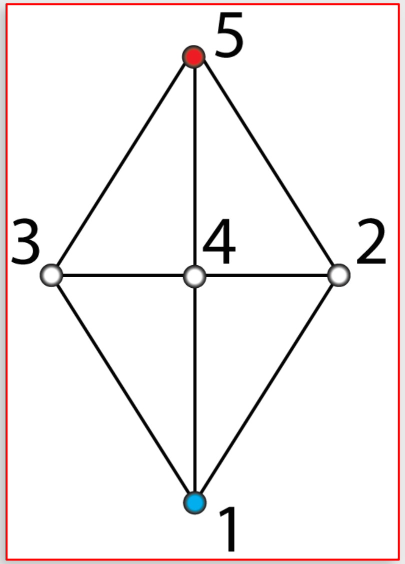
Notions introduced:

•F is a discrete Morse function if for every i-simplex

$$\#\{\tau \in cb(\sigma) \mid F(\tau) \leq F(\sigma)\} \leq 1 \quad \text{AND} \quad \#\{\tau \in b(\sigma) \mid F(\tau) \geq F(\sigma)\} \leq 1$$

•The two conditions are exclusive and induce a pairings on the simplexes of Σ.

- A pair (τ, σ) can be viewed as an arrow formed by an head i-simplex σ and a tail (i-1)-simplex τ.



Discrete Morse theory

Let Σ be a simplicial complex

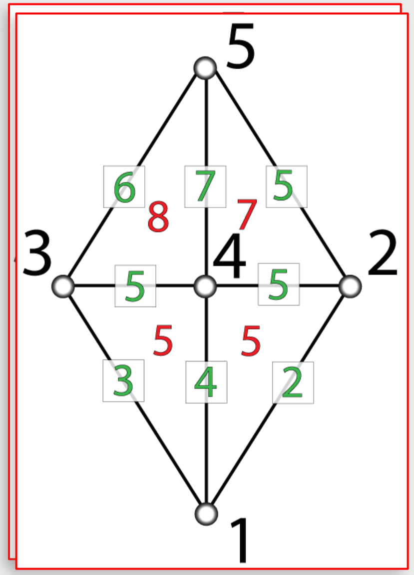
- Function F: Σ → R, defined on every simplex σ of Σ

Notions introduced:

- F is a discrete Morse function if for every i-simplex

$$\#\{\tau \in cb(\sigma) \mid F(\tau) \leq F(\sigma)\} \leq 1 \quad \text{AND} \quad \#\{\tau \in b(\sigma) \mid F(\tau) \geq F(\sigma)\} \leq 1$$

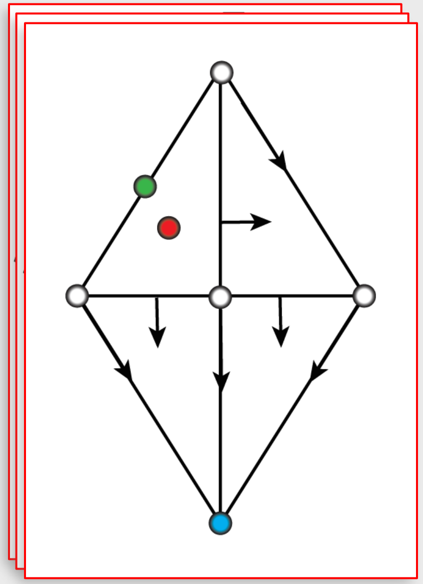
- The two conditions are exclusive and induce a pairings on the simplexes of Σ.
    - A pair (τ, σ) can be viewed as an arrow formed by an head i-simplex σ and a tail (i-1)-simplex τ.



Discrete Morse theory

Let Σ be a simplicial complex

•Function F: Σ → R, defined on every simplex σ of Σ



Notions introduced:

•F is a discrete Morse function if for every i-simplex

$$\#\{\tau \in cb(\sigma) \mid F(\tau) \le F(\sigma)\} \le 1 \quad \text{AND} \quad \#\{\tau \in b(\sigma) \mid F(\tau) \ge F(\sigma)\} \le 1$$

•The two conditions are exclusive and induce a pairings on the simplexes of Σ.

- A pair (τ, σ) can be viewed as an arrow formed by an head i-simplex σ and a tail (i-1)-simplex τ.

Discrete Morse theory

Let Σ be a simplicial complex

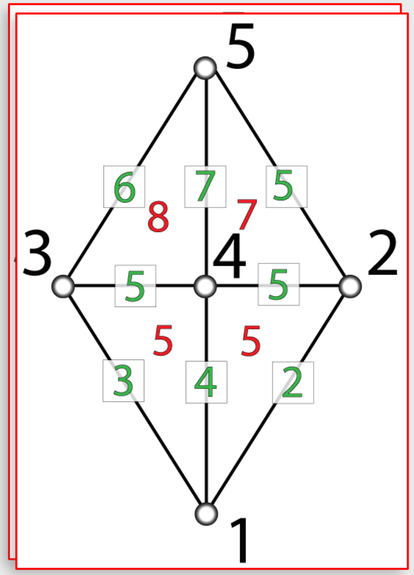
- Function F: Σ R, defined on every simplex σ of Σ

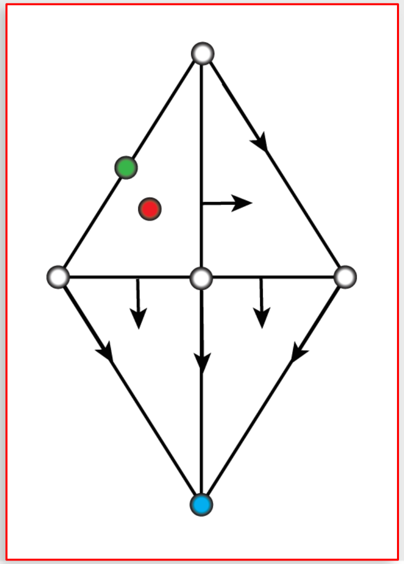
Notions introduced:

- F is a discrete Morse function if for every i-simplex

$$\#\{\tau \in cb(\sigma) \mid F(\tau) \leq F(\sigma)\} \leq 1 \quad \text{AND} \quad \#\{\tau \in b(\sigma) \mid F(\tau) \geq F(\sigma)\} \leq 1$$

- The two conditions are exclusive and induce a pairings on the simplexes of Σ.
  - A pair (τ, σ) can be viewed as an arrow formed by an head i-simplex σ and a tail (i-1)-simplex τ.





Discrete Morse theory

Let Σ be a simplicial complex

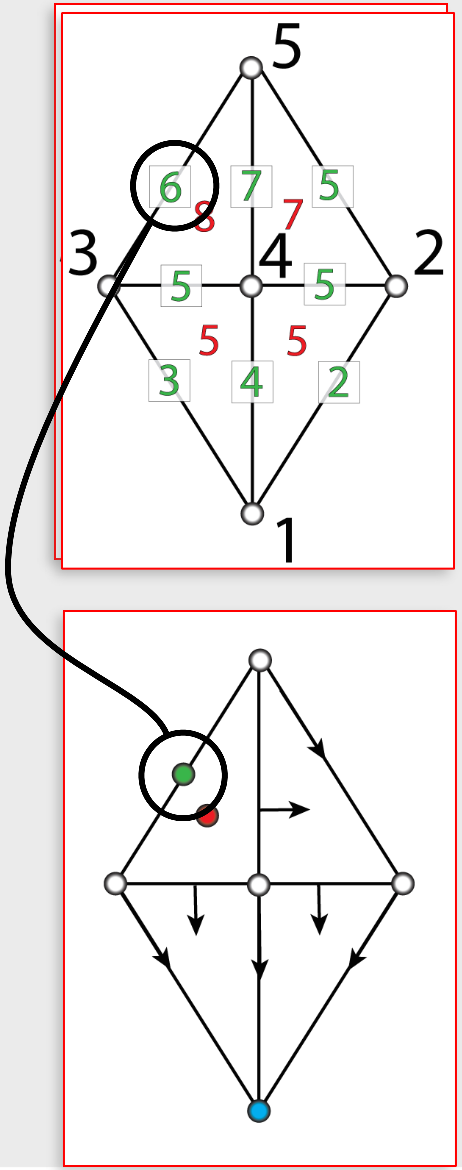
- Function $F: \Sigma \rightarrow \mathbb{R}$, defined on every simplex σ of Σ

Notions introduced:

- F is a discrete Morse function if for every i-simplex

$$\#\{\tau \in cb(\sigma) \mid F(\tau) \leq F(\sigma)\} \leq 1 \quad \text{AND} \quad \#\{\tau \in b(\sigma) \mid F(\tau) \geq F(\sigma)\} \leq 1$$

- The two conditions are exclusive and induce a pairings on the simplexes of Σ.
  - A pair (τ, σ) can be viewed as an arrow formed by an head i-simplex σ and a tail (i-1)-simplex τ.

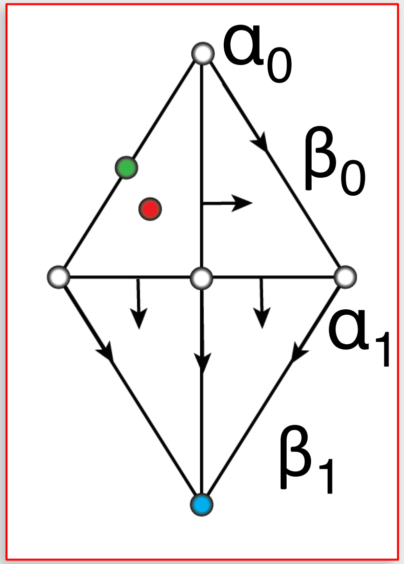


Discrete Morse theory

- A discrete vector field V on Σ is a collection of pairs (τ, σ) ∊ Σ x Σ such that τ < σ and each simplex of Σ is in at most one pair of V

- Given a discrete vector field V, a *V-path* is a sequence of pairs of V

$$\alpha_0, \beta_0, \alpha_1, \beta_1, \alpha_2, \ldots, \alpha_{r-1}, \beta_{r-1}, \alpha_r$$



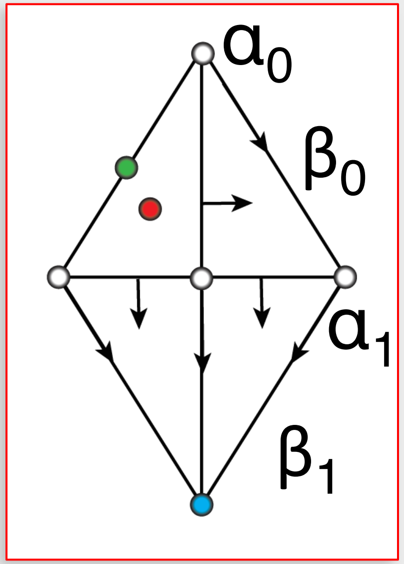
Discrete Morse theory

- A discrete vector field V on Σ is a collection of pairs (τ, σ) ∈ Σ x Σ such that τ < σ and each simplex of Σ is in at most one pair of V

- Given a discrete vector field V, a *V-path* is a sequence of pairs of V

$$\alpha_0, \beta_0, \alpha_1, \beta_1, \alpha_2, \ldots, \alpha_{r-1}, \beta_{r-1}, \alpha_r$$

Gradient pair



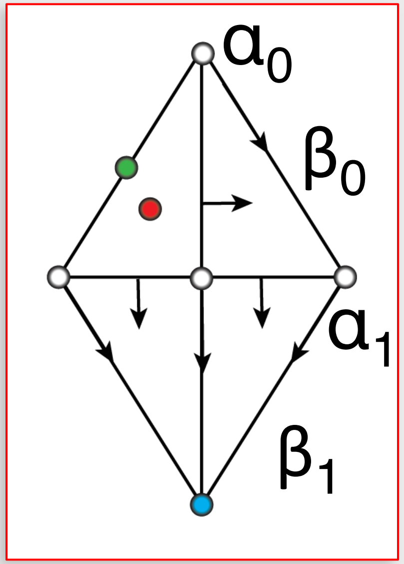
Discrete Morse theory

- A discrete vector field V on Σ is a collection of pairs (τ, σ) ∊ Σ x Σ such that τ < σ and each simplex of Σ is in at most one pair of V

- Given a discrete vector field V, a *V-path* is a sequence of pairs of V

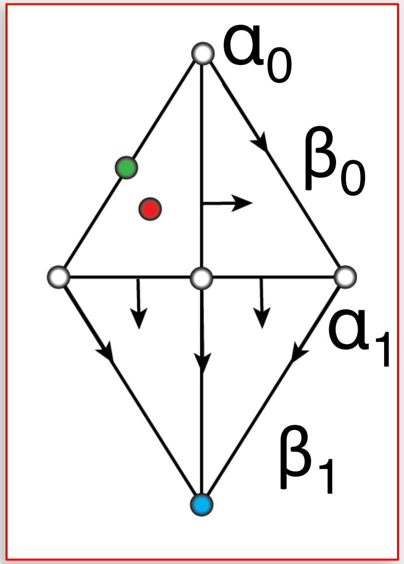
$$\alpha_0, \beta_0, \alpha_1, \beta_1, \alpha_2, \dots, \alpha_{r-1}, \beta_{r-1}, \alpha_r$$

Gradient pair



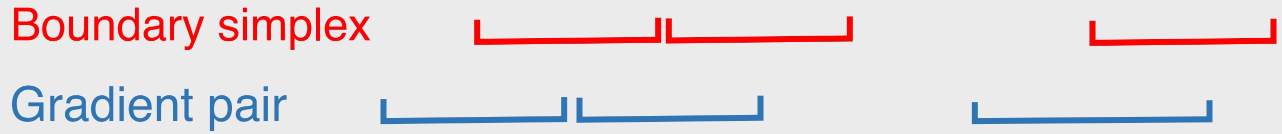
Discrete Morse theory

- A discrete vector field V on Σ is a collection of pairs (τ, σ) ∈ Σ x Σ such that τ < σ and each simplex of Σ is in at most one pair of V



- Given a discrete vector field V, a *V-path* is a sequence of pairs of V

$$\alpha_0, \beta_0, \alpha_1, \beta_1, \alpha_2, \ldots, \alpha_{r-1}, \beta_{r-1}, \alpha_r$$



Discrete Morse theory

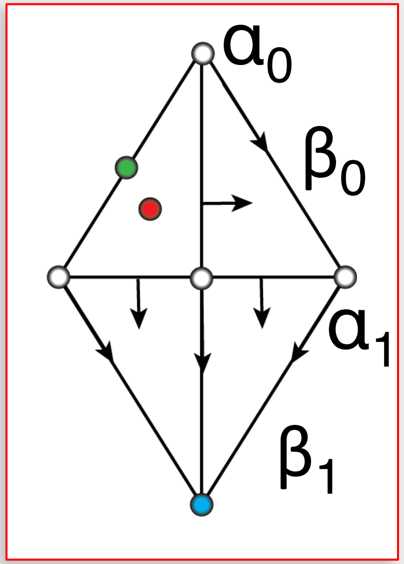
- A discrete vector field V on Σ is a collection of pairs (τ, σ) ∈ Σ x Σ such that τ < σ and each simplex of Σ is in at most one pair of V

- Given a discrete vector field V, a *V-path* is a sequence of pairs of V

$$\alpha_0, \beta_0, \alpha_1, \beta_1, \alpha_2, \ldots, \alpha_{r-1}, \beta_{r-1}, \alpha_r$$

Boundary simplex

Gradient pair



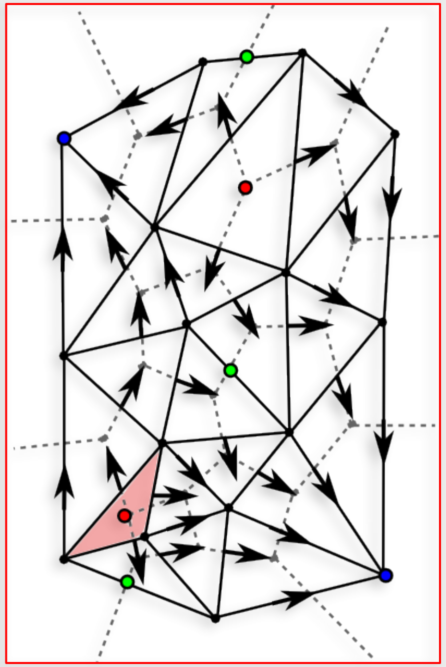
A discrete vector field V is the (Forman) gradient vector field of a discrete Morse function if and only if there are no non-trivial closed paths

Discrete Morse complex

- Navigating the V-paths, discrete Morse complexes and separatrix lines are retrieved
- For the descending Morse complex
  - From critical triangles navigating *edge-face* arrows
  - From critical edges navigating edge-vertex arrows

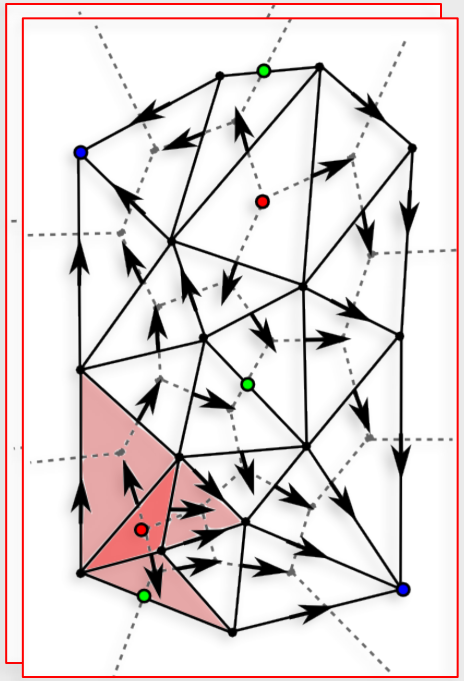
Discrete Morse complex

- Navigating the V-paths, discrete Morse complexes and separatrix lines are retrieved
- For the descending Morse complex
  - From critical triangles navigating *edge-face* arrows
  - From critical edges navigating edge-vertex arrows



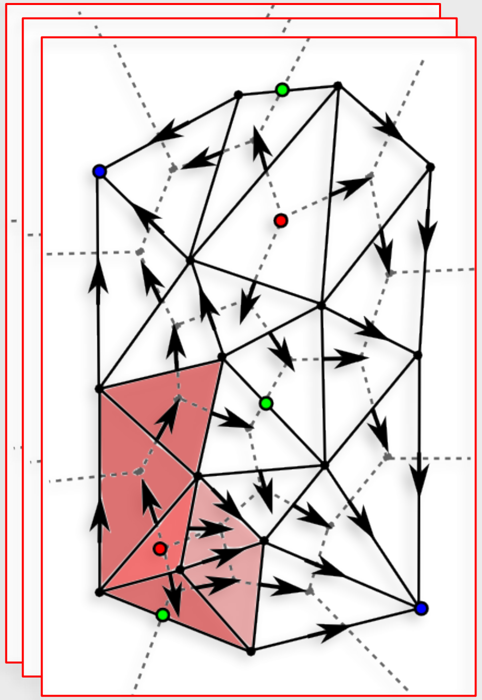
Discrete Morse complex

- Navigating the V-paths, discrete Morse complexes and separatrix lines are retrieved
- For the descending Morse complex
  - From critical triangles navigating *edge-face* arrows
  - From critical edges navigating edge-vertex arrows



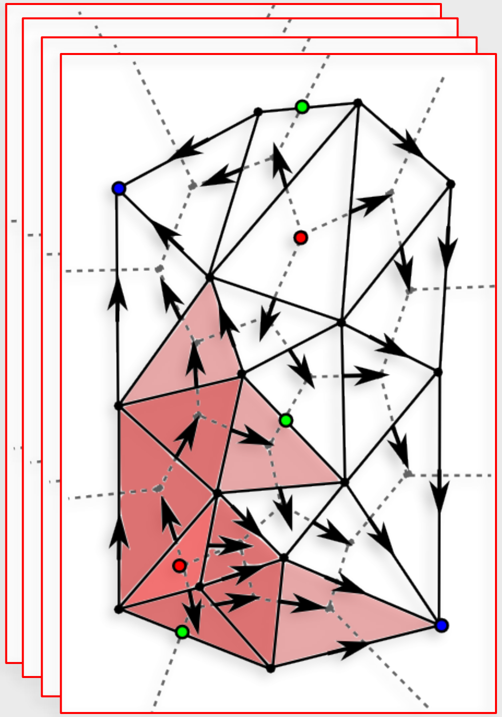
Discrete Morse complex

- Navigating the V-paths, discrete Morse complexes and separatrix lines are retrieved
- For the descending Morse complex
  - From critical triangles navigating *edge-face* arrows
  - From critical edges navigating edge-vertex arrows



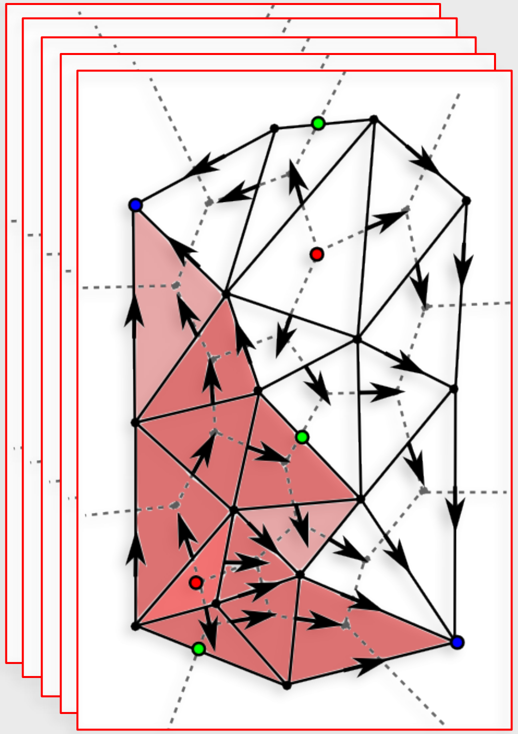
Discrete Morse complex

- Navigating the V-paths, discrete Morse complexes and separatrix lines are retrieved
- For the descending Morse complex
  - From critical triangles navigating *edge-face* arrows
  - From critical edges navigating edge-vertex arrows



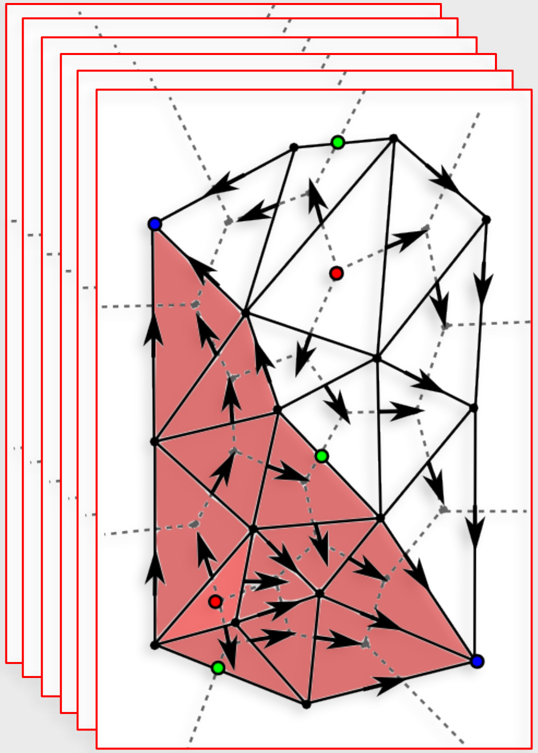
Discrete Morse complex

- Navigating the V-paths, discrete Morse complexes and separatrix lines are retrieved
- For the descending Morse complex
  - From critical triangles navigating *edge-face* arrows
  - From critical edges navigating edge-vertex arrows



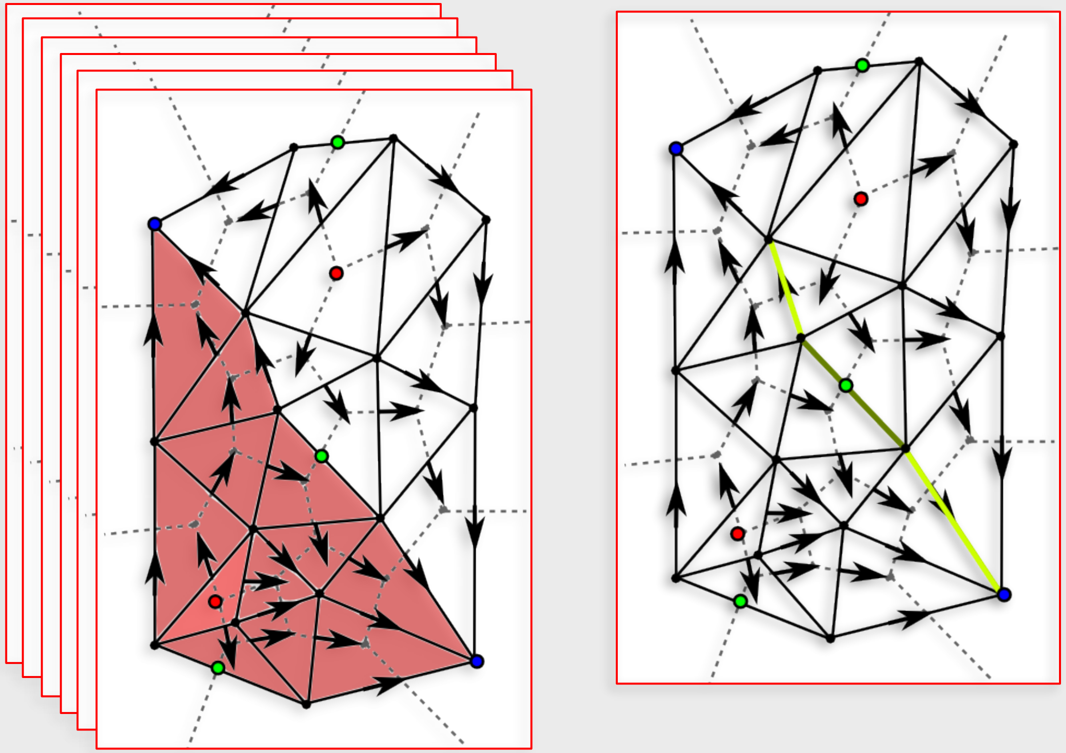
Discrete Morse complex

- Navigating the V-paths, discrete Morse complexes and separatrix lines are retrieved
- For the descending Morse complex
  - From critical triangles navigating *edge-face* arrows
  - From critical edges navigating edge-vertex arrows

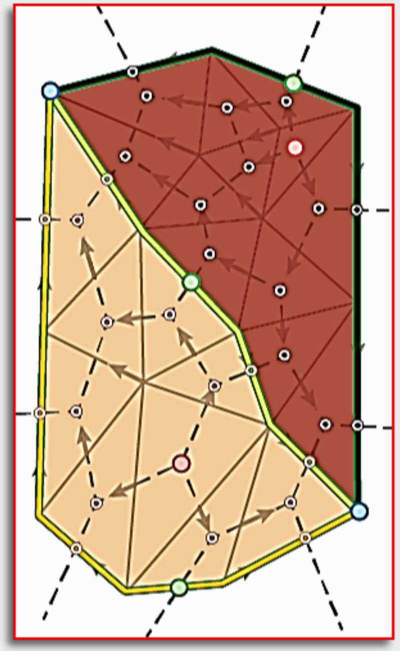


Discrete Morse complex

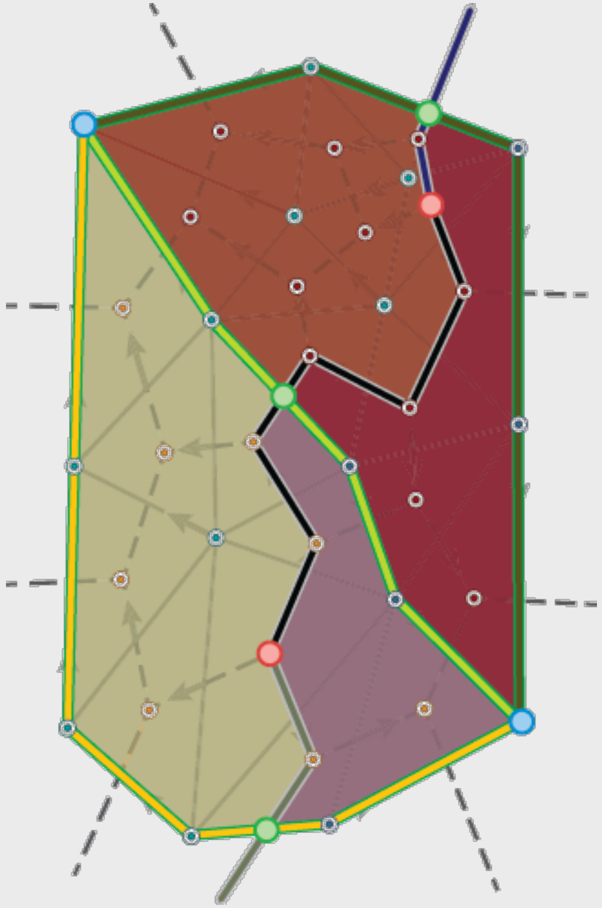
- Navigating the V-paths, discrete Morse complexes and separatrix lines are retrieved
- For the descending Morse complex
  - From critical triangles navigating *edge-face* arrows
  - From critical edges navigating edge-vertex arrows



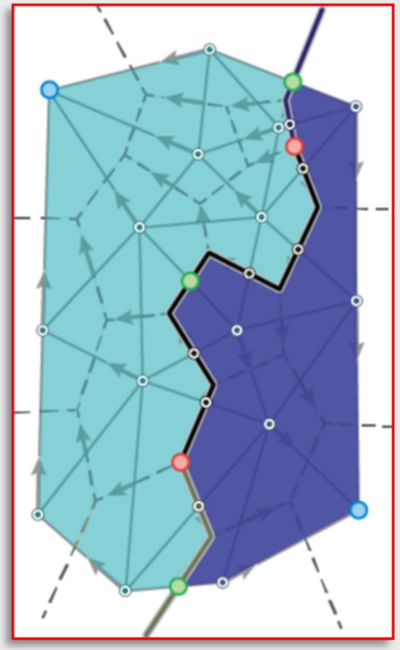
Discrete Morse complex



Descending Morse complex



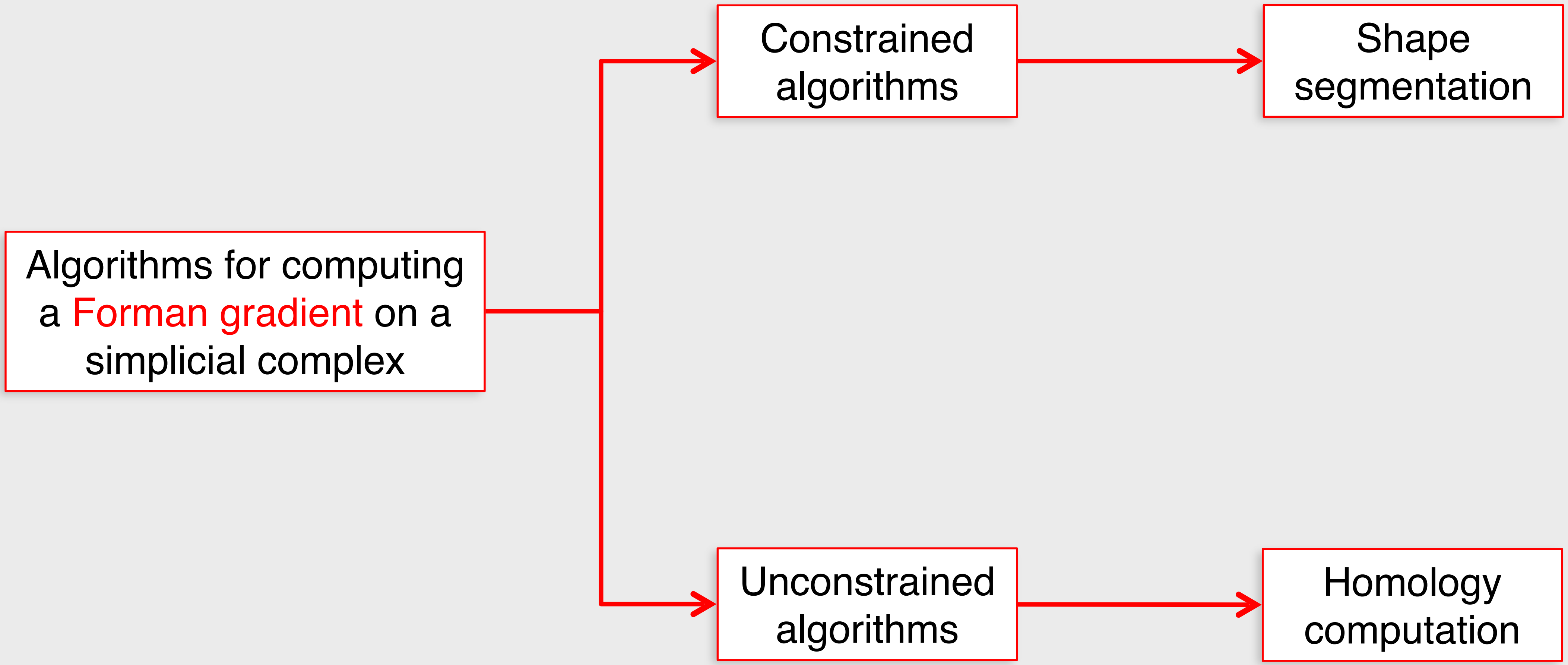
Morse-Smale complex



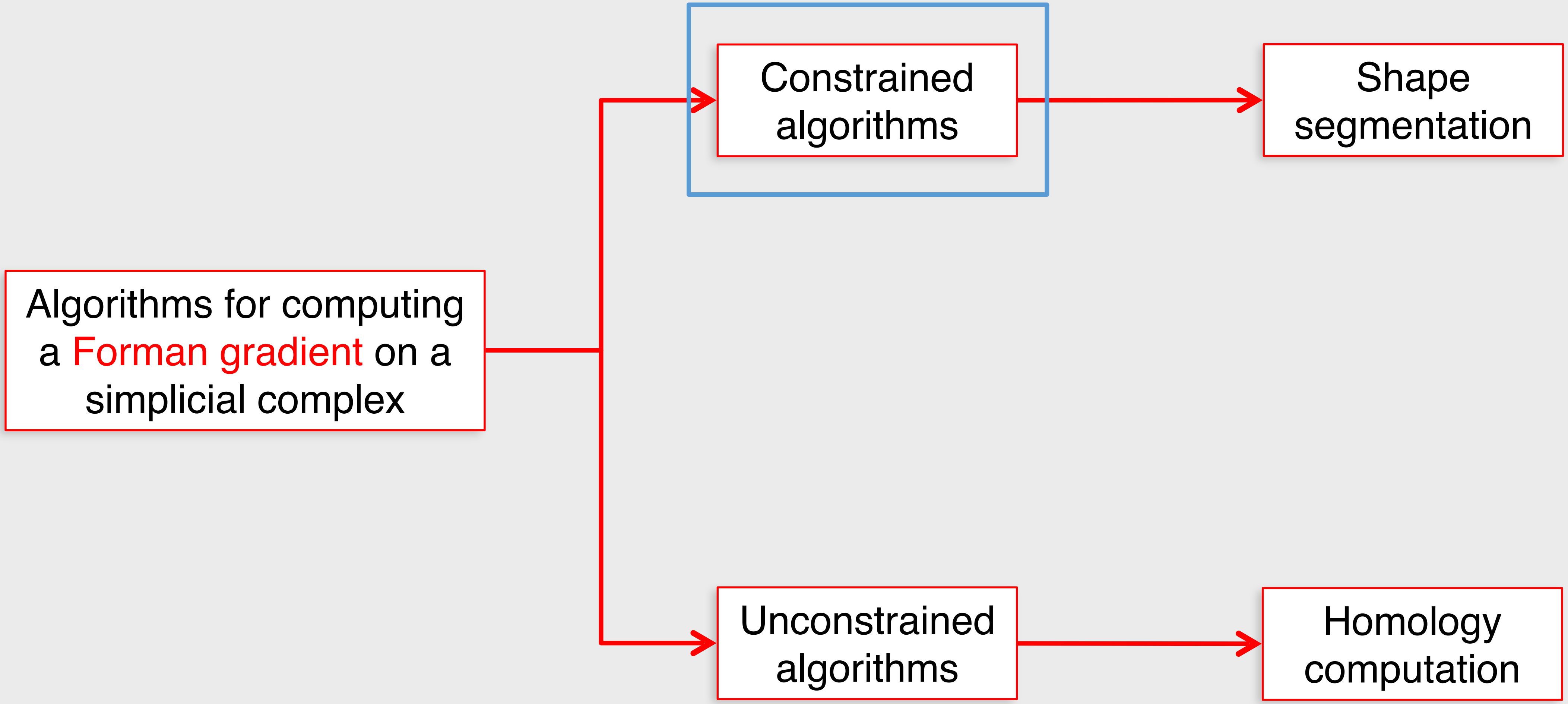
Ascending Morse complex

Images from *[Weiss et al., 2013]*

Discrete Morse theory



Discrete Morse theory



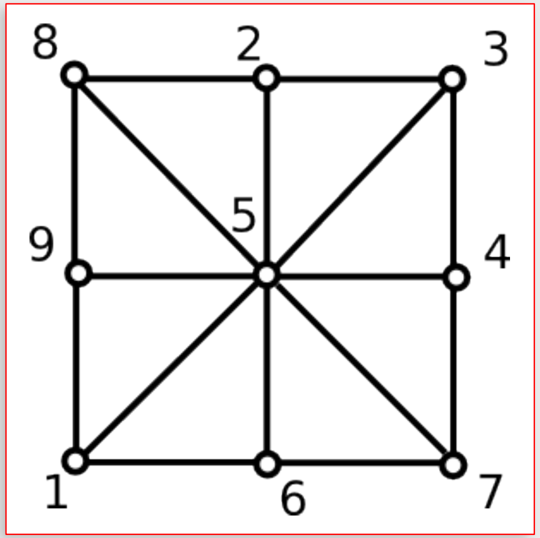
Computing a Forman gradient

- Basics:
  - Starting from the vertices, simulate a Forman function while building the discrete gradient vector field
  - Navigate the V-paths of the discrete gradient vector field starting from the critical simplexes

- Parallelize the computation:
  - Working on the link of each vertex *[King et al., 2005]*
  - Divide and conquer approach *[Gyulassy et al., 2008]*
  - Working on the star of each vertex *[Robins et al., 2011]*
- Minimize the number of critical simplexes *[Cazals et al., 2003][Robins et al., 2011]*
- Compute accurate geometry of the Morse complexes *[Gyulassy et al., 2012]*

Computing a Forman gradient

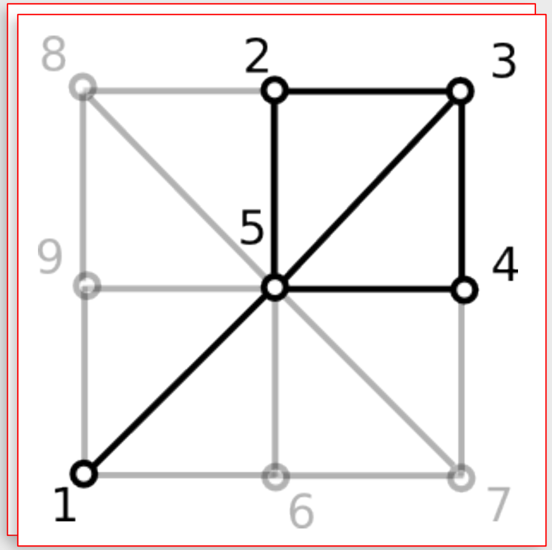
- Basics:
  - Starting from the vertices, simulate a Forman function while building the discrete gradient vector field
  - Navigate the V-paths of the discrete gradient vector field starting from the critical simplexes



- Parallelize the computation:
  - Working on the link of each vertex *[King et al., 2005]*
  - Divide and conquer approach *[Gyulassy et al., 2008]*
  - Working on the star of each vertex *[Robins et al., 2011]*
- Minimize the number of critical simplexes *[Cazals et al., 2003][Robins et al., 2011]*
- Compute accurate geometry of the Morse complexes *[Gyulassy et al., 2012]*

Computing a Forman gradient

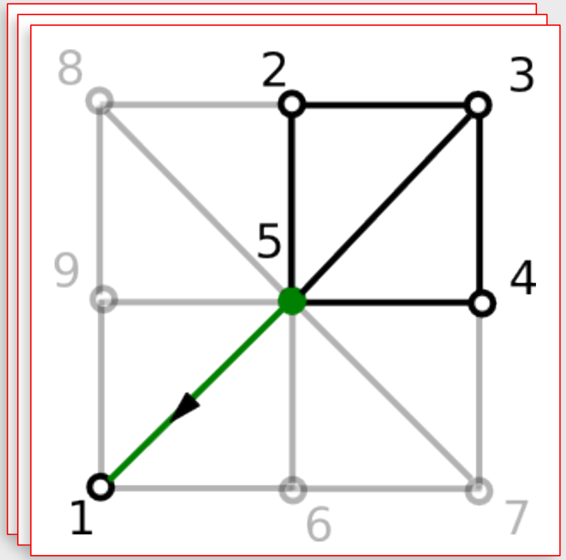
- Basics:
  - Starting from the vertices, simulate a Forman function while building the discrete gradient vector field
  - Navigate the V-paths of the discrete gradient vector field starting from the critical simplexes



- Parallelize the computation:
  - Working on the link of each vertex *[King et al., 2005]*
  - Divide and conquer approach *[Gyulassy et al., 2008]*
  - Working on the star of each vertex *[Robins et al., 2011]*
- Minimize the number of critical simplexes *[Cazals et al., 2003][Robins et al., 2011]*
- Compute accurate geometry of the Morse complexes *[Gyulassy et al., 2012]*

Computing a Forman gradient

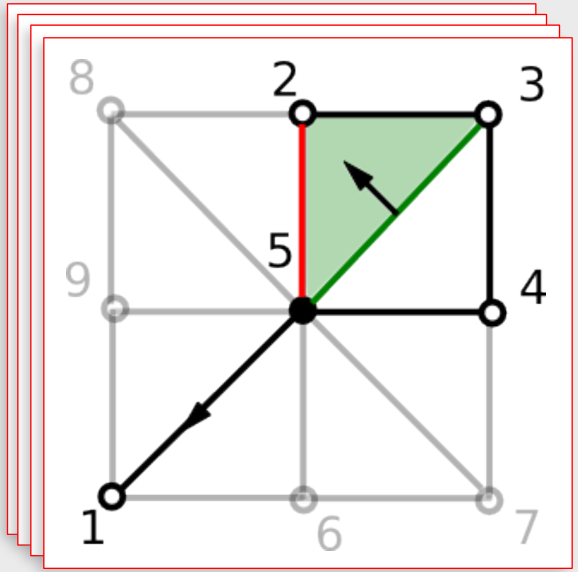
- Basics:
  - Starting from the vertices, simulate a Forman function while building the discrete gradient vector field
  - Navigate the V-paths of the discrete gradient vector field starting from the critical simplexes



- Parallelize the computation:
  - Working on the link of each vertex *[King et al., 2005]*
  - Divide and conquer approach *[Gyulassy et al., 2008]*
  - Working on the star of each vertex *[Robins et al., 2011]*
- Minimize the number of critical simplexes *[Cazals et al., 2003][Robins et al., 2011]*
- Compute accurate geometry of the Morse complexes *[Gyulassy et al., 2012]*

Computing a Forman gradient

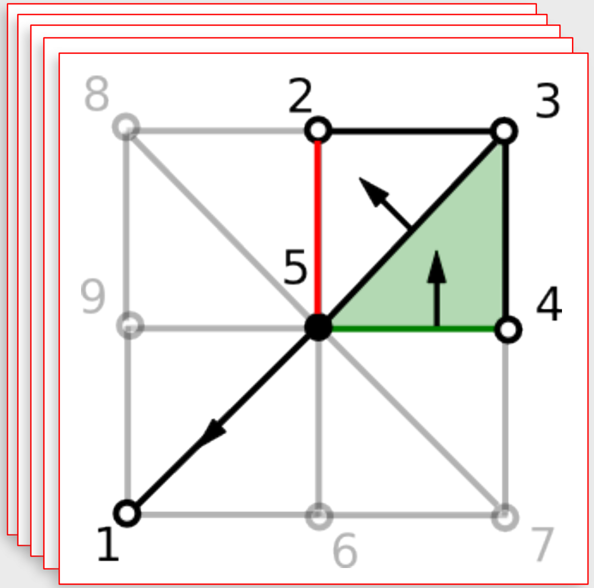
- Basics:
  - Starting from the vertices, simulate a Forman function while building the discrete gradient vector field
  - Navigate the V-paths of the discrete gradient vector field starting from the critical simplexes



- Parallelize the computation:
  - Working on the link of each vertex *[King et al., 2005]*
  - Divide and conquer approach *[Gyulassy et al., 2008]*
  - Working on the star of each vertex *[Robins et al., 2011]*
- Minimize the number of critical simplexes *[Cazals et al., 2003][Robins et al., 2011]*
- Compute accurate geometry of the Morse complexes *[Gyulassy et al., 2012]*

Computing a Forman gradient

- Basics:
  - Starting from the vertices, simulate a Forman function while building the discrete gradient vector field
  - Navigate the V-paths of the discrete gradient vector field starting from the critical simplexes



- Parallelize the computation:
  - Working on the link of each vertex *[King et al., 2005]*
  - Divide and conquer approach *[Gyulassy et al., 2008]*
  - Working on the star of each vertex *[Robins et al., 2011]*
- Minimize the number of critical simplexes *[Cazals et al., 2003][Robins et al., 2011]*
- Compute accurate geometry of the Morse complexes *[Gyulassy et al., 2012]*

Navigating Forman gradient

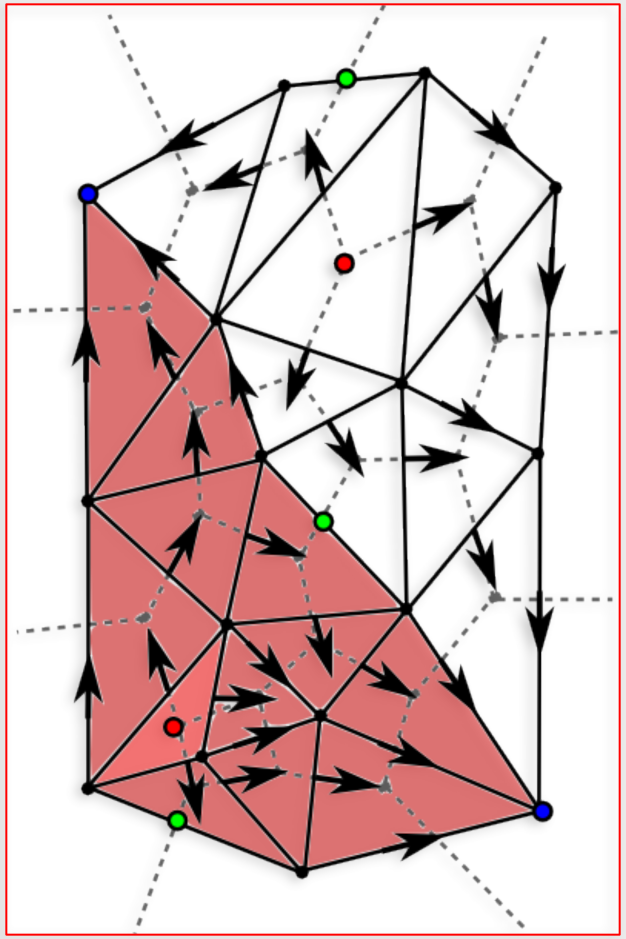
- Basics:
  - Starting from the vertices, simulate a Forman function while building the discrete gradient vector field
  - Navigate the V-paths of the discrete gradient vector field starting from the critical simplexes

- Boolean function for visiting each simplex only once *[Gunther et al., 2012][Weiss et al., 2013]*

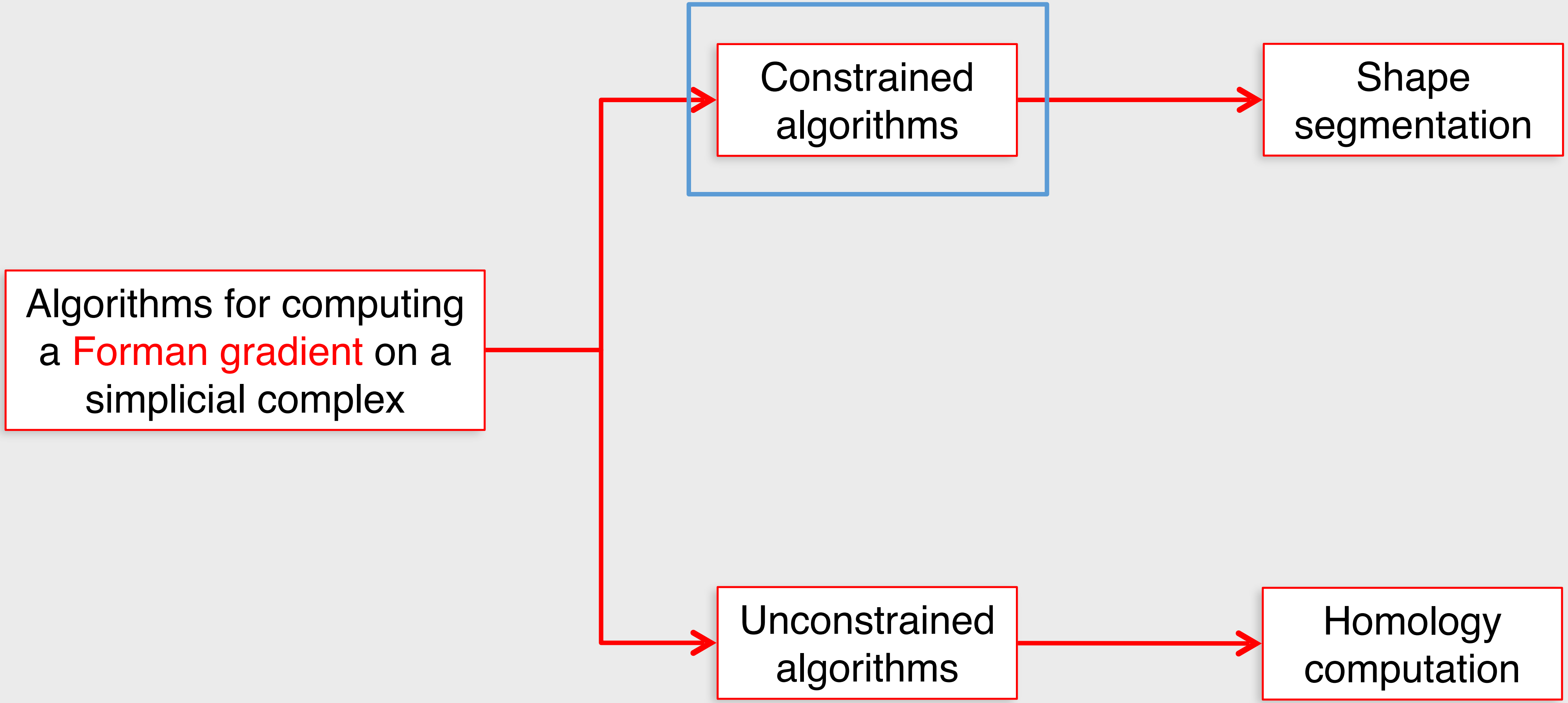
- Avoid the boolean function for minimizing memory consumption *[Shivashankarar et al., 2012]*

Navigating Forman gradient

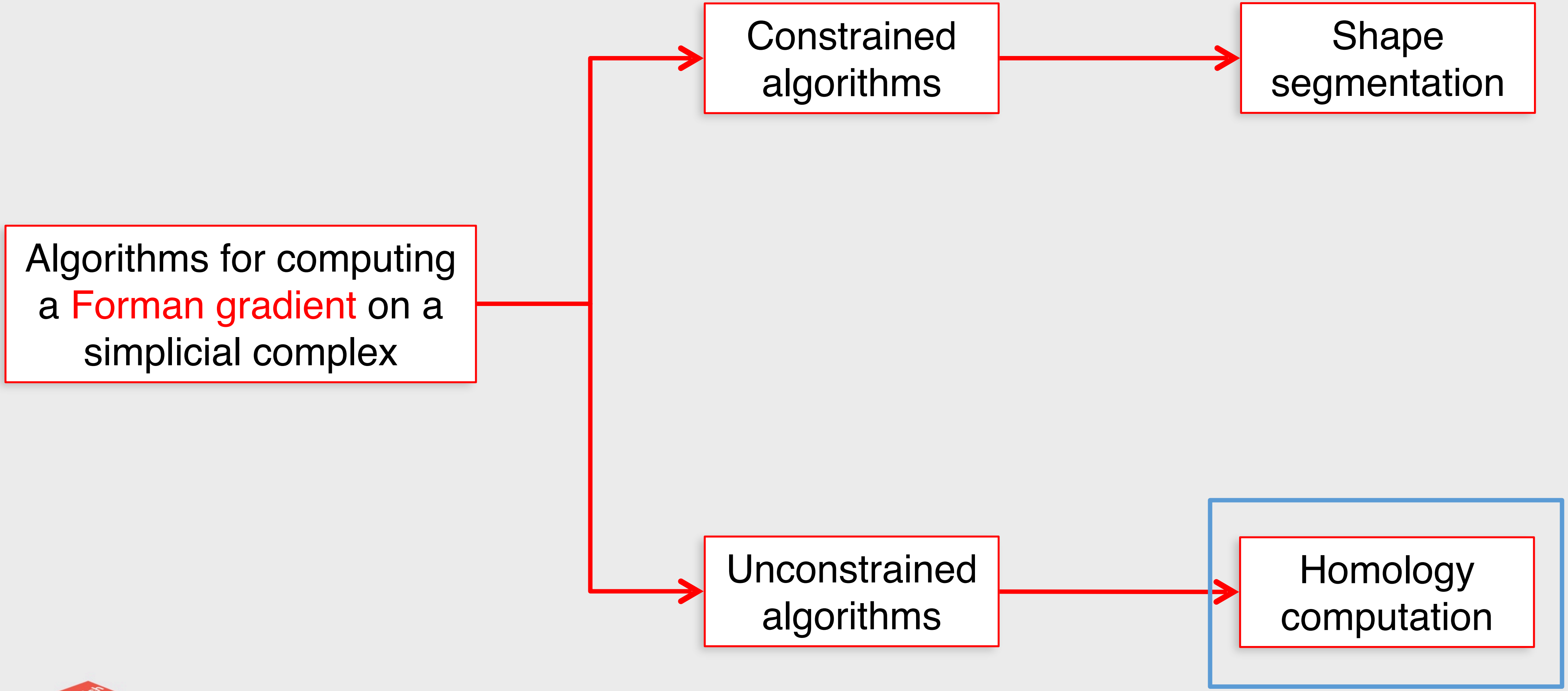
- Basics:
  - Starting from the vertices, simulate a Forman function while building the discrete gradient vector field
  - Navigate the V-paths of the discrete gradient vector field starting from the critical simplexes
- Boolean function for visiting each simplex only once *[Gunther et al., 2012][Weiss et al., 2013]*
- Avoid the boolean function for minimizing memory consumption *[Shivashankarar et al., 2012]*



Discrete Morse theory



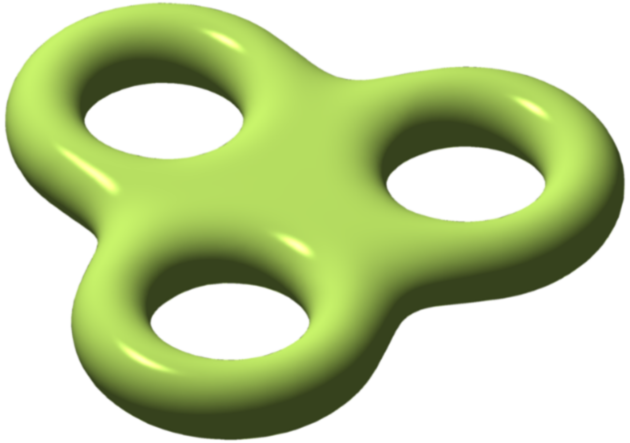
Discrete Morse theory



Simplicial Homology

Homology is a *topological invariant*

- roughly speaking, it counts and detects the *holes of various dimensions* in a topological space

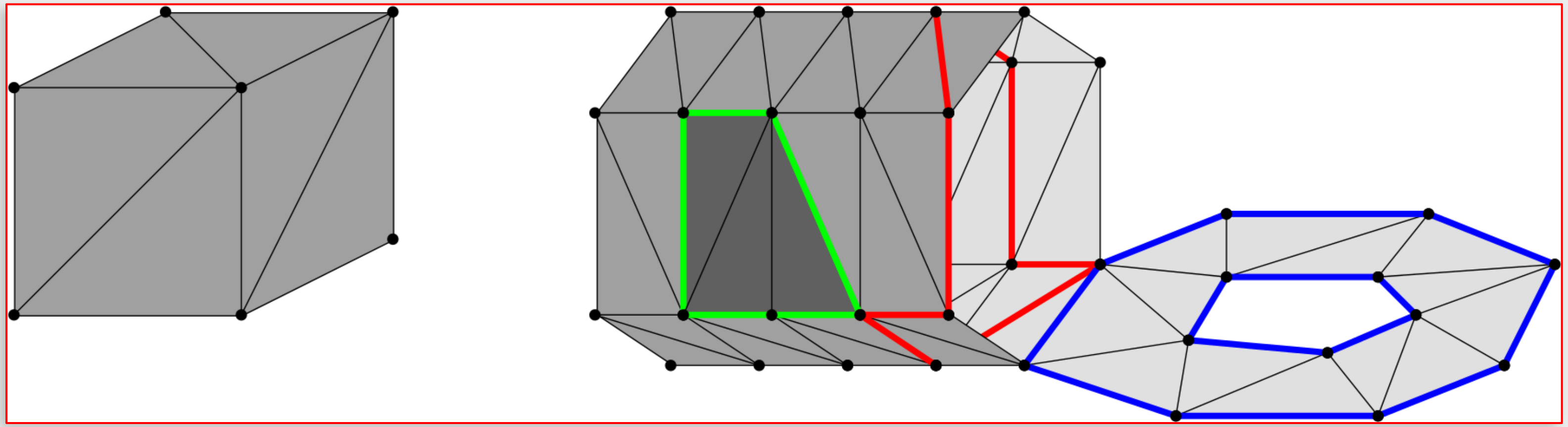


$$H_i(\Sigma) = \begin{cases} \mathbb{Z} & \text{if } i = 0 \\ \mathbb{Z}^6 & \text{if } i = 1 \\ \mathbb{Z} & \text{if } i = 2 \end{cases}$$

Homology groups can be computed, as opposed to homotopy groups or homeomorphism equivalence classes

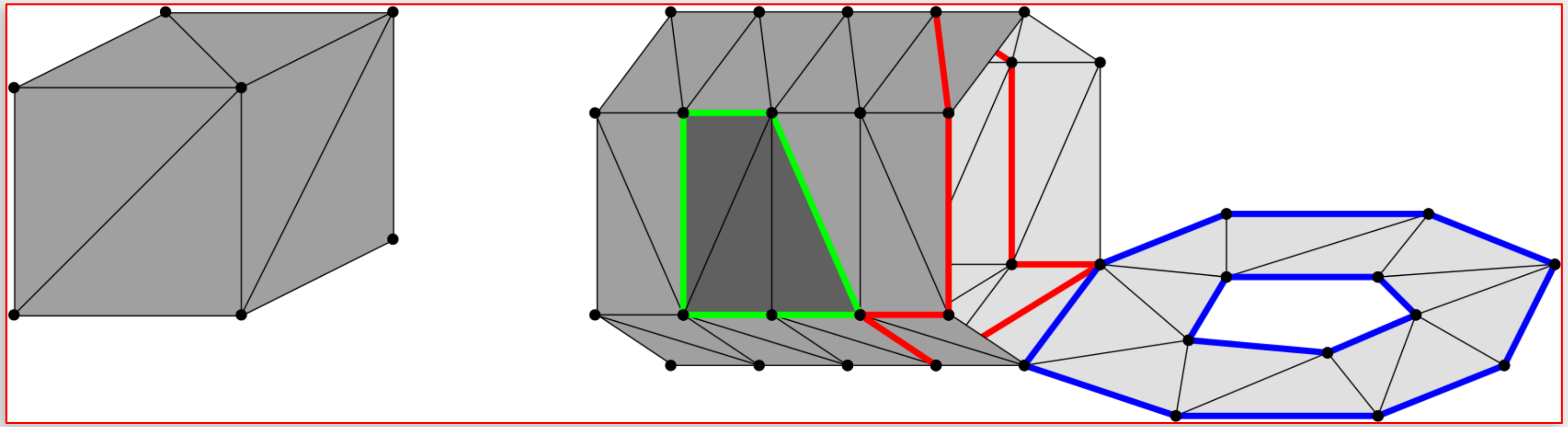
Simplicial homology - chains

- An i-chain c is a linear combination of i-simplices in Σ
- An i-cycle is a closed *i*-chain
  - Non-bounding cycle (e.g. blue or red cycles)
  - Bounding cycle (e.g. green cycle)



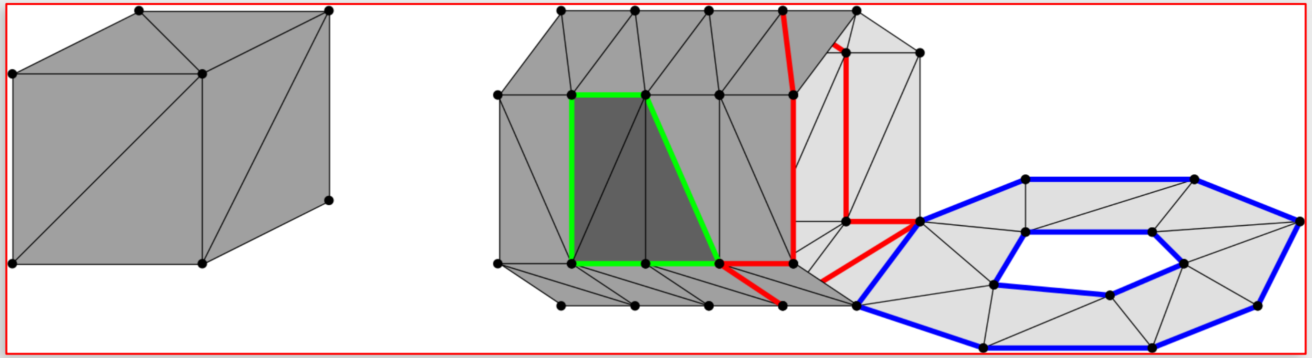
Simplicial homology - non-bounding cycles

- Two Non-Bounding cycles can be
  - Dependent (the two blue 1-cycles) if they represent the same homology class (the same hole)
  - Independent (the red and blue 1-cycles) if they represent different homology classes



Simplicial homology - Betti numbers

- Betti numbers count the number of independent non-bounding cycles in the object
  - i-th Betti number counts the number of i-cycles
  - Non-bounding cycles are also called generators



- $\beta_0 = 2$
  - two connected components
- $\beta_1 = 2$
  - two independent 1-cycles
- $\beta_2 = 1$
  - one 2-cycle

How can we compute homology?

- The classical technique is the *Smith Normal Formal algorithm (SNF)* *[Munkres, 1984]*

- It is based on the reduction of the boundary matrices of *K* which encode the boundary relationships between all the simplices of *K*.

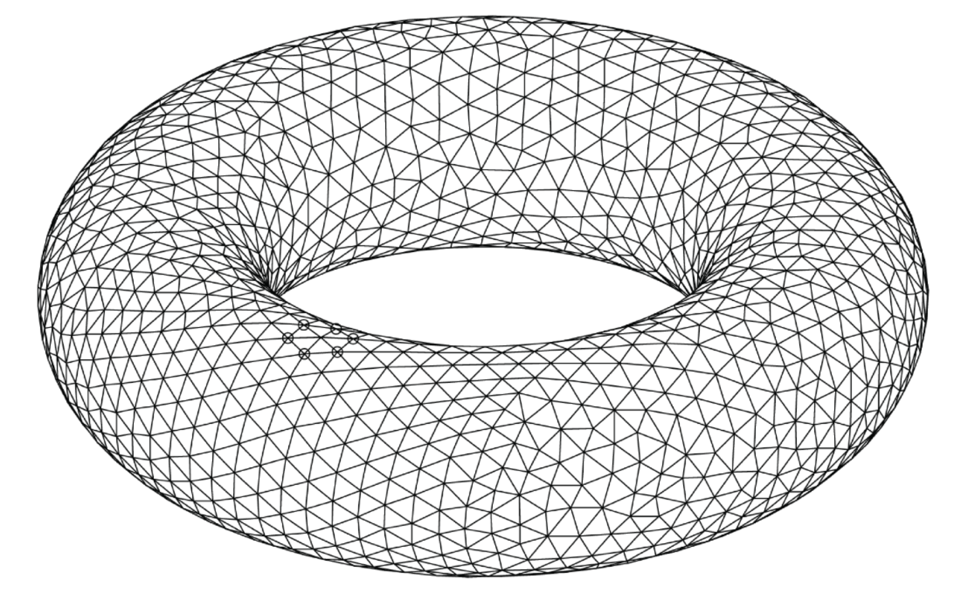
- The time complexity of the SNF algorithm is super-cubical in the number of the simplices of *K*

Homology and discrete Morse theory

- Reduction in the complexity of homology computation on a simplicial complex Σ
  - by considering a discrete Morse complex M associated with Σ
- Steps:
  - Generate a discrete Morse gradient V on the simplicial complex
  - Compute Morse complex
- Result: Σ and of M have isomorphic homology groups
- M has fewer cells than Σ

Homology and discrete Morse theory

- Reduction in the complexity of homology computation on a simplicial complex Σ
  - by considering a discrete Morse complex M associated with Σ

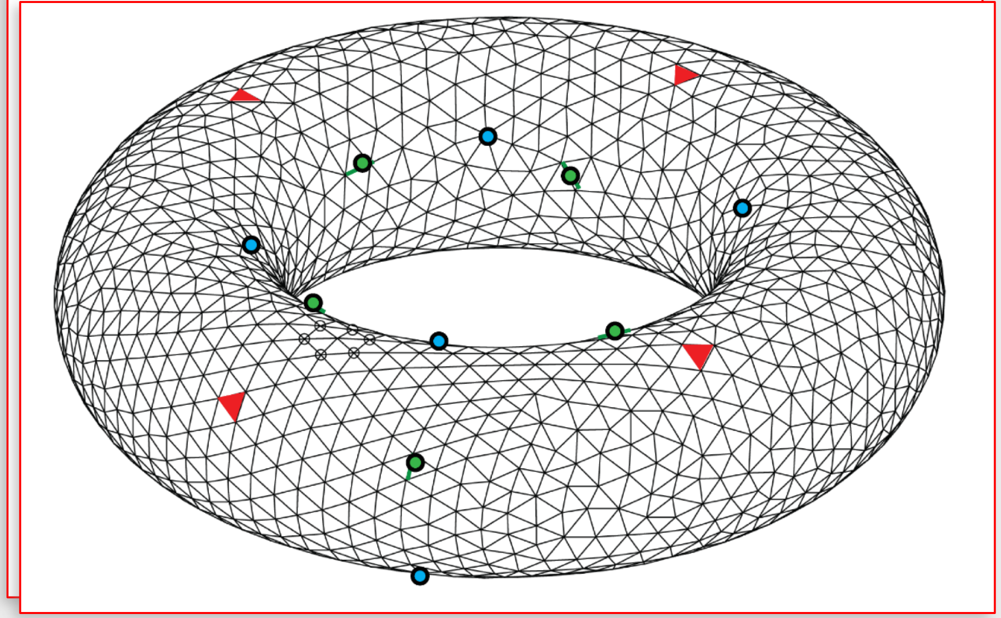


- Steps:
  - Generate a discrete Morse gradient V on the simplicial complex
  - Compute Morse complex

- Result: Σ and of M have isomorphic homology groups
- M has fewer cells than Σ

Homology and discrete Morse theory

- Reduction in the complexity of homology computation on a simplicial complex Σ
  - by considering a discrete Morse complex M associated with Σ

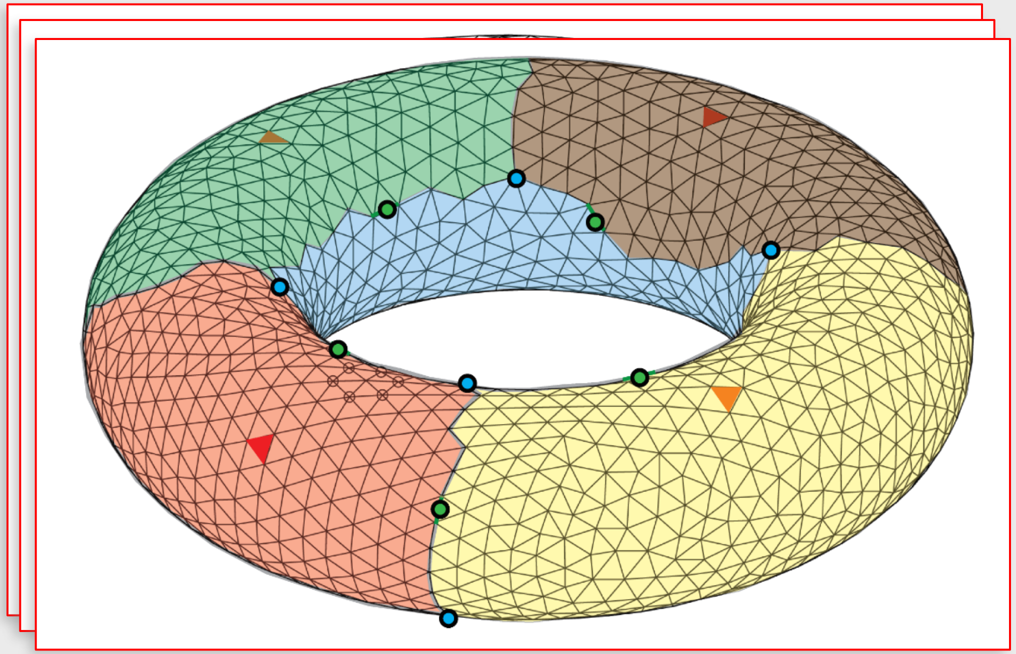


- Steps:
  - Generate a discrete Morse gradient V on the simplicial complex
  - Compute Morse complex

- Result: Σ and of M have isomorphic homology groups
- M has fewer cells than Σ

Homology and discrete Morse theory

- Reduction in the complexity of homology computation on a simplicial complex Σ
  - by considering a discrete Morse complex M associated with Σ



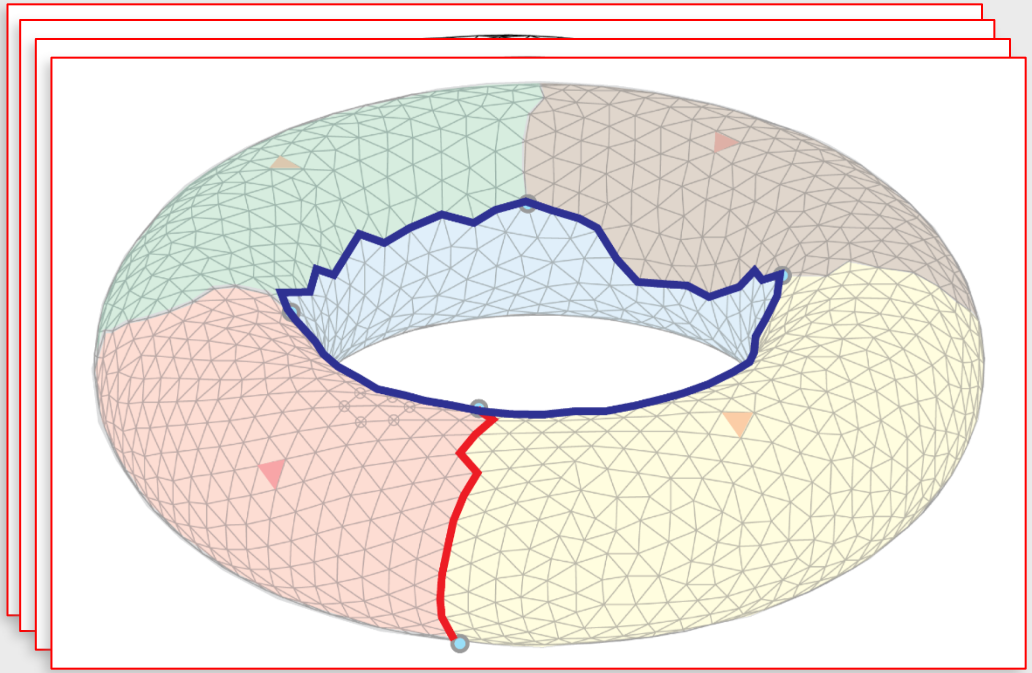
- Steps:
  - Generate a discrete Morse gradient V on the simplicial complex
  - Compute Morse complex

- Result: Σ and of M have isomorphic homology groups
- M has fewer cells than Σ

Homology and discrete Morse theory

- Reduction in the complexity of homology computation on a simplicial complex Σ
  - by considering a discrete Morse complex M associated with Σ

- Steps:
  - Generate a discrete Morse gradient V on the simplicial complex
  - Compute Morse complex

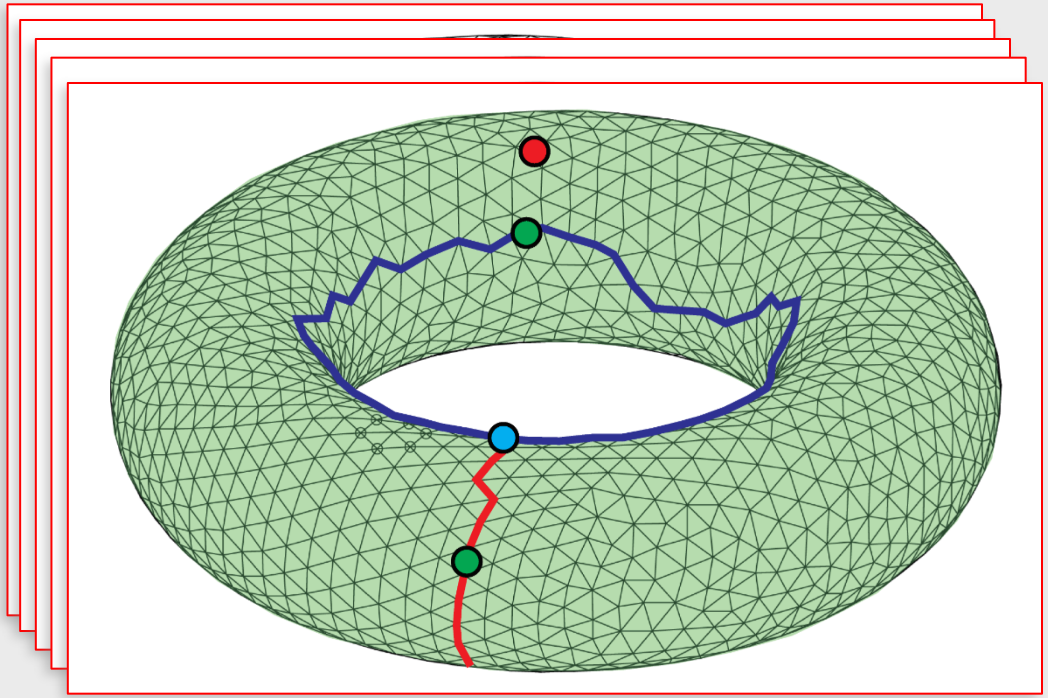


- Result: Σ and of M have isomorphic homology groups
- M has fewer cells than Σ

Homology and discrete Morse theory

- Reduction in the complexity of homology computation on a simplicial complex Σ
  - by considering a discrete Morse complex M associated with Σ

- Steps:
  - Generate a discrete Morse gradient V on the simplicial complex
  - Compute Morse complex



- Result: Σ and of M have isomorphic homology groups
- M has fewer cells than Σ

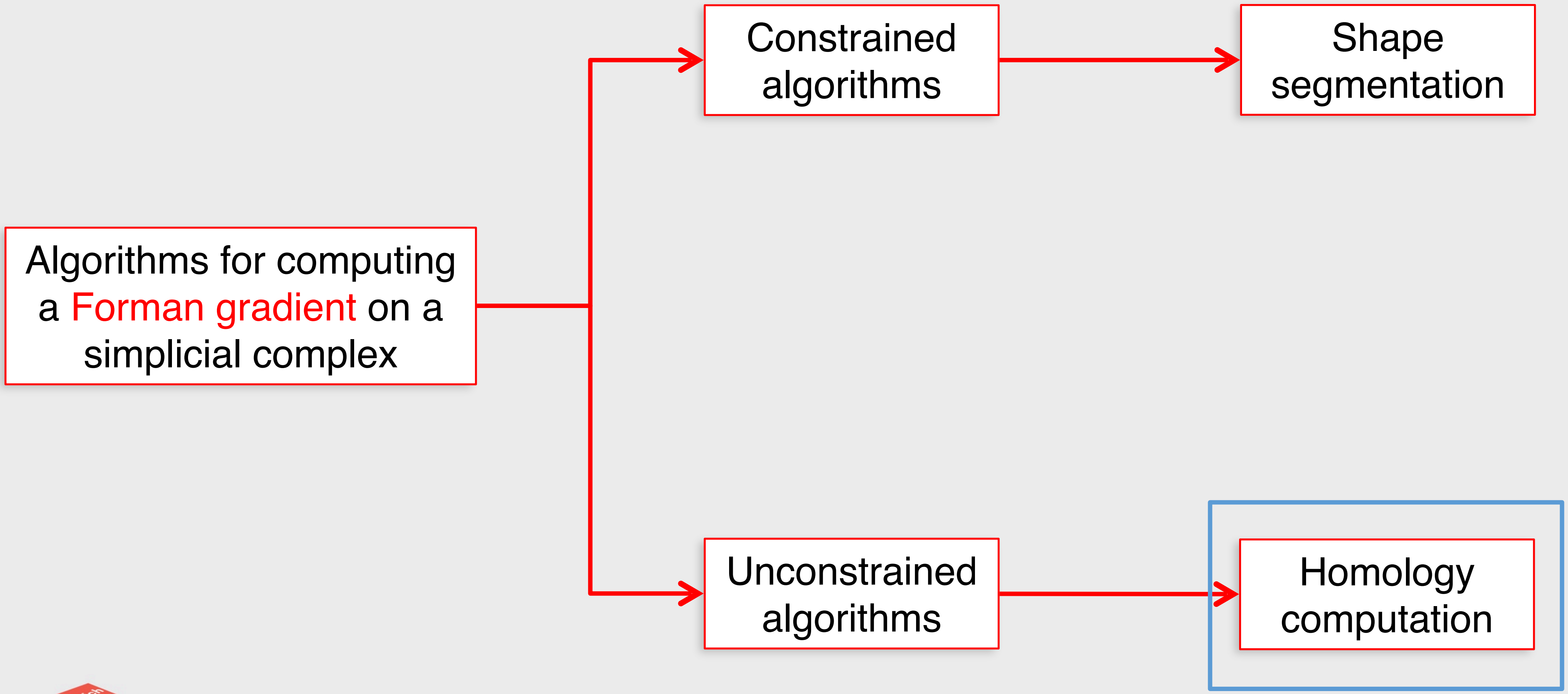
Perfect Morse Matching

β_0 = #{ 0-saddles } = 1

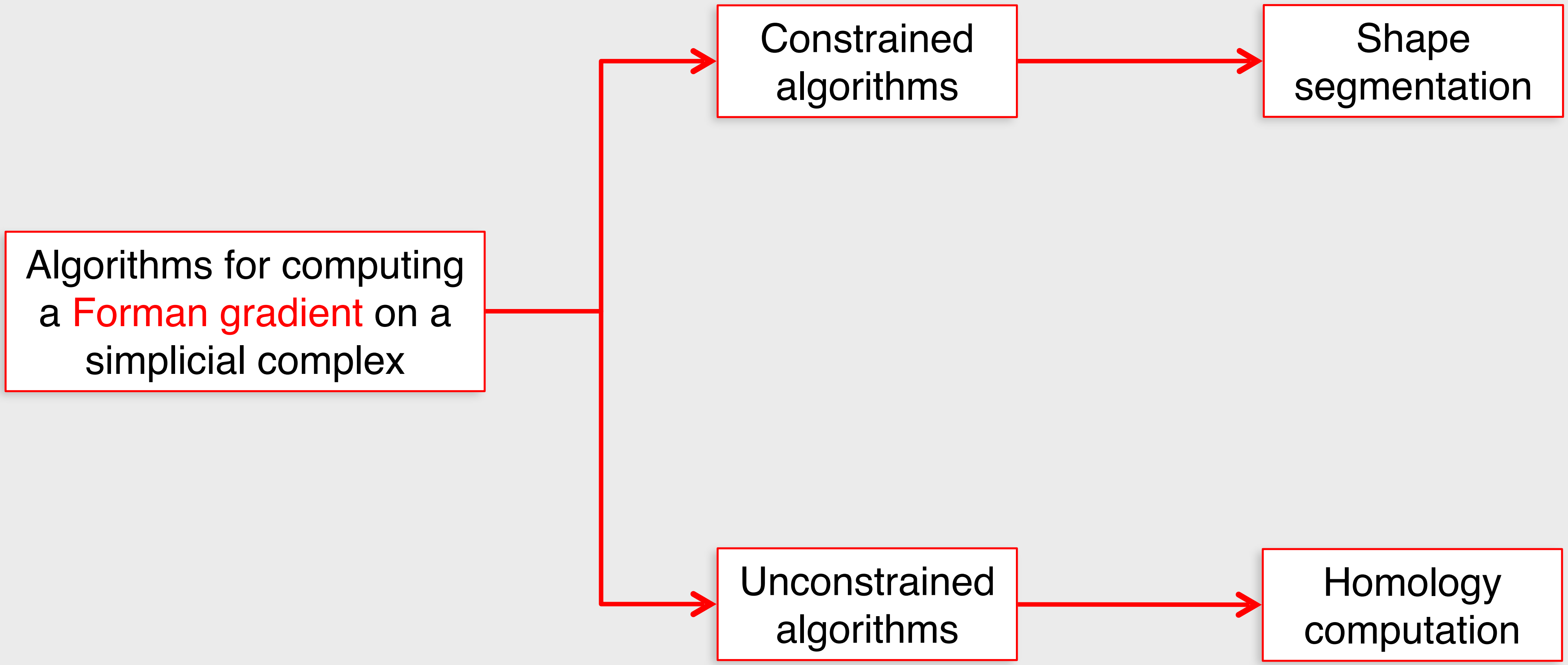
β_1 = #{ 1-saddles } = 2

β_2 = #{ 2-saddles } = 1

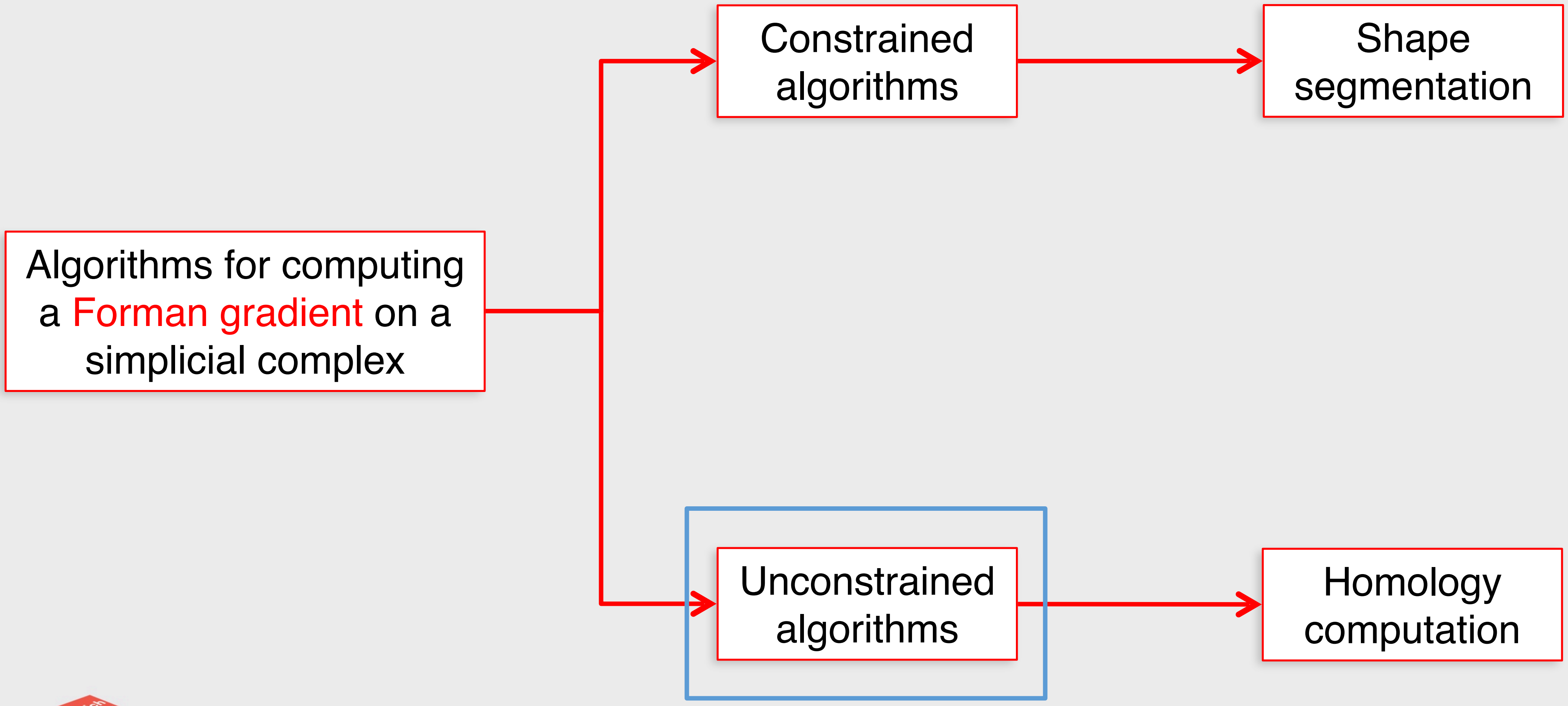
Discrete Morse theory



Discrete Morse theory



Discrete Morse theory



Unconstrained algorithm for homology comptuation

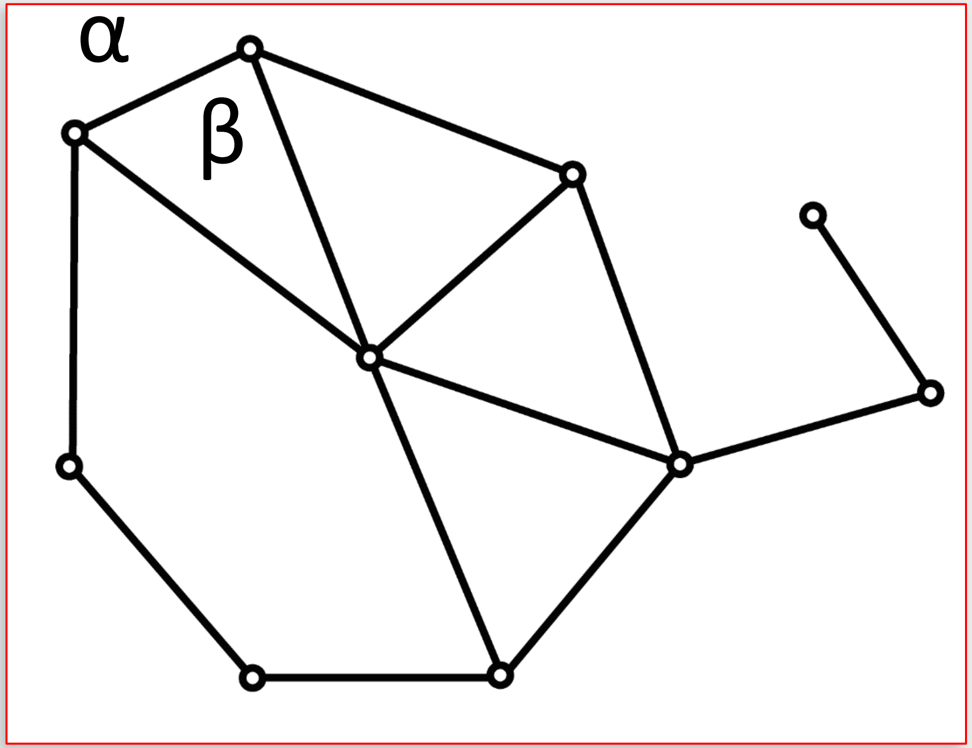
- Unconstrained algorithm: no scalar value
- Dimension dependent
  - 2-dimensional cell complexes *[Lewiner et al., 2003]*
- Approaches based on pairings critical simplex pairs:
  - Starting from top simplexes (reduction based algorithms) *[Benedetti et al., 2014]*
  - Starting from vertices (coreduction based algorithms) *[Harker et al., 2010] [Harker et al., 2014]*

Reduction based algorithm *[Benedetti et al., 2014]*

- Starting from maximal-simplexes
  - α: i-simplex
  - β: (i+1)-simplex

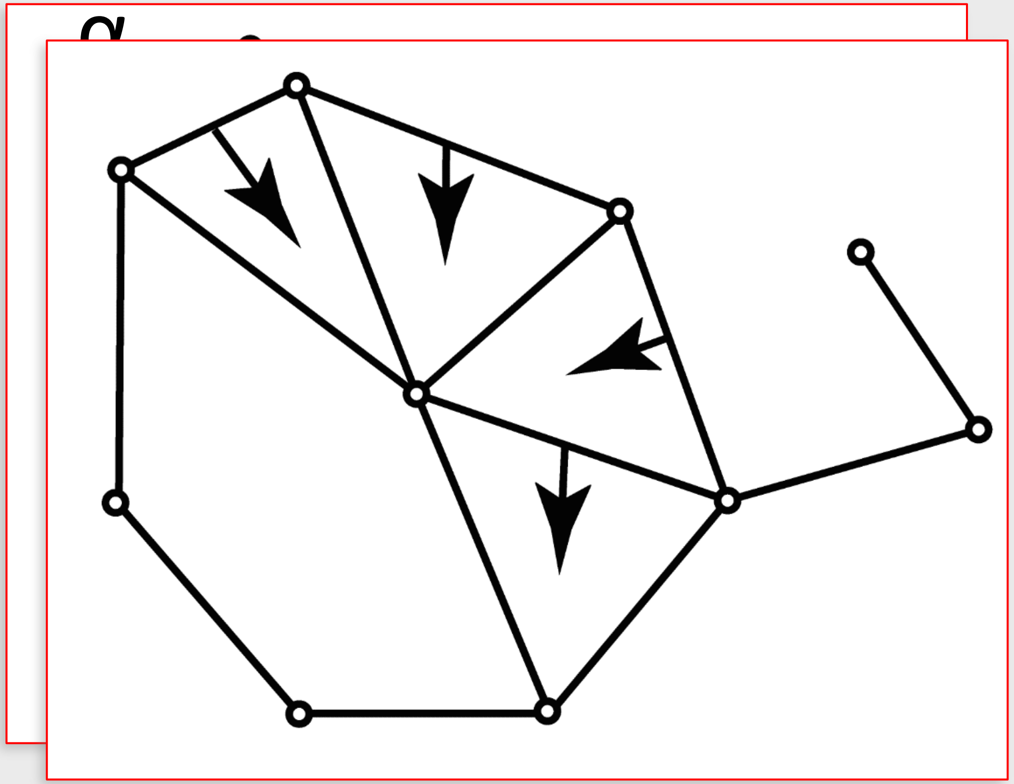
Reduction based algorithm *[Benedetti et al., 2014]*

- Starting from maximal-simplexes
  - α: i-simplex
  - β: (i+1)-simplex
  - α and β are paired if and only if β is the only simplex in the coboundary of α



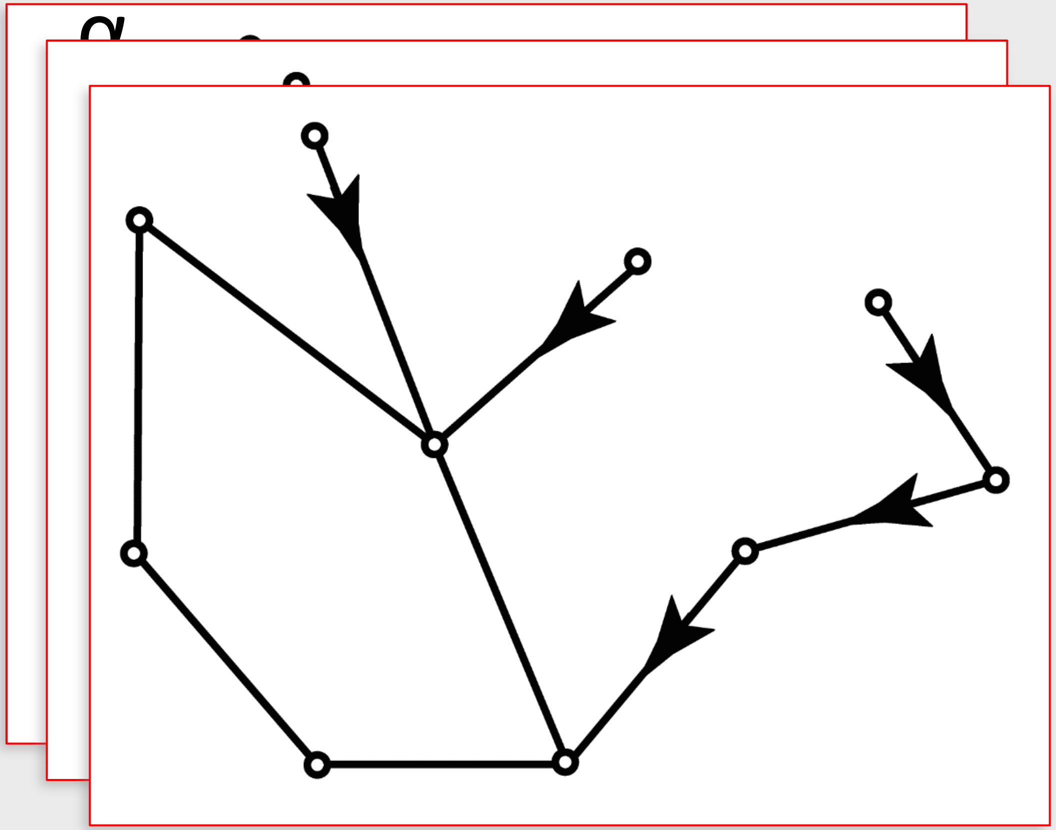
Reduction based algorithm *[Benedetti et al., 2014]*

- Starting from maximal-simplexes
  - α: i-simplex
  - β: (i+1)-simplex
  - α and β are paired if and only if β is the only simplex in the coboundary of α



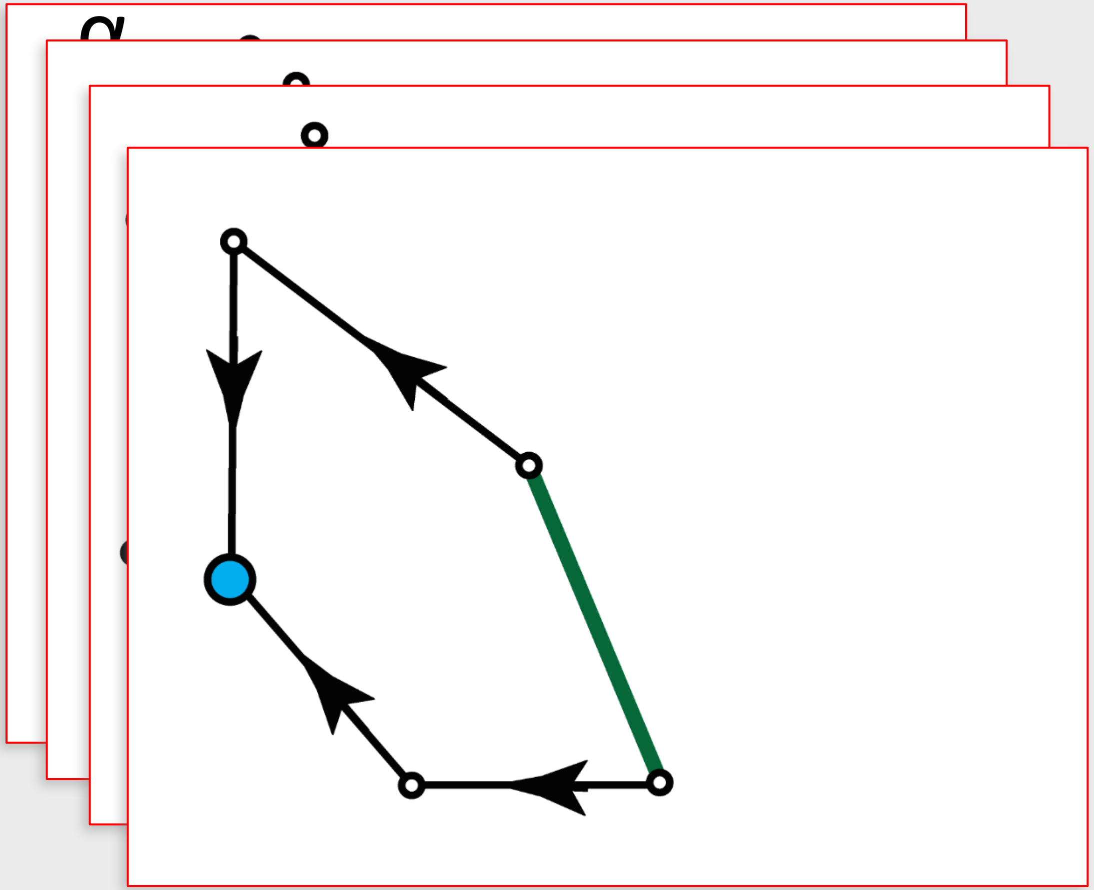
Reduction based algorithm *[Benedetti et al., 2014]*

- Starting from maximal-simplexes
  - α: i-simplex
  - β: (i+1)-simplex
  - α and β are paired if and only if β is the only simplex in the coboundary of α
  - when no more (i+1)-simplexes are available the working dimension is decreased



Reduction based algorithm *[Benedetti et al., 2014]*

- Starting from maximal-simplexes
  - α: i-simplex
  - β: (i+1)-simplex
  - α and β are paired if and only if β is the only simplex in the coboundary of α
  - when no more (i+1)-simplexes are available the working dimension is decreased



Coreduction based algorithm *[Harker et al., 2010] [Harker et al., 2014]*

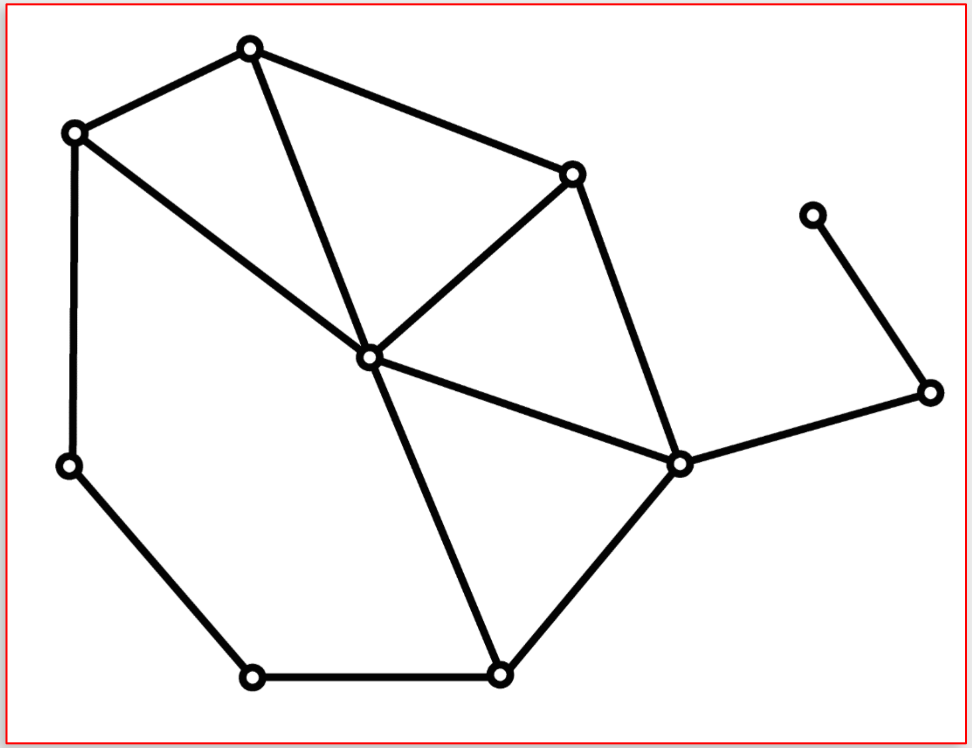
- Starting from vertices
  - α: i-simplex
  - β: (i+1)-simplex

Coreduction based algorithm *[Harker et al., 2010] [Harker et al., 2014]*

- Starting from vertices
  - α: i-simplex
  - β: (i+1)-simplex
  - α and β are paired if and only if β is the only free simplex in the boundary of α
  - when no more (i+1)-simplexes are available the working dimension is increased

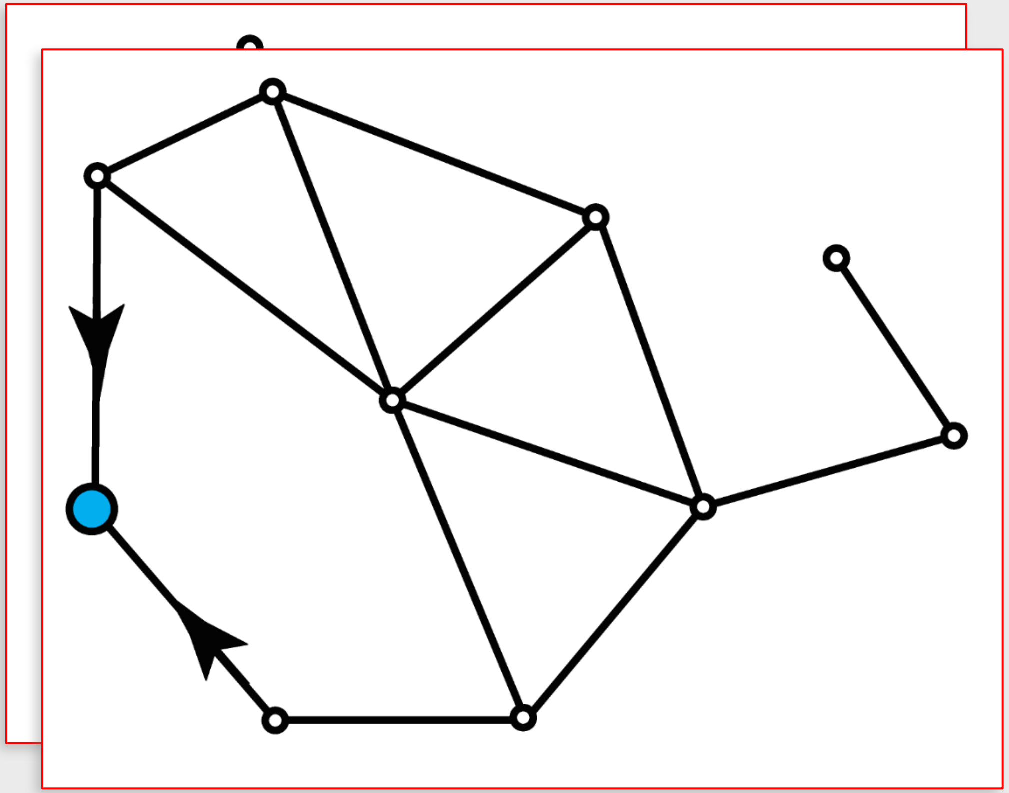
Coreduction based algorithm *[Harker et al., 2010] [Harker et al., 2014]*

- Starting from vertices
  - α: i-simplex
  - β: (i+1)-simplex
  - α and β are paired if and only if β is the only free simplex in the boundary of α
  - when no more (i+1)-simplexes are available the working dimension is increased

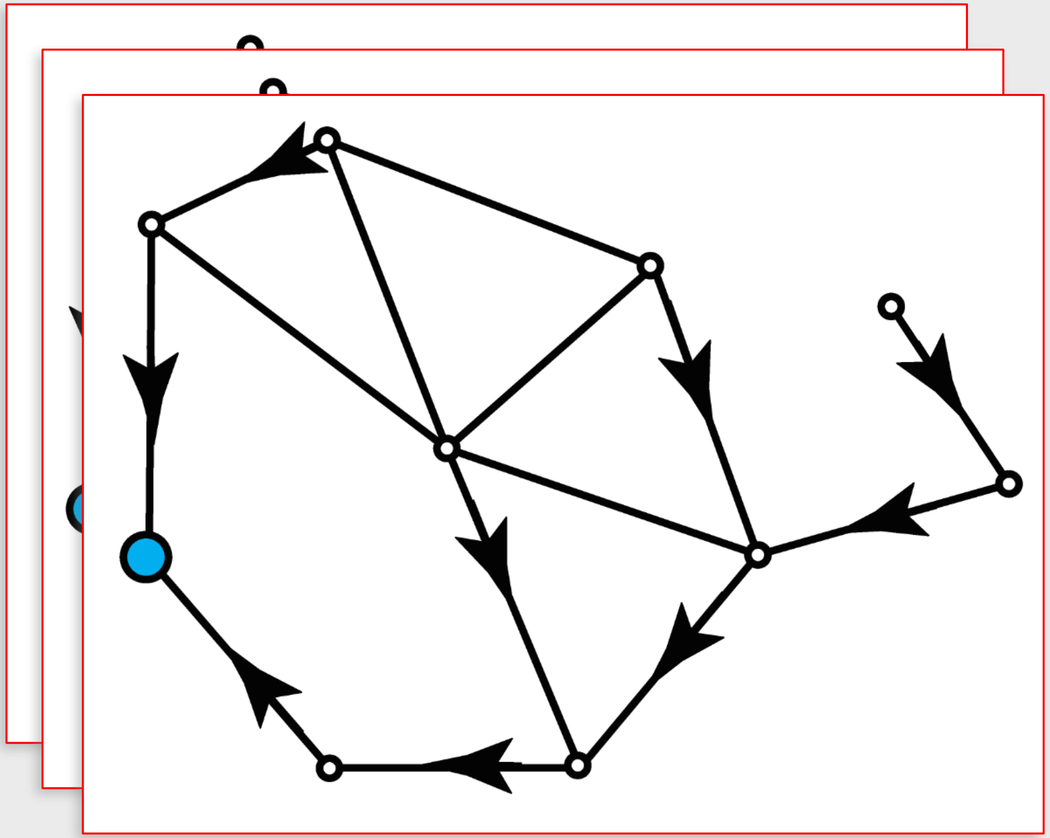


Coreduction based algorithm *[Harker et al., 2010] [Harker et al., 2014]*

- Starting from vertices
  - α: i-simplex
  - β: (i+1)-simplex
  - α and β are paired if and only if β is the only free simplex in the boundary of α
  - when no more (i+1)-simplexes are available the working dimension is increased



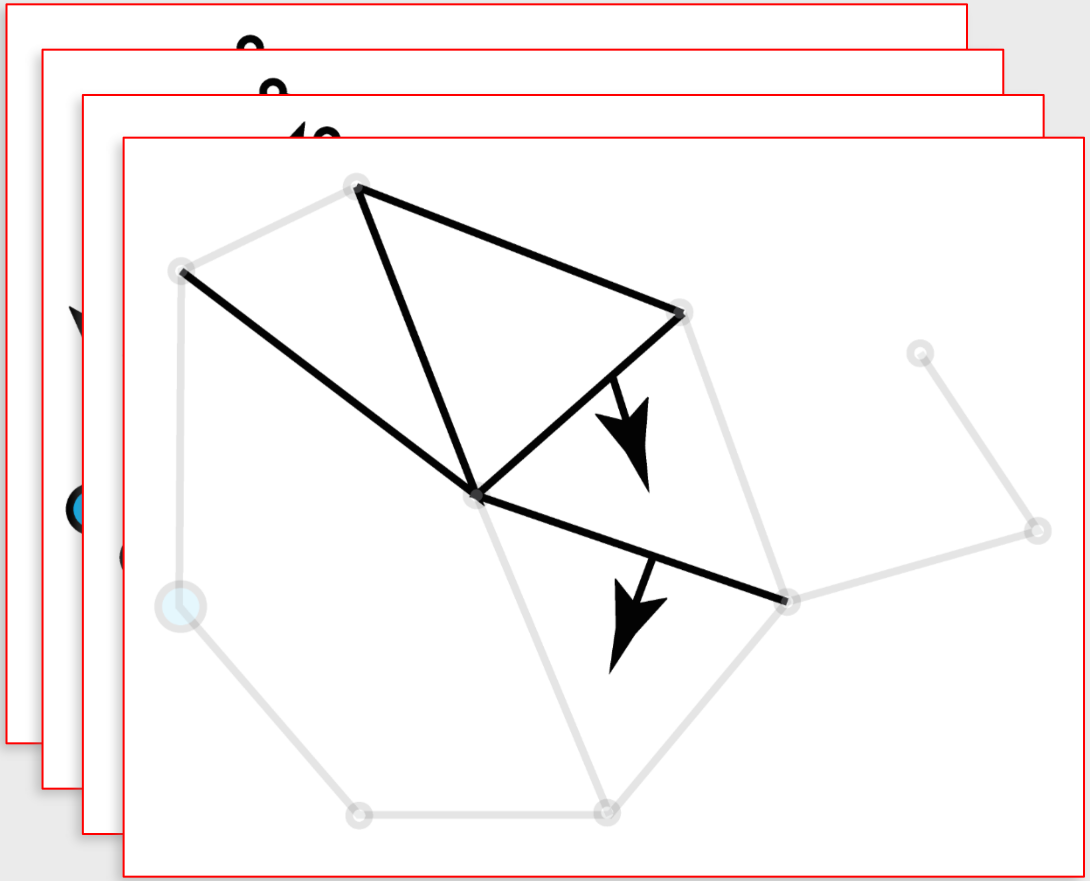
Coreduction based algorithm *[Harker et al., 2010] [Harker et al., 2014]*



- Starting from vertices
  - α: i-simplex
  - β: (i+1)-simplex
  - α and β are paired if and only if β is the only free simplex in the boundary of α
  - when no more (i+1)-simplexes are available the working dimension is increased

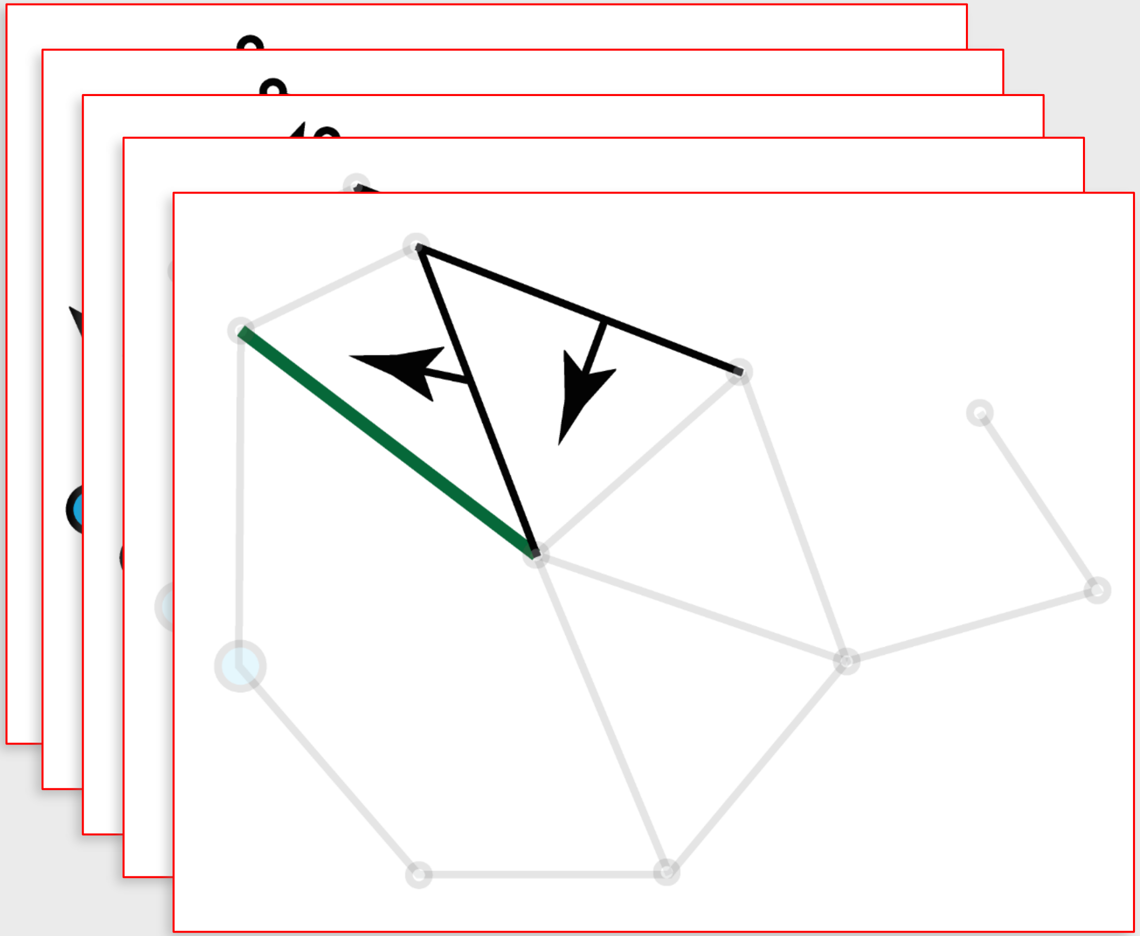
Coreduction based algorithm *[Harker et al., 2010] [Harker et al., 2014]*

- Starting from vertices
  - α: i-simplex
  - β: (i+1)-simplex
  - α and β are paired if and only if β is the only free simplex in the boundary of α
  - when no more (i+1)-simplexes are available the working dimension is increased



Coreduction based algorithm *[Harker et al., 2010] [Harker et al., 2014]*

- Starting from vertices
  - α: i-simplex
  - β: (i+1)-simplex
  - α and β are paired if and only if β is the only free simplex in the boundary of α
  - when no more (i+1)-simplexes are available the working dimension is increased



Algorithms for computing a Forman gradient

Approach	Input	Output	Algorithm
Constrained	Triangle meshes	Forman gradient	*Cazals et al., 2003*
	Tetrahedral meshes	"	*King et al., 2005*
	nD cell complex	"	*Gyulassy et al., 2008*
	nD cell complex	"	*Robins et al., 2011*
	nD cell complex	"	*Gyulassy et al., 2012*
Unconstrained	2D cell complex	"	*Lewiner et al., 2003*
	nD cell complex	"	*Benedetti et al., 2014*
	nD cell complex	"	*Harker et al., 2014*
	nD simplicial	"	*Fugacci et al., 2014*
Gradient Traversal	Forman gradient	All MS cells	*Gunther et al., 2012*
	Forman gradient	All MS cells	*Shivashankarar et al., 2012*
	Forman gradient	All MS cells	*Weiss et al., 2013*

Algorithms for computing a Forman gradient

Approach	Input	Output	Algorithm
Constrained	Triangle meshes	Forman gradient	*Cazals et al., 2003*
	Tetrahedral meshes	“	*King et al., 2005*
	nD cell complex	“	*Gyulassy et al., 2008*
	nD cell complex	“	*Robins et al., 2011*
	nD cell complex	“	*Gyulassy et al., 2012*
Unconstrained	2D cell complex	“	*Lewiner et al., 2003*
	nD cell complex	“	*Benedetti et al., 2014*
	nD cell complex	“	*Harker et al., 2014*
	nD simplicial	“	*Fugacci et al., 2014*
Gradient Traversal	Forman gradient	All MS cells	*Gunther et al., 2012*
	Forman gradient	All MS cells	*Shivashankarar et al., 2012*
	Forman gradient	All MS cells	*Weiss et al., 2013*

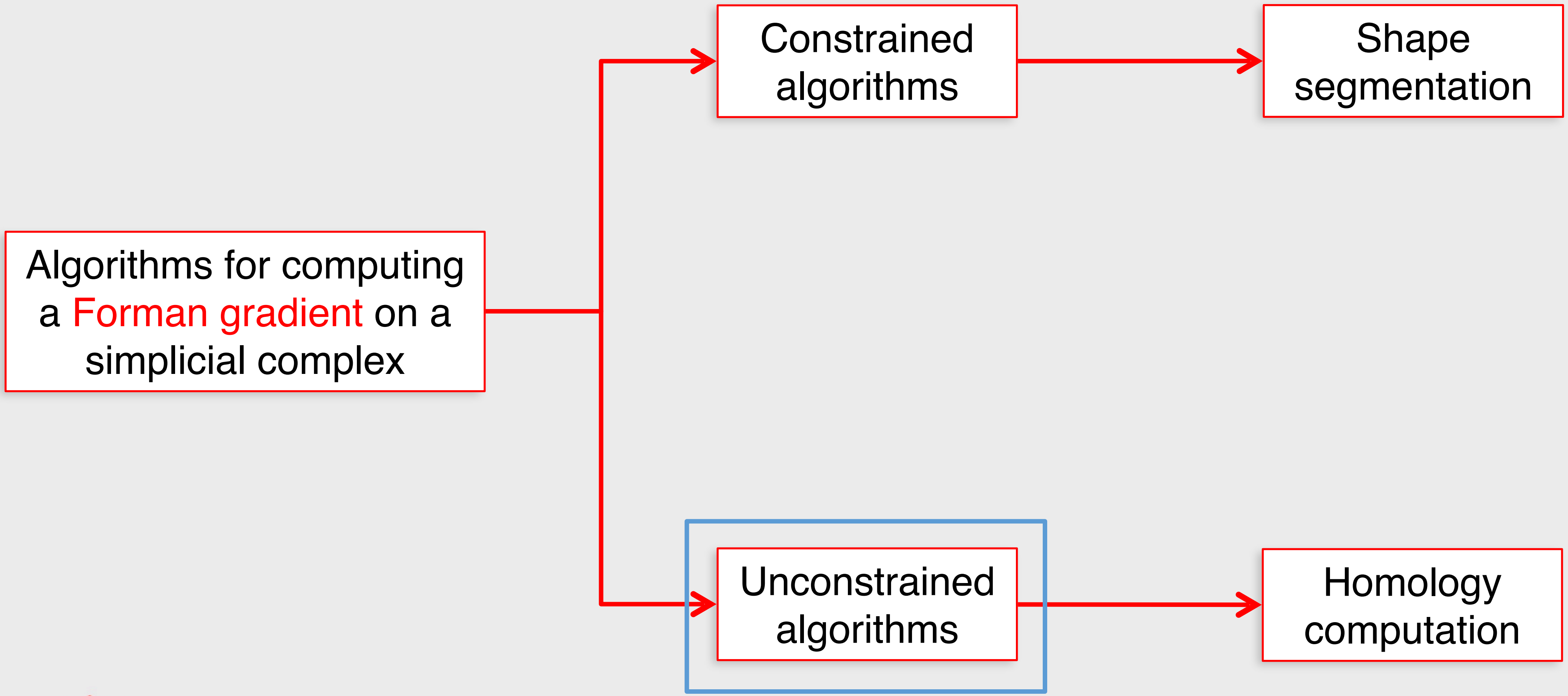
Algorithms for computing a Forman gradient

Approach	Input	Output	Algorithm
Constrained	Triangle meshes	Forman gradient	*Cazals et al., 2003*
	Tetrahedral meshes	"	*King et al., 2005*
	nD cell complex	"	*Gyulassy et al., 2008*
	nD cell complex	"	*Robins et al., 2011*
	nD cell complex	"	*Gyulassy et al., 2012*
Unconstrained	2D cell complex	"	*Lewiner et al., 2003*
	nD cell complex	"	*Benedetti et al., 2014*
	nD cell complex	"	*Harker et al., 2014*
	nD simplicial	"	*Fugacci et al., 2014*
Gradient Traversal	Forman gradient	All MS cells	*Gunther et al., 2012*
	Forman gradient	All MS cells	*Shivashankarar et al., 2012*
	Forman gradient	All MS cells	*Weiss et al., 2013*

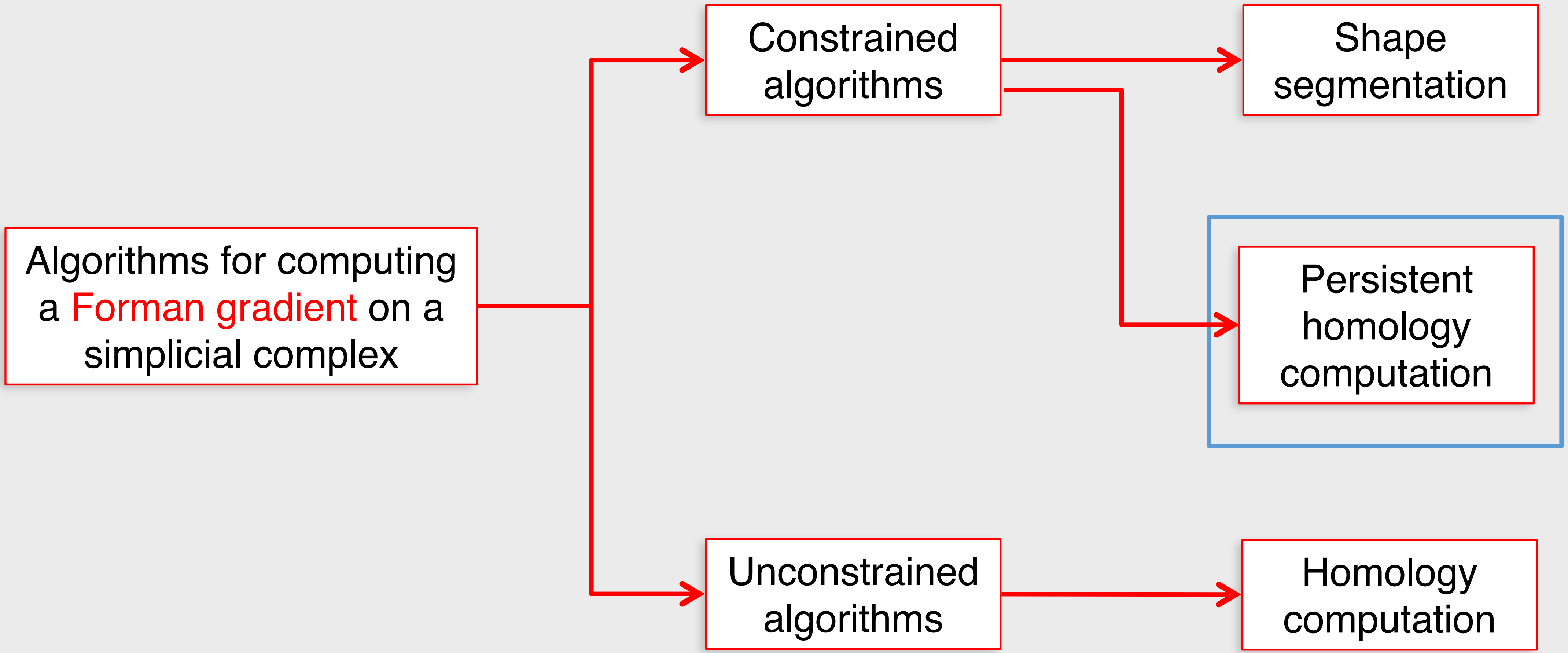
Algorithms for computing a Forman gradient

Approach	Input	Output	Algorithm
Constrained	Triangle meshes	Forman gradient	*Cazals et al., 2003*
	Tetrahedral meshes	"	*King et al., 2005*
	nD cell complex	"	*Gyulassy et al., 2008*
	nD cell complex	"	*Robins et al., 2011*
	nD cell complex	"	*Gyulassy et al., 2012*
Unconstrained	2D cell complex	"	*Lewiner et al., 2003*
	nD cell complex	"	*Benedetti et al., 2014*
	nD cell complex	"	*Harker et al., 2014*
	nD simplicial	"	*Fugacci et al., 2014*
Gradient Traversal	Forman gradient	All MS cells	*Gunther et al., 2012*
	Forman gradient	All MS cells	*Shivashankarar et al., 2012*
	Forman gradient	All MS cells	*Weiss et al., 2013*

Discrete Morse theory



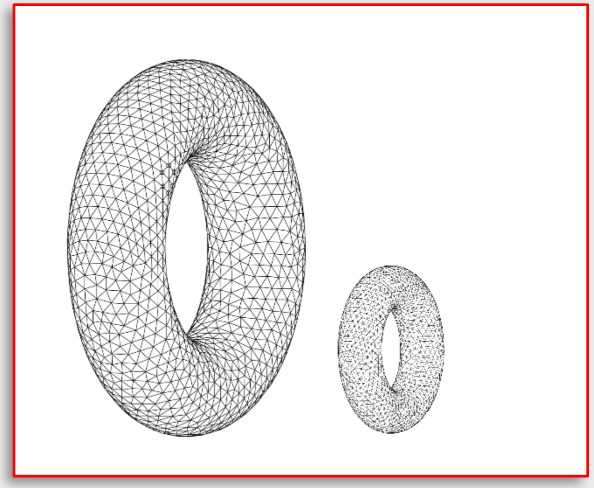
Discrete Morse theory



Persistent homology *[Edelsbrunner et Harer, 2008]*

- Defined for overcoming the limitations of homology

- First defining a scalar function on an object, persistent homology studies the changes in the homology of the object at the vary of the sublevel sets of the function

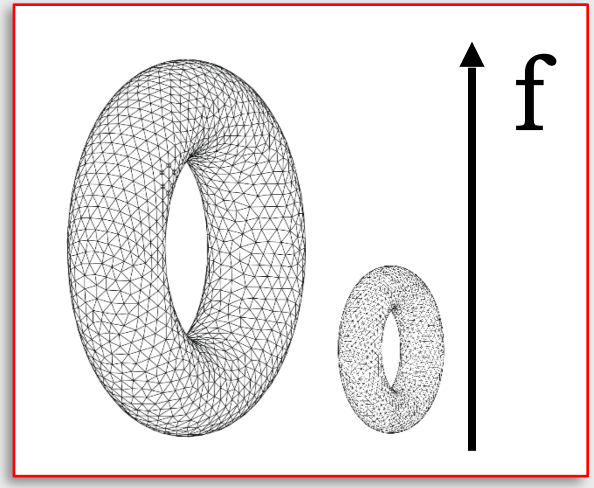


Characterize the homology of two different shapes

Persistent homology *[Edelsbrunner et Harer, 2008]*

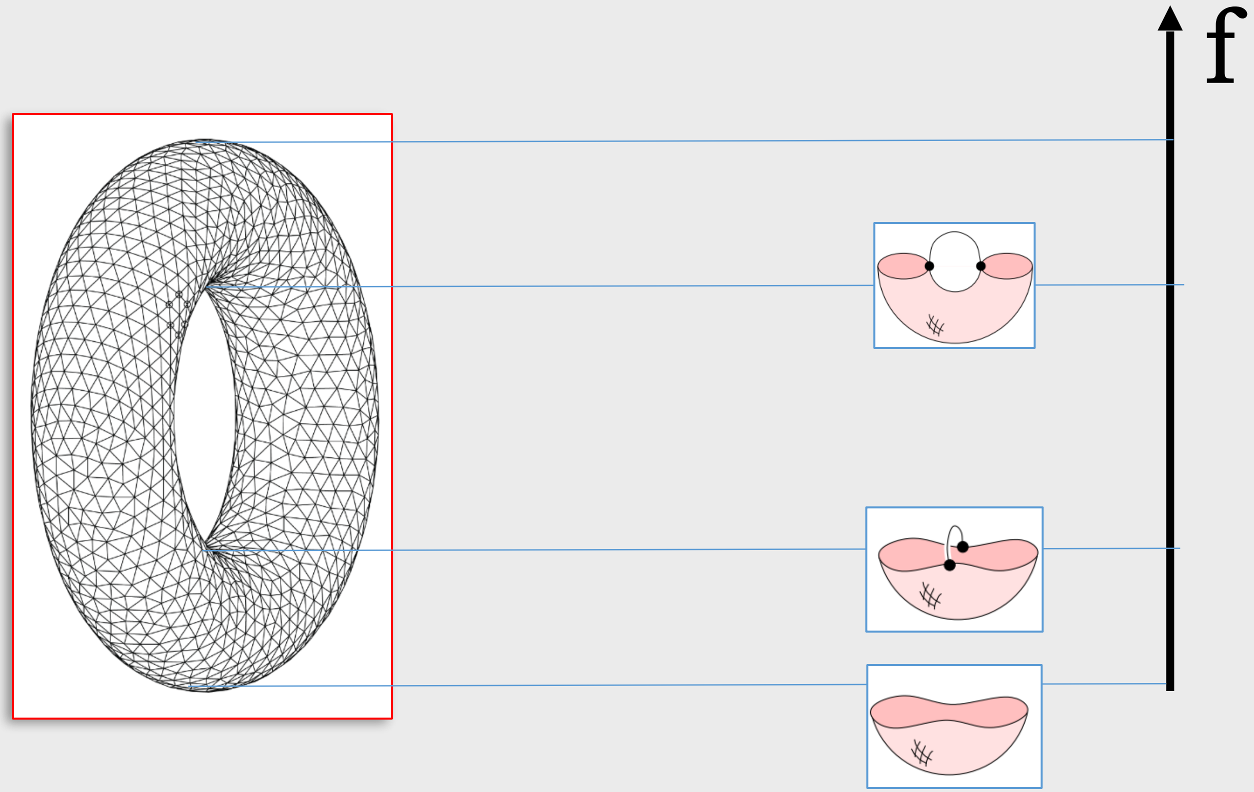
- Defined for overcoming the limitations of homology

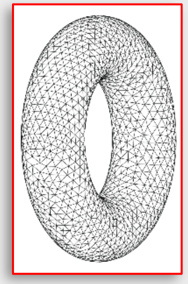
- First defining a scalar function on an object, persistent homology studies the changes in the homology of the object at the vary of the sublevel sets of the function



Characterize the homology of two different shapes

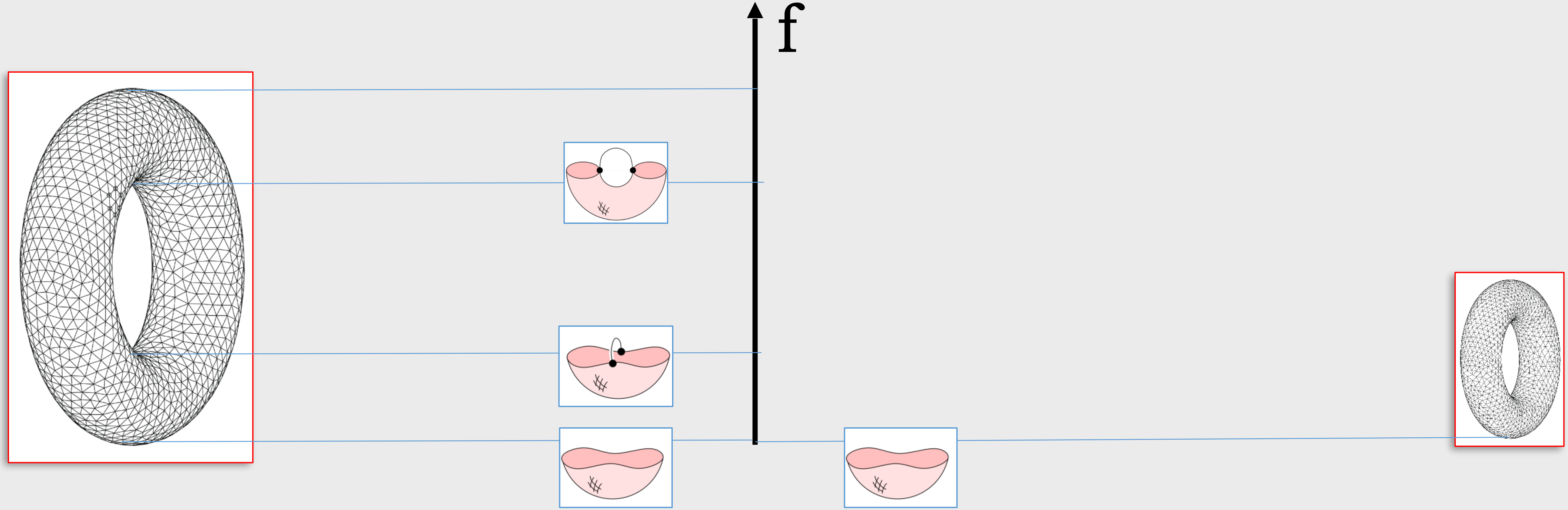
Persistent Homology (cont'd)





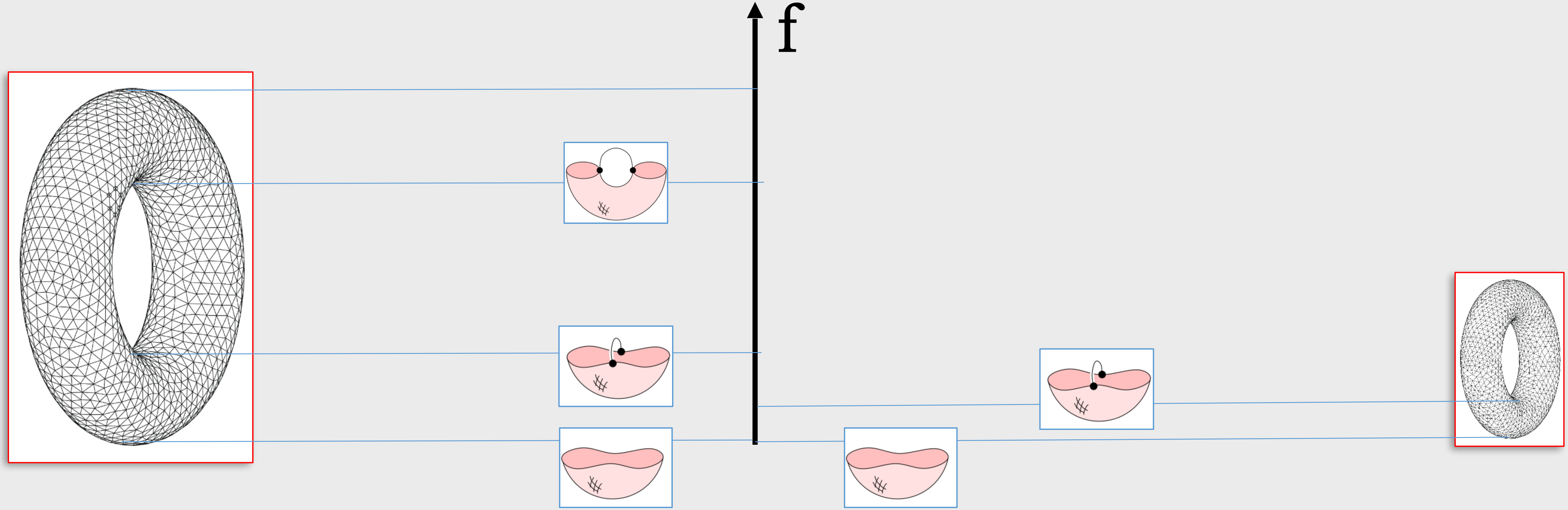
- Persistent Homology can be computed defining a scalar function on the vertices of the simplicial complex and computing the discrete Morse complex on it with a constrained approach.

Persistent Homology (cont'd)



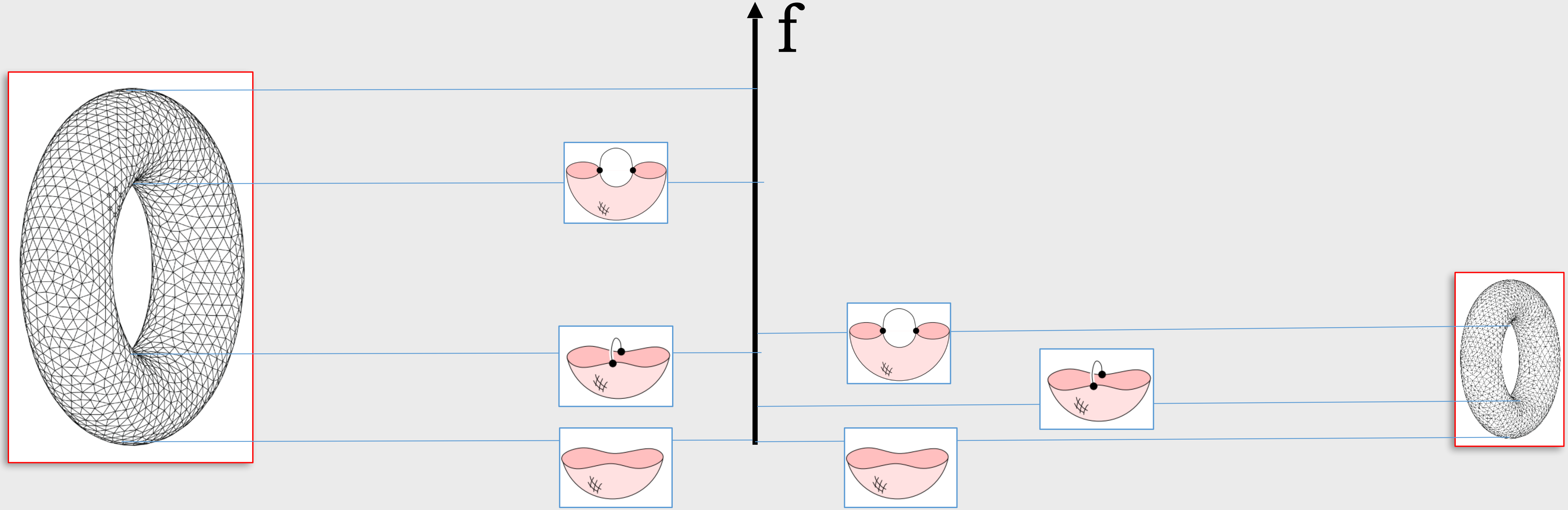
- Persistent Homology can be computed defining a scalar function on the vertices of the simplicial complex and computing the discrete Morse complex on it with a constrained approach.

Persistent Homology (cont'd)



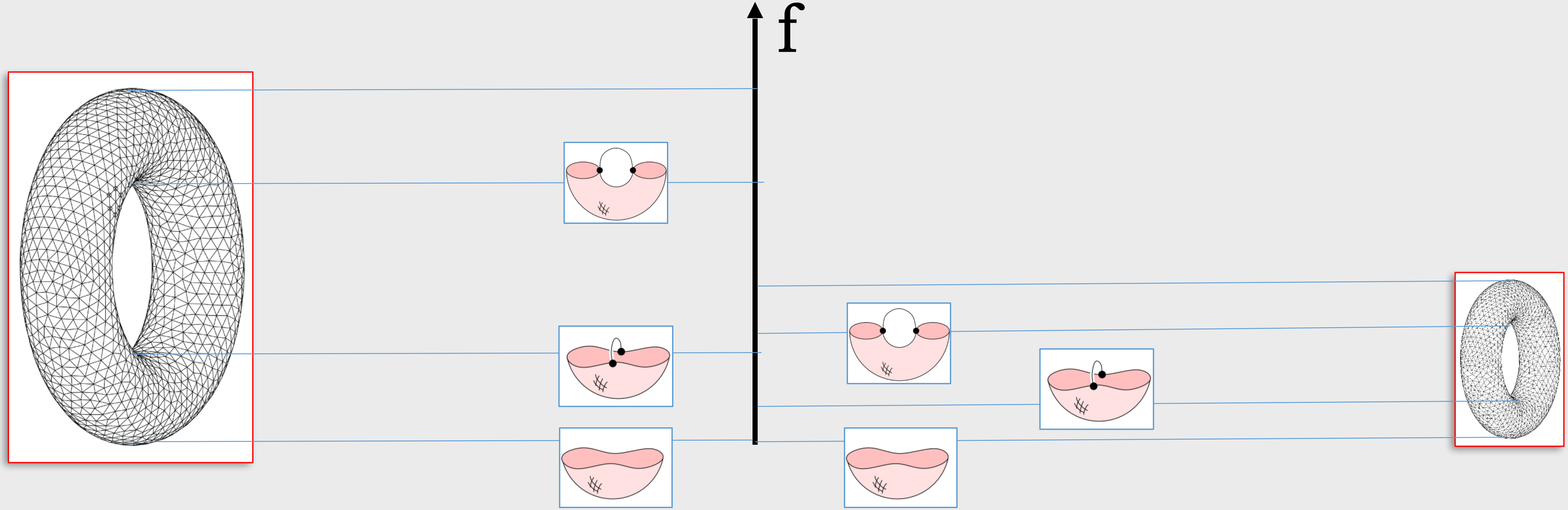
- Persistent Homology can be computed defining a scalar function on the vertices of the simplicial complex and computing the discrete Morse complex on it with a constrained approach.

Persistent Homology (cont'd)



- Persistent Homology can be computed defining a scalar function on the vertices of the simplicial complex and computing the discrete Morse complex on it with a constrained approach.

Persistent Homology (cont'd)



- Persistent Homology can be computed defining a scalar function on the vertices of the simplicial complex and computing the discrete Morse complex on it with a constrained approach.

Computing persistent homology

- For the 2D and 3D case *[Robins et al., 2011]* critical cells identified are in one-to-one correspondence with the topological changes in the sub-level sets of the function

- *[Gunther et al.,2012]* an efficient implementation has been defined for volumetric data

- *[Nanda et al., 2013]* a general algorithm for nD simplicial complexes has been defined.

Future developments

- Analysis of time dependent vector fields based on Morse theory
  - Works done in the 2D case [Reininghaus et al, 2011] [Kasten et al., 2011]
  - Semantic problems: identifying which topological structure best represent time varying data in 3D
  - Efficiency problems: how can we track these structure over time efficiently.
- Big data analysis:
  - Understanding the structure of high-dimensional data through homology and persistent homology
  - Need for new tools capable of dealing with large data sets in low, medium and high dimensions
- Persistence homology for multi-variate functions *[Carlson and Zomorodian, 2007] [Allili et al., 2015]*

Thank you for your attention

Questions?

Slides can be downloaded from
http://www.umiacs.umd.edu/~iuricich/

All the references and much more can be found on our paper

Zürich 2015

Eurographics 2015

The 36th Annual Conference of the
European Association for Computer Graphics