Lecture 7: Geometric Modeling and Visualization

Boundary & Finite Element Meshed Models III: Topologically Accurate Non-Linear Elements

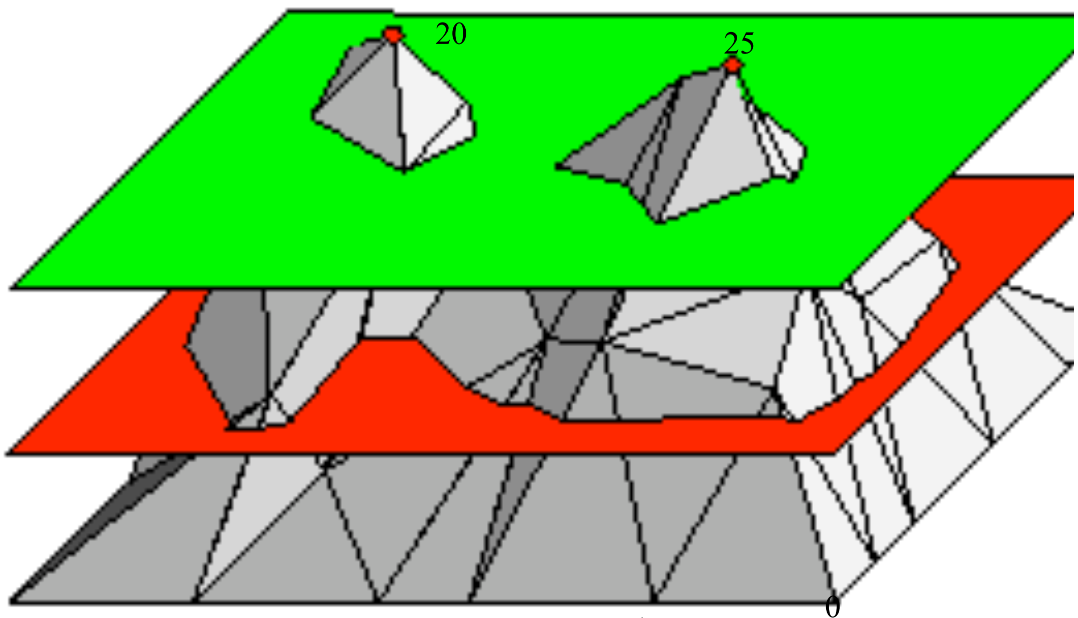
Chandrajit Bajaj

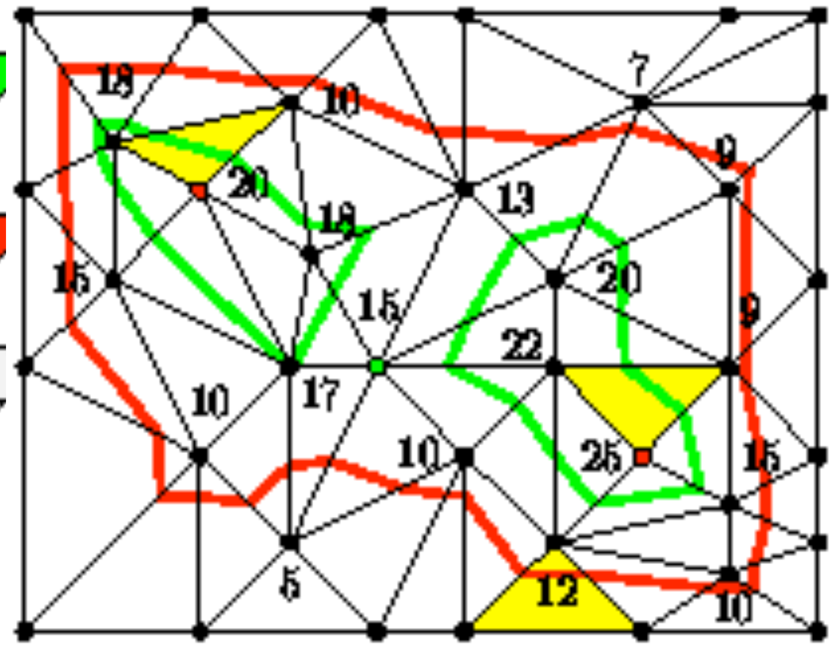
http://www.cs.utexas.edu/~bajaj

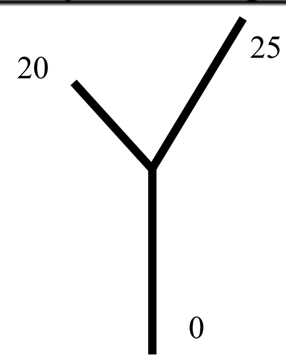
Center for Computational Visualization
Institute of Computational and Engineering Sciences
Department of Computer Sciences
University of Texas at Austin

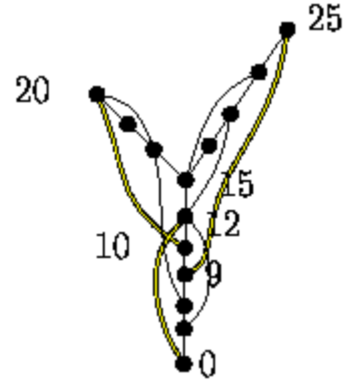
October 2007

Contouring: Capturing the Topology and Geometry of Zero Sets





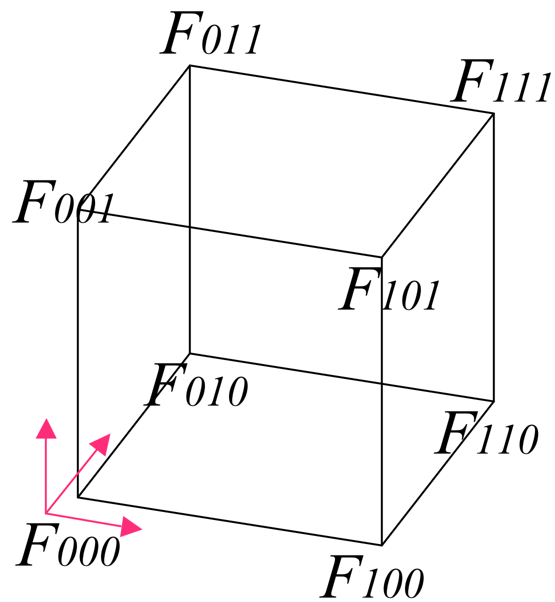




Isosurface of Trilinear Function

- Trilinear Function

$$\begin{aligned} F(x,y,z) &= F_{000}(1-x)(1-y)(1-z) \\ &+ F_{001}(1-x)(1-y)z \\ &+ F_{010}(1-x)y(1-z) \\ &+ F_{011}(1-x)yz \\ &+ F_{100}x(1-y)(1-z) \\ &+ F_{101}x(1-y)z \\ &+ F_{110}xy(1-z) \\ &+ F_{111}xyz \end{aligned}$$

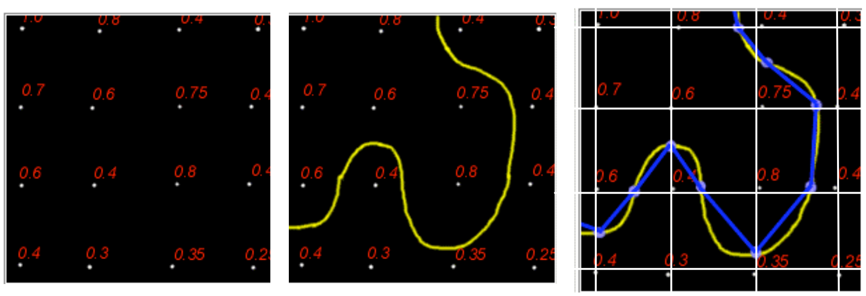


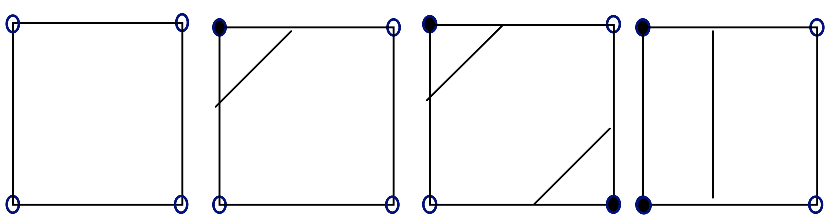
- Bilinear Function

$$\begin{aligned} F^f(x,y) &= F_{00}(1-x)(1-y) + F_{01}(1-x)y \\ &+ F_{10}x(1-y) + F_{11}xy \end{aligned}$$

Marching Cubes (MC) : Triangular Approximation

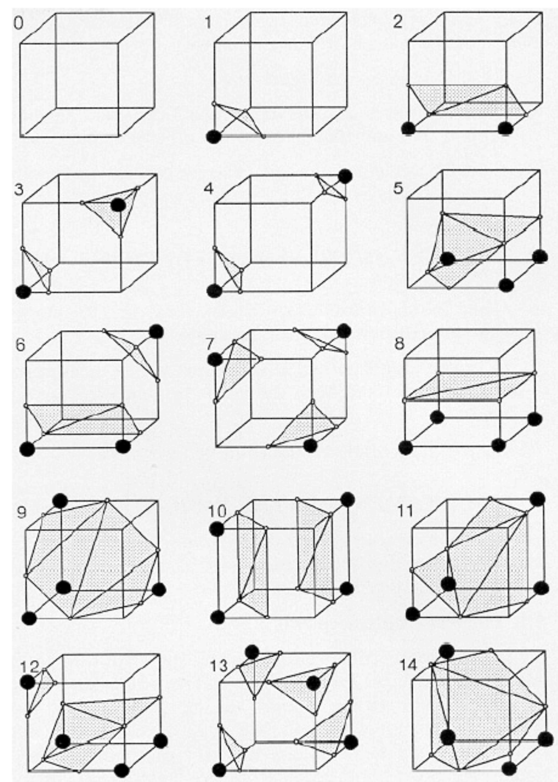
- 2D rectangle





- 3D cube :

  15 Cases for Triangulation



Saddle Points Computation

- Face Saddle Point

  $F(x,y) = ax + by + cxy + d$ (bilinear interpolant)

  First derivatives : $Fx = a + cy = 0$, $Fy = b + cx = 0$

  Saddle point $S = (-b/c , -a/c)$

- Body Saddle Point

$$F(x,y,z) = a + ex + cy + bz + gxy + fxz + dyz + hxyz$$

First derivatives = 0 :

$$F_x = e + gy + fz + hyz = 0$$

$$F_y = c + gx + dz + hxz = 0$$

$$F_z = b + fx + dy + hxy = 0$$

Face and Body Saddle Points

- We obtain saddle points :

$$x = -\frac{c+dz}{g+hz}$$

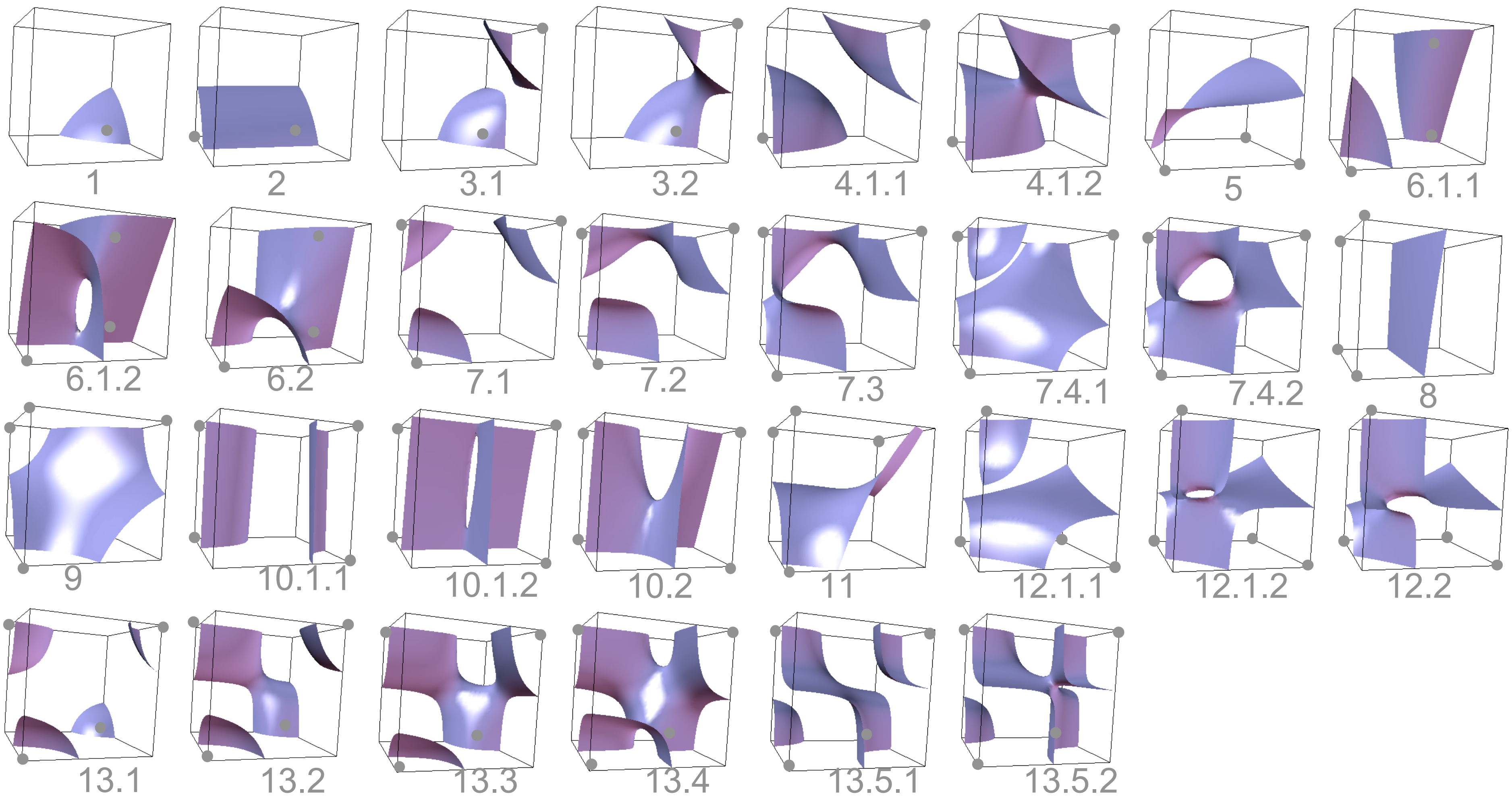
$$y = \frac{k_0+k_1 z}{k_2}$$

$$z = -\frac{g}{h} \pm \frac{\sqrt{g^2 k_1^2 - h k_1^{1/2} (e k_2 + g k_0)}}{h}$$

$$k_0 = cf - bg, k_1 = df - bh, k_2 = dg - ch$$

- saddle point outside the cube → discard
  (only case 13.5 has more than one valid body saddle point.)

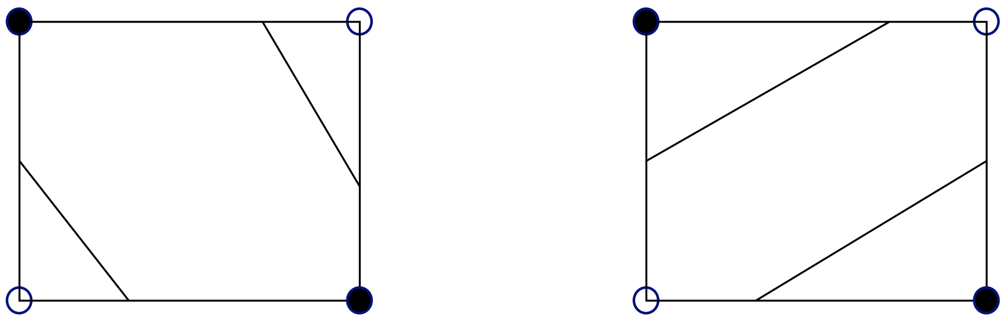
Trilinear Isosurface Topology 31 cases



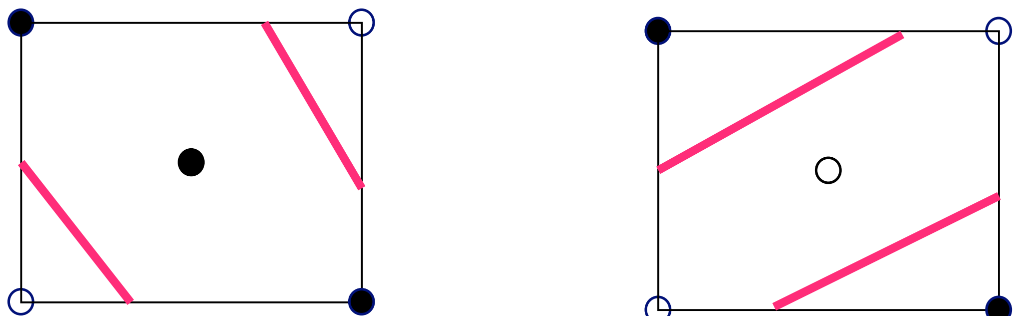
Center for Computational Visualization
Institute of Computational and Engineering Sciences
Department of Computer Sciences
University of Texas at Austin

Decision on Contour Topology (Nielson 92 : Asymptotic Decider)

- Resolving Face Ambiguity
  - Ambiguity (face saddle)



  - Decision based on the value *s* of saddle point

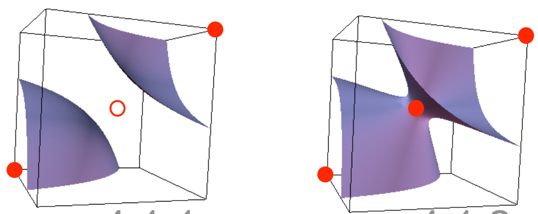


s is positive　　*s is negative*

Center for Computational Visualization
Institute of Computational and Engineering Sciences
Department of Computer Sciences
University of Texas at Austin

Decision on Contour Topology (Natarajan 94)

- Resolving Internal Ambiguity
  - Ambiguity (body saddle)



4.1.1 4.1.2

  - Decision based on the value *s* of saddle point

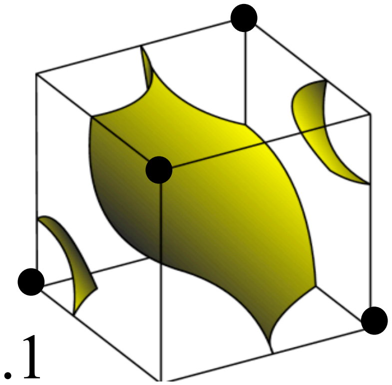
(i) s is positive → tunnel
(ii) s is negative → two pieces

Contour Topology Decision

- Trilinear isosurface connectivity is determined by sign configuration of saddle points and 8 corner vertices of a cube
- Marching Cubes : Consider only 8 corner vertices. Additional Ambiguity problems exist

31 Cases

- In the table,



13.5.1

MC case number

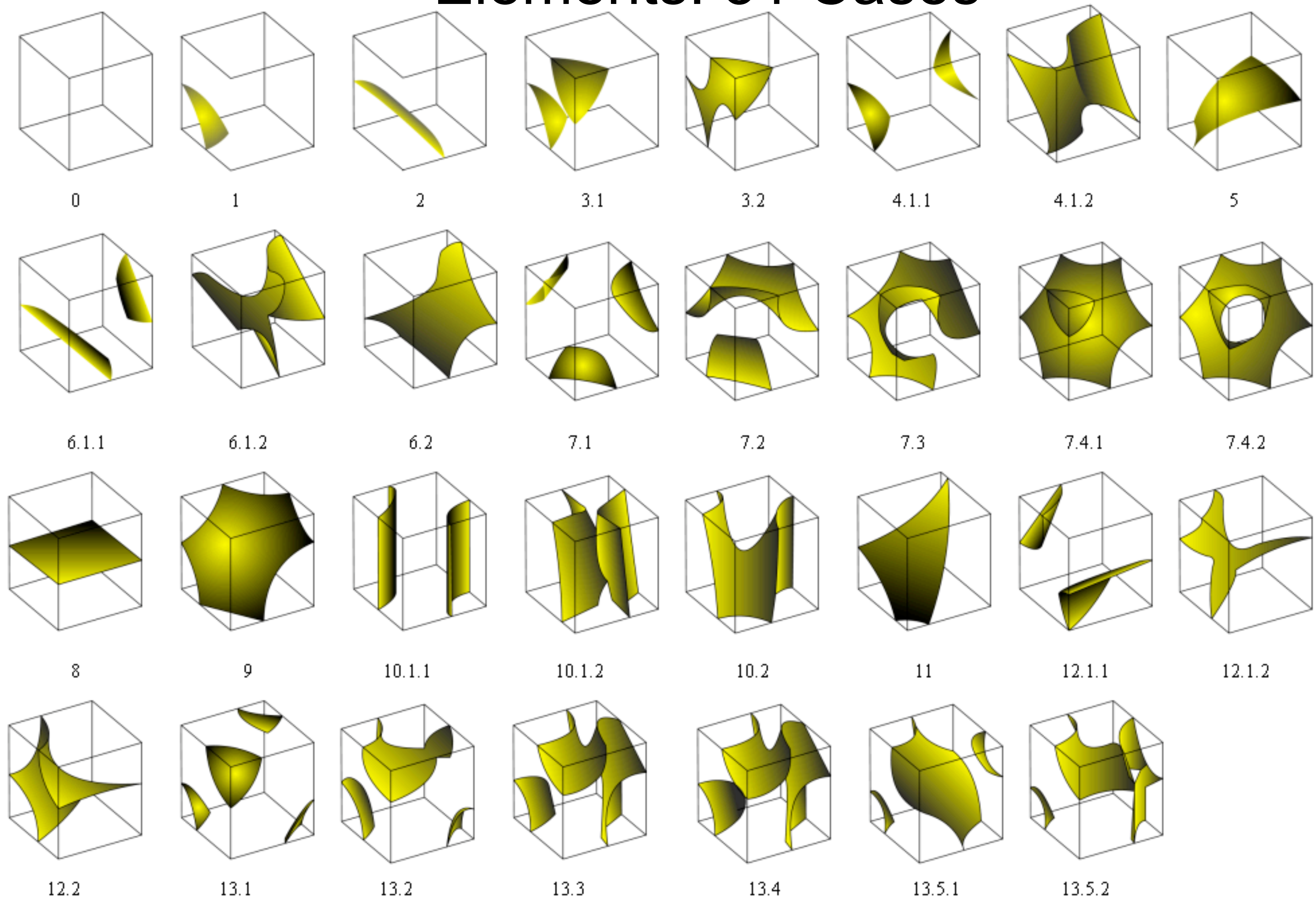
Face ambiguity

Interior ambiguity

- MC : 15(further reduced to 14) cases based on vertex coloring (symm).
- 31 cases ⟵ (vertex coloring , face ambiguity , internal ambiguity)

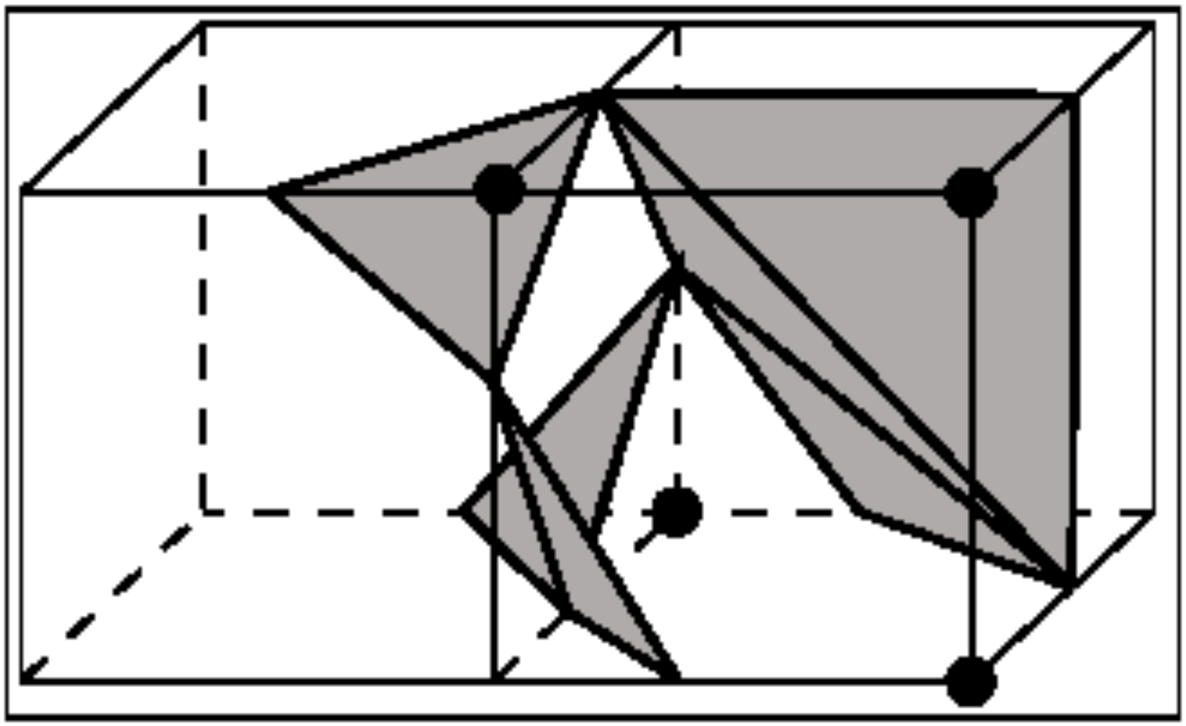
Symmetry of different configurations are used to reduce the cases.

Trilinear Implicit Surface Boundary Elements: 31 Cases



Triangulation Ambiguity

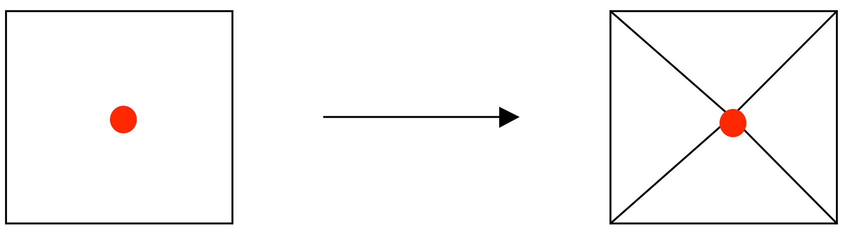
<wrong surface>



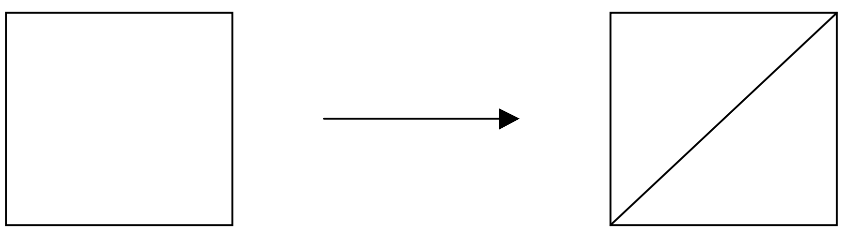
- Saddle points play important roles in determining contour connectivity

Topology Preserving Tetrahedral Decompostion

- 2D case
  - If there is a saddle point

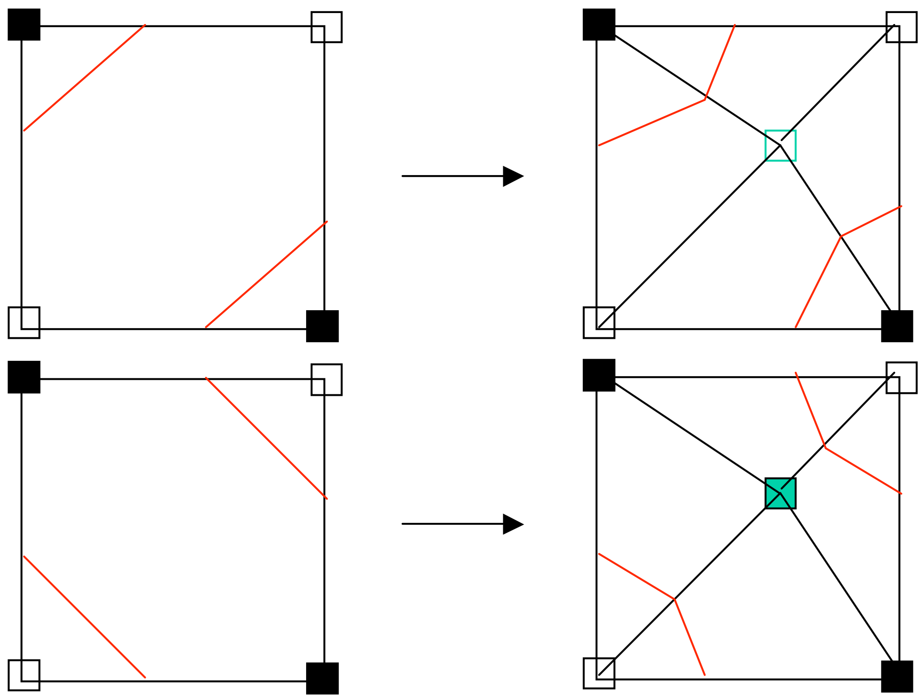


  - If there is no saddle point



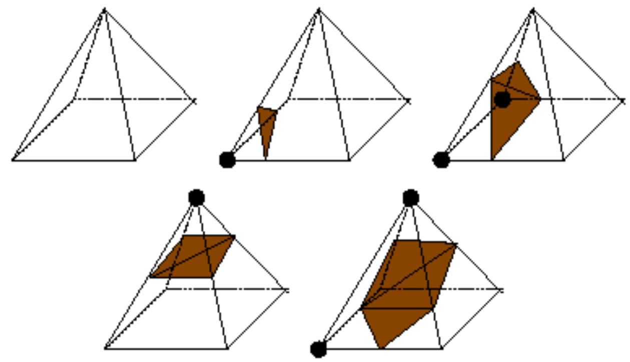
Cell Decomposition Method

- Decompose a cell when a saddle point affects the contour connectivity



Main Decomposition Rule for Trilinear Cell with Topological Ambiguity

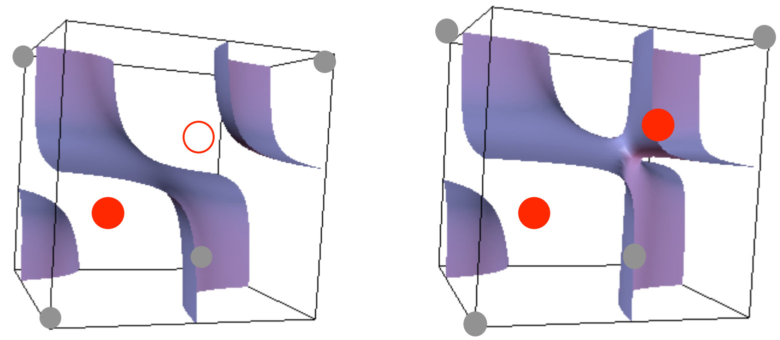
- If isosurface has a tunnel
  - With a body saddle point generate six pyramids with the cube faces
  - Further decompose pyramids that have face ambiguity into four tetrahedra
- If isosurface has no tunnel
  - Choose a face saddle and generate five pyramids with remaining faces
  - Further decompose pyramids that have face ambiguity into four tetrahedra
- Case 13 is an exception

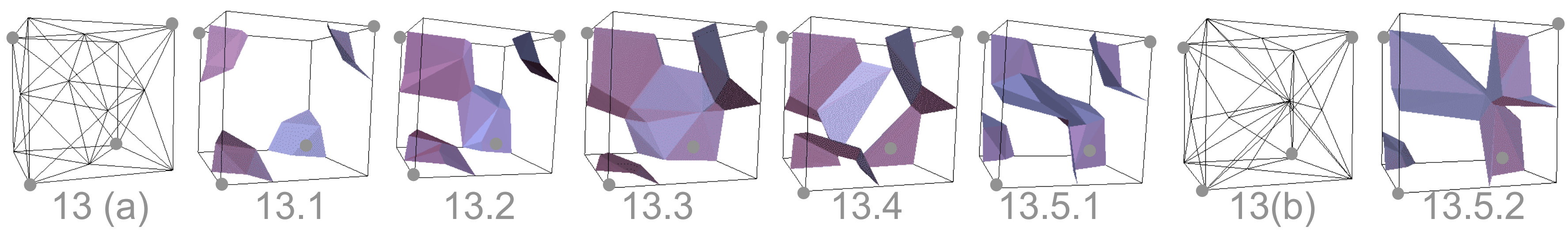


< pyramid triangulation >

Complicated Topology

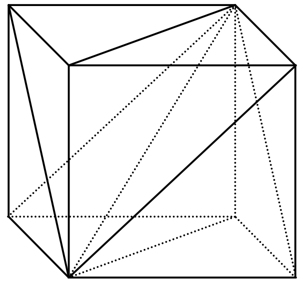
- Case 15 of MC and # 13
  - the most complicated case in geometry and topology
  - involve Face saddles for each face and two body saddles

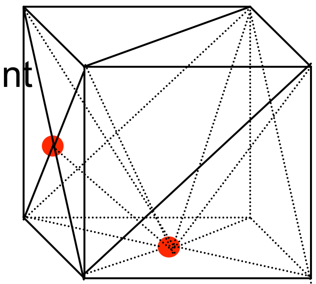


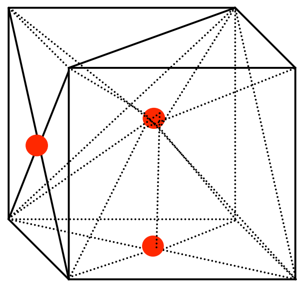


Topology Preserving Tetrahedral Decompostion

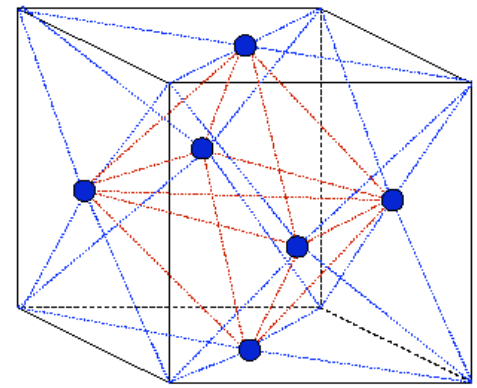
- 3D case (Sb and Sf = # of body and face saddles)
  - (i) Sb = 0 ; Sf = 0
    - Standard decomposition (6 tet)
  - (ii) Sb = 0 & 1 <= Sf <= 4
    - Decompose a face with a face saddle into 4 tris
    - Decompose a face without a face saddle into 2 tris
    - Choose one face saddle and connect it to each face to form 5 pyramids. Each pyramid decomposed into four or two tets (Choice of 2nd largest face saddle point when 3 or 4 face saddles present)
  - (iii) Sb = 1 & 1 <= Sf <= 4
    - Connect a body saddle to each face to form 6 pyramids
    - Each pyramid decomposed into four or two tets, depending on presence or absence of face saddles







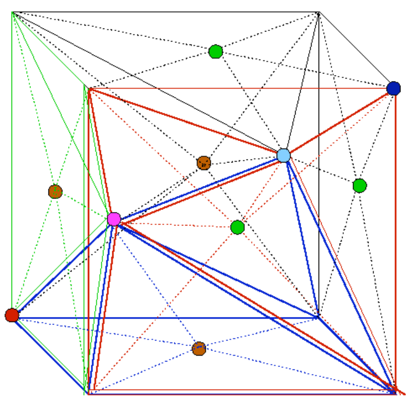
Topology Preserving Tetrahedral Decompostion (#13)

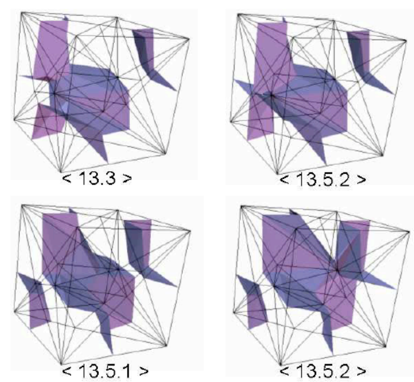


- (iv) Sb = 0 & Sf = 6

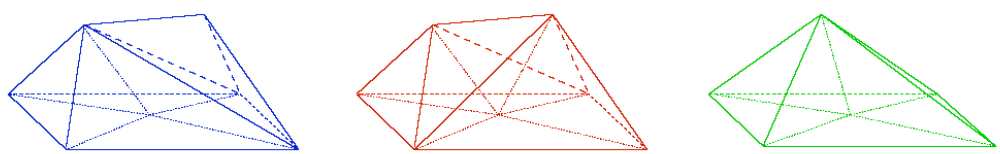
  24 tetrahedral split

- (v) 1 <=Sb <=2 & Sf = 6





Decompose into pyramids & prisms and then further split into tetrahedra

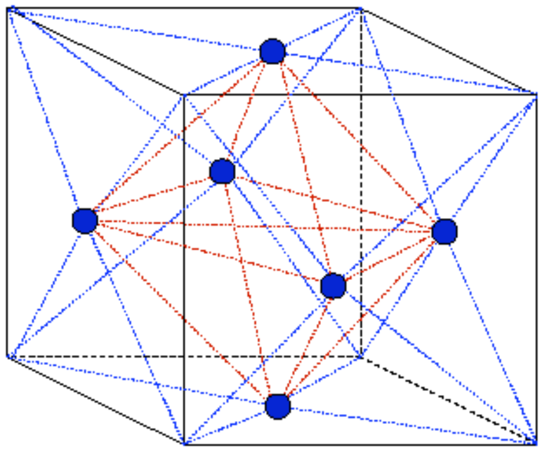


Center for Computational Visualization
Institute of Computational and Engineering Sciences
Department of Computer Sciences
University of Texas at Austin

#13 Case 4: Sb = 0 & Sf = 6

- Connect the 6 face saddles forming an 8 triangular-facet diamond, which is split into 4 tetrahedra
- 12 tetrahedra are created by joining 2 vertices of an edge with 2 face saddles of the faces incident at the edge
- 8 additional tetrahedra are created by connecting each facet of the diamond to the 8 vertices of the cube

Overall 24 tetrahedral split



Case 4: 1 <=Sb<=2 & Sf = 6

Order the saddle points in increasing order of saddle values and into 3 small face saddles, small body saddle, big body saddle, and 3 big face saddles

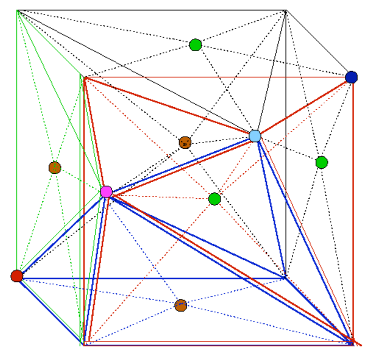
Small/Big corner vertex is a vertex adjacent to the three faces containing small/big face saddles

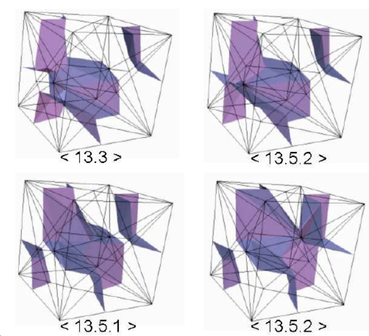
– Sb = 2

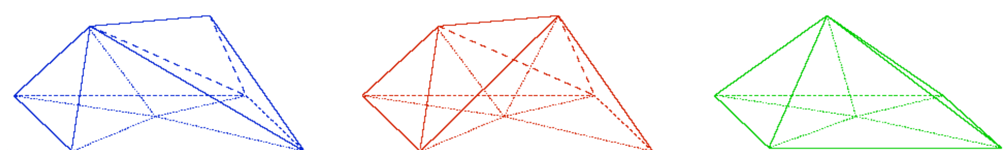
Decompose into 2 Pyramids and 4 Trunc-prisms.

– Sb = 1

Decompose into 1 Pyramid and 4-Trunc prsims

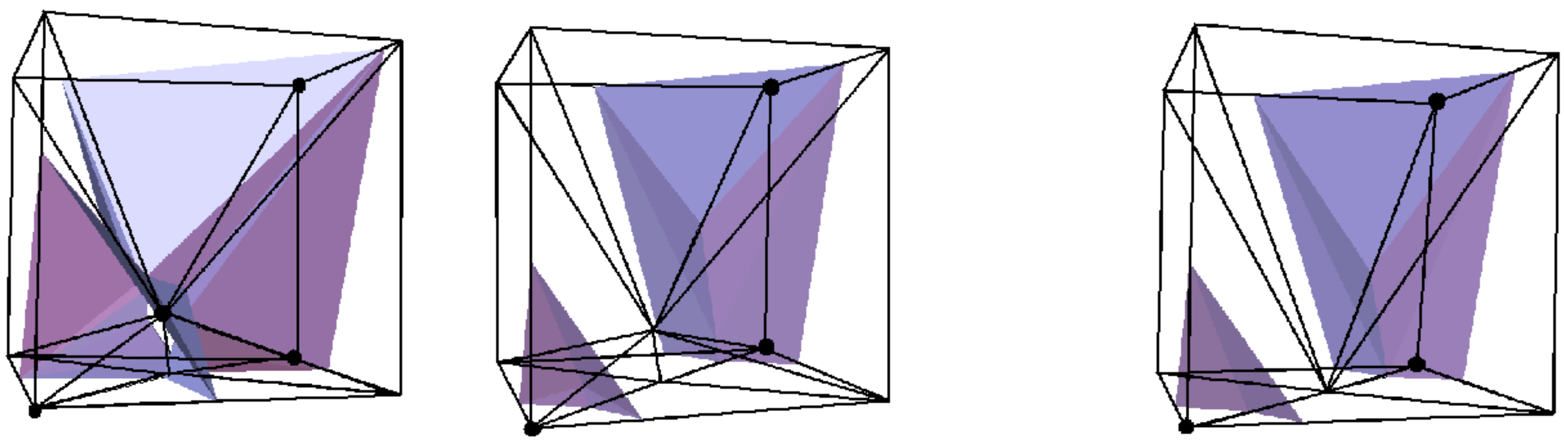






Cell Decomposition Method

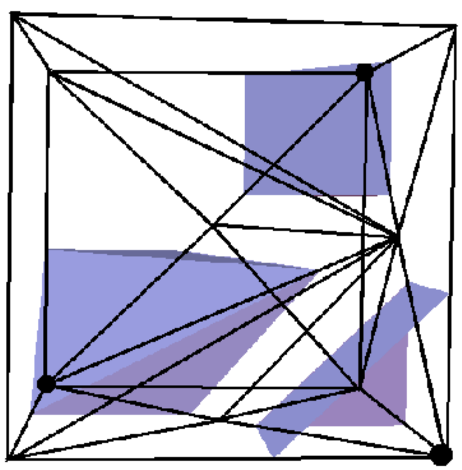
- Disambiguate internal topology



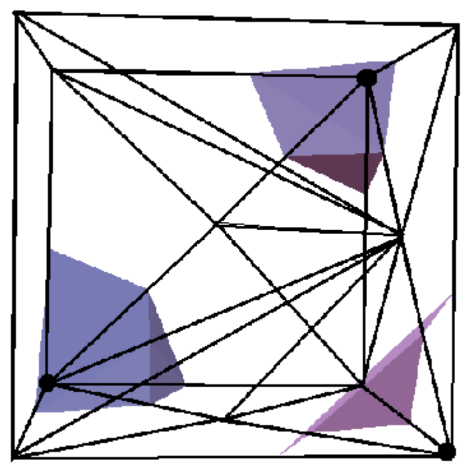
Body saddle can be ignored when no tunnel exists

Geometric Improvement

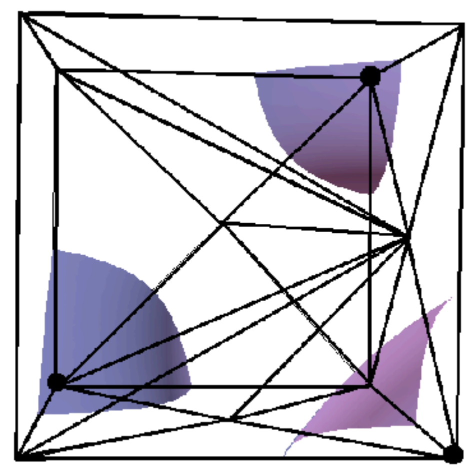
- Compute true intersection between an edge and isosurface



<linear interpolation Along edge >

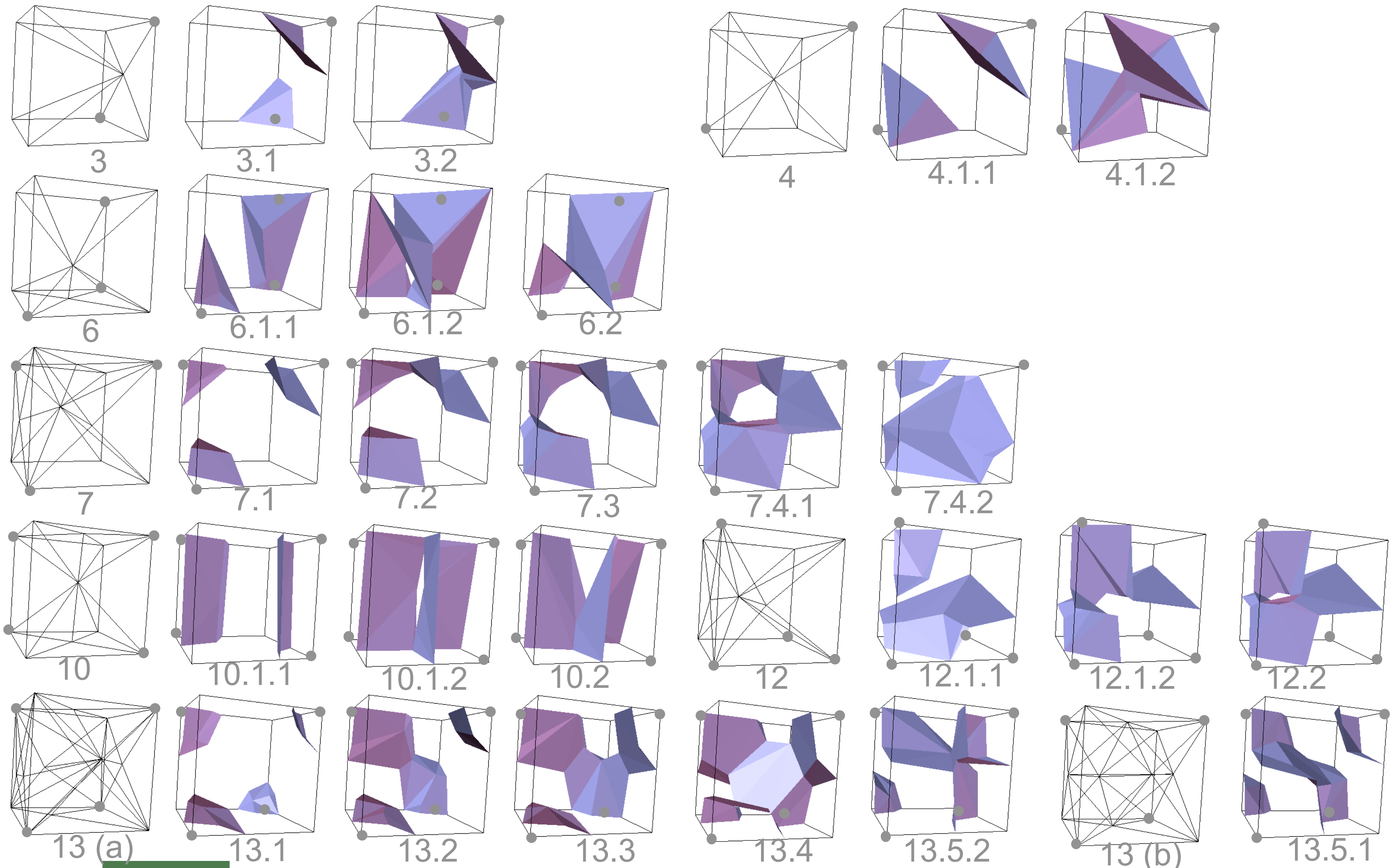


<intersection between edge and true isosurface >



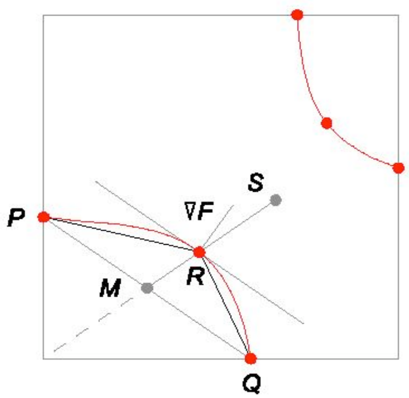
<trilinear isosurface >

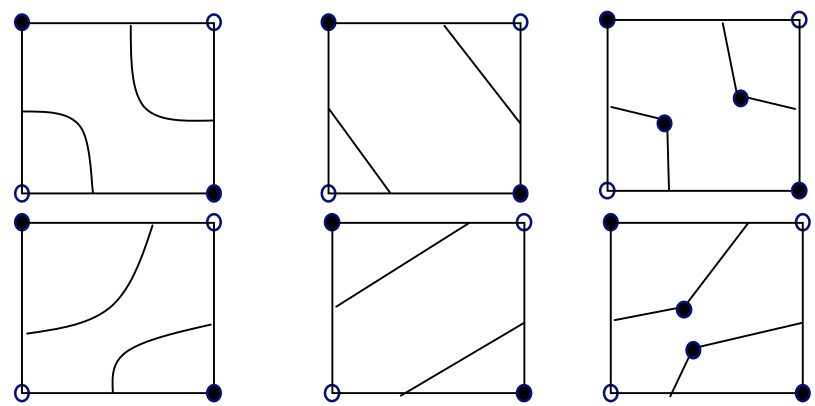
Results

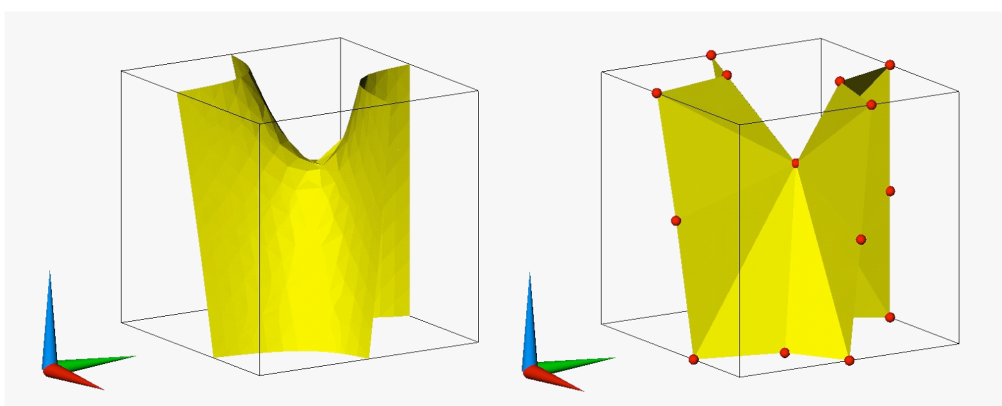

3
3.1
3.2
4
4.1.1
4.1.2
6
6.1.1
6.1.2
6.2
7
7.1
7.2
7.3
7.4.1
7.4.2
10
10.1.1
10.1.2
10.2
12
12.1.1
12.1.2
12.2
13 (a)
13.1
13.2
13.3
13.4
13.5.2
13 (b)
13.5.1

Geometric Approximations

- Better appoximation of trilinear interpolant
  - Adding a shoulder and inflection points

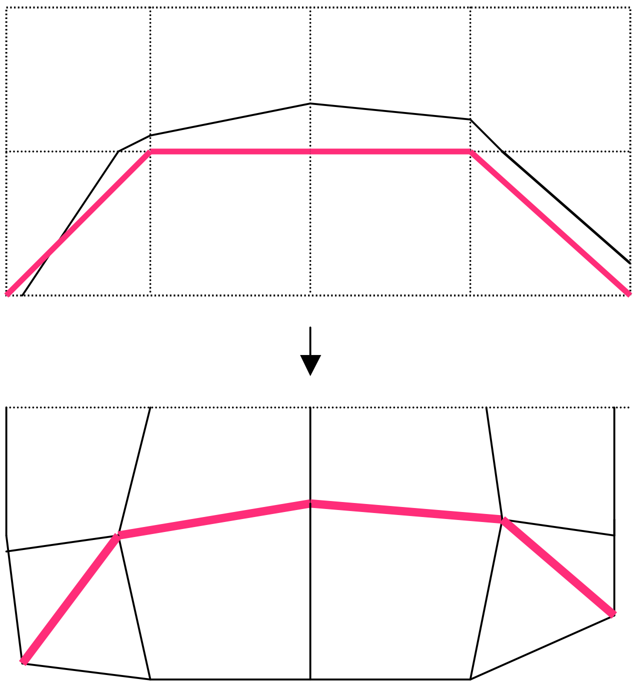




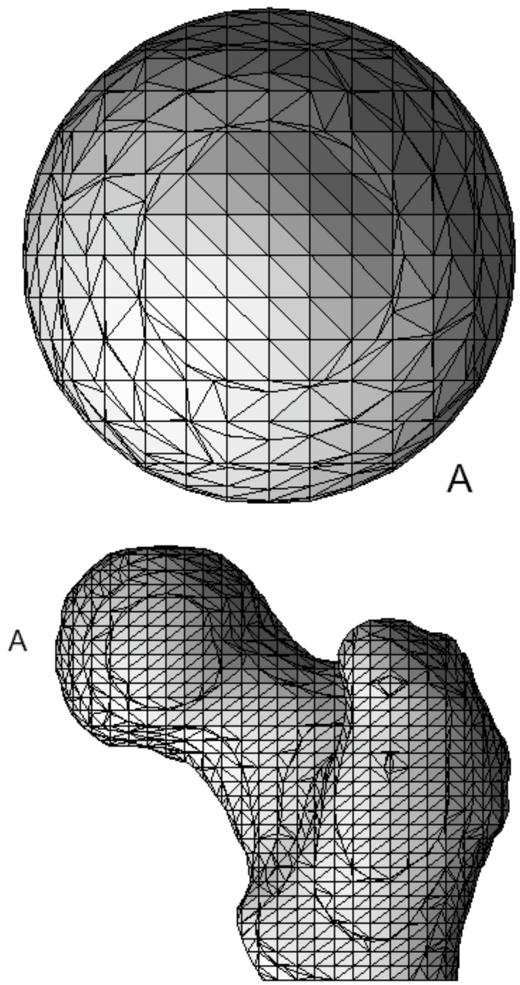


Mesh Displacement

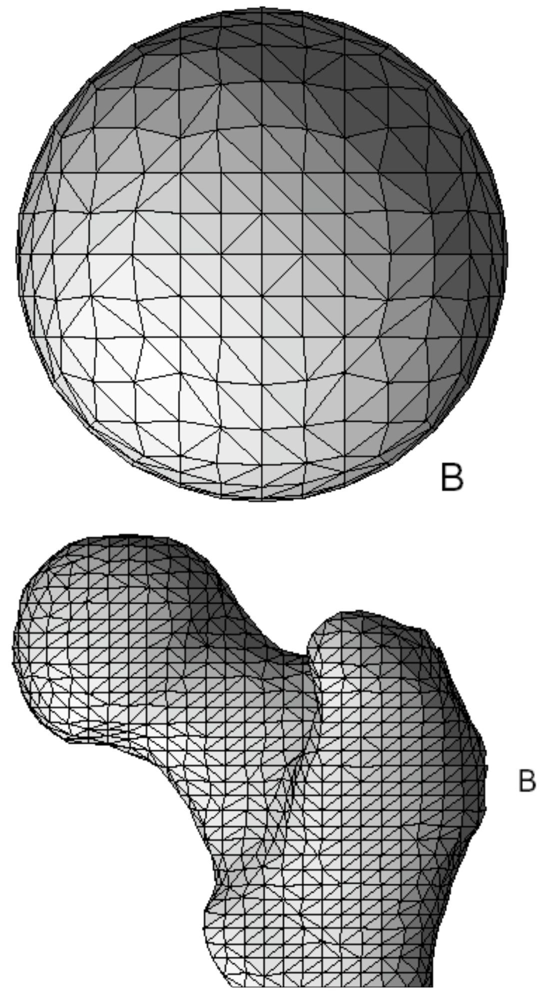
– Remove small triangles + good aspect ratio



Mesh Displacement



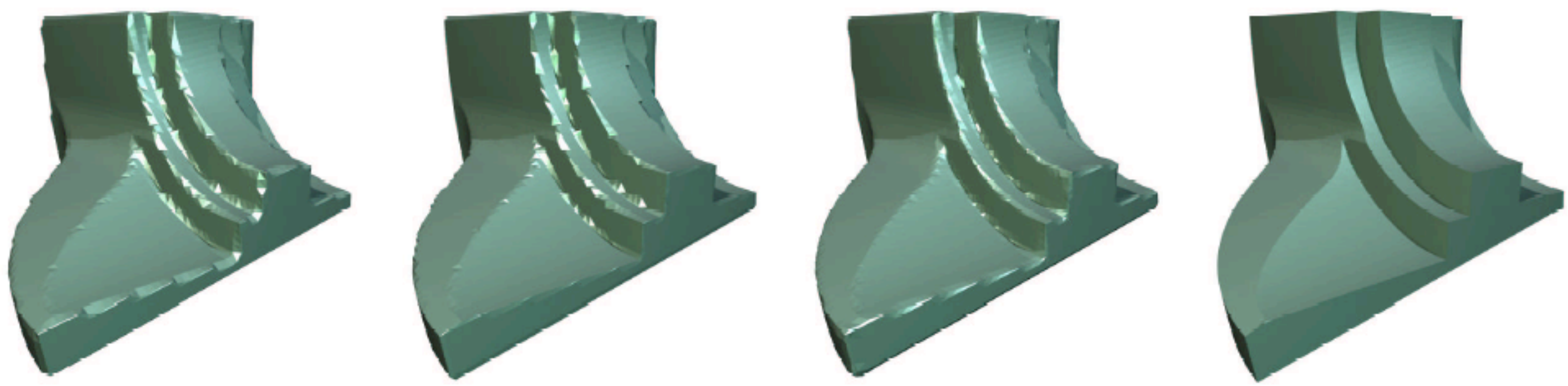
Marching Cubes

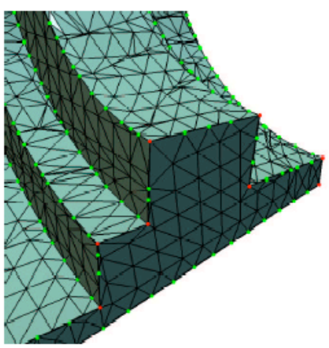


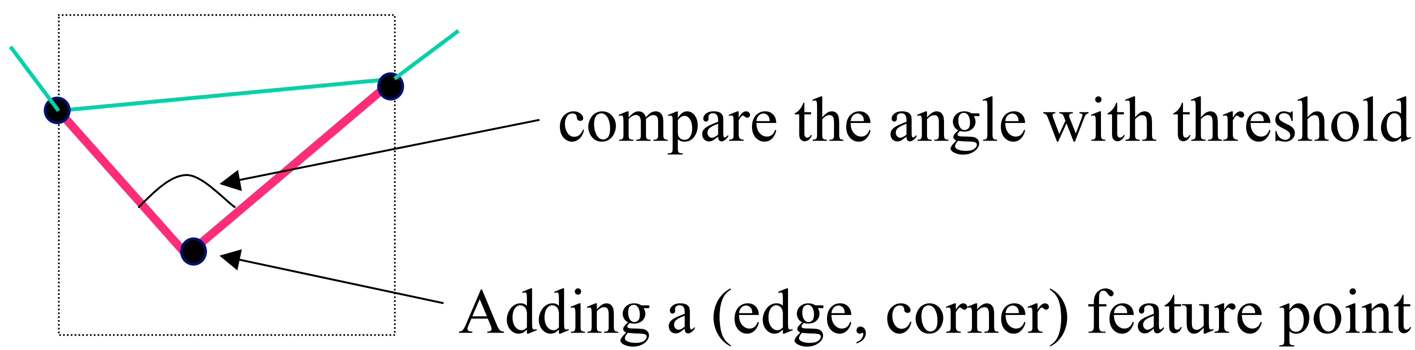
Mesh Displacement

Feature Sensitive Surface Extraction

- Extended Marching Cubes

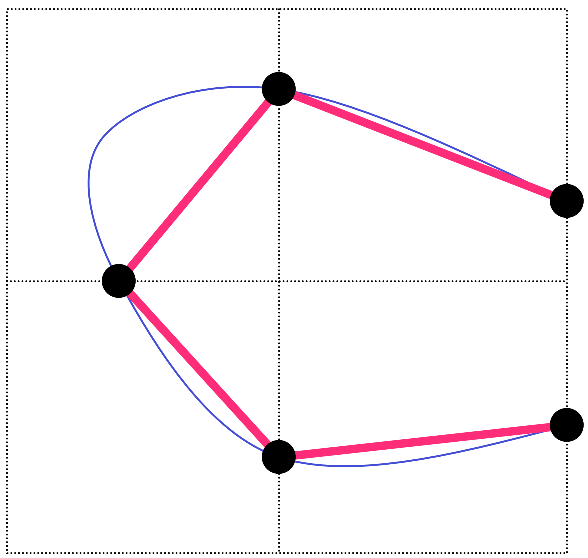




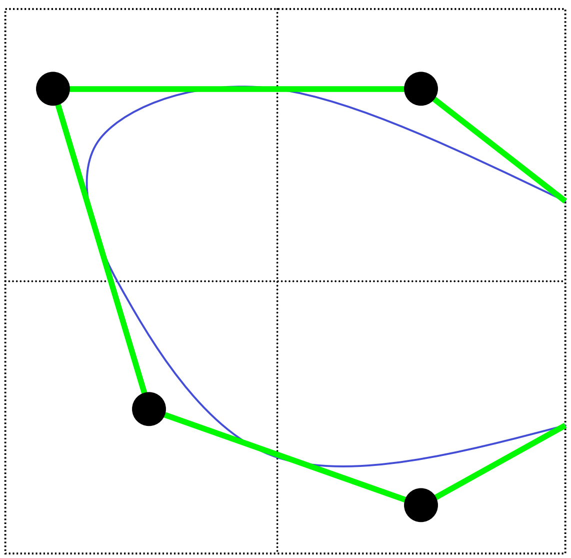


Dual Contouring

- Primal Contouring vs Dual Contouring



Primal contour

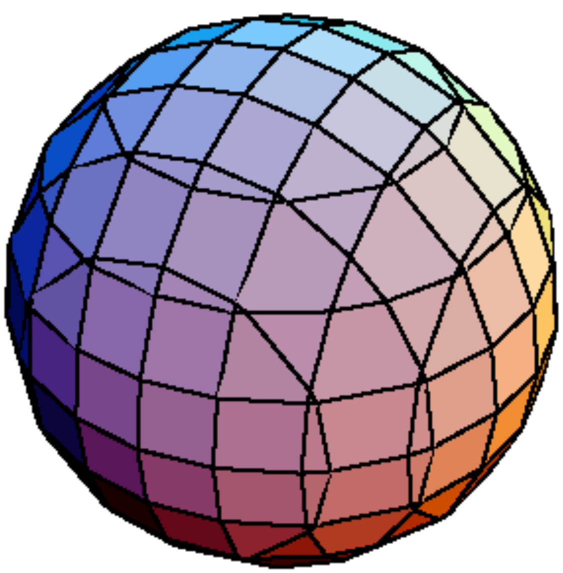


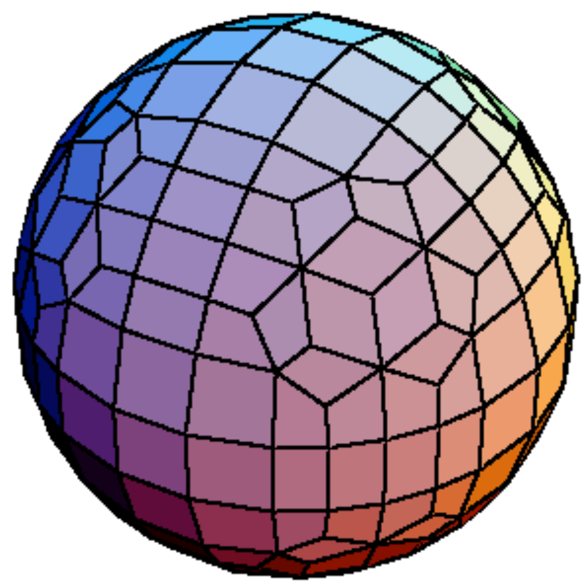
Dual Contour

Center for Computational Visualization
Institute of Computational and Engineering Sciences
Department of Computer Sciences
University of Texas at Austin

Dual Contouring

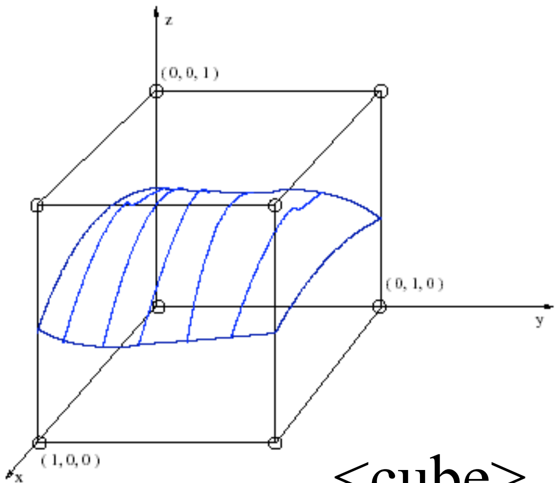
- Polygons with better aspect ratio



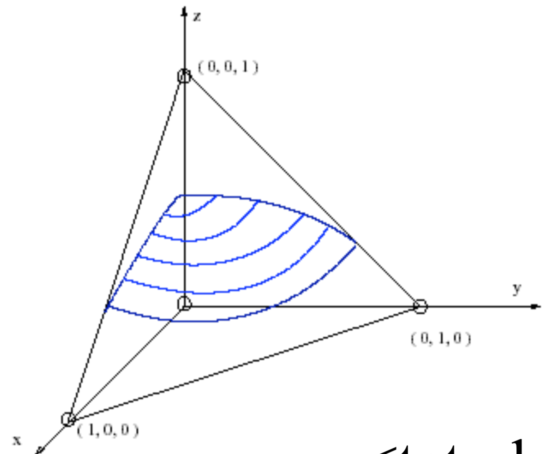


Algebraic Patches: Smooth Boundary Elements

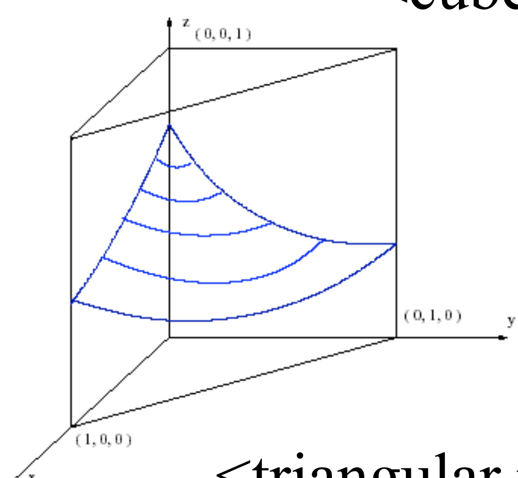
- Implicit form of Isocontour : $f(x,y,z) = w$



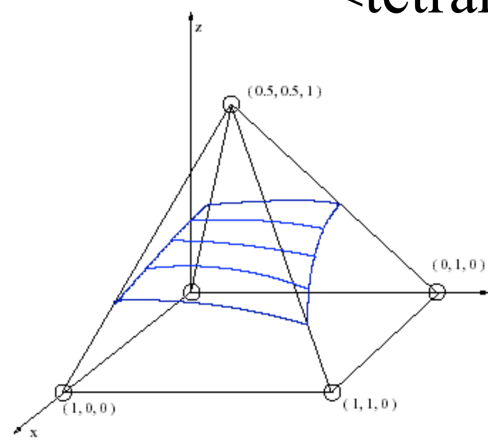
<cube>



<tetrahedron>



<triangular prism>

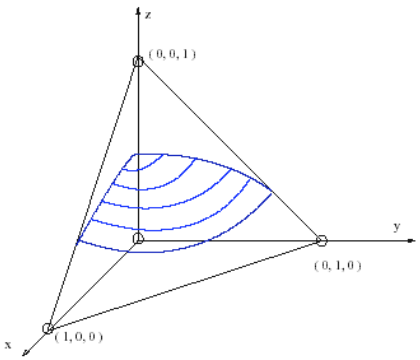


<square pyramid>

A-Patches

- Given tetrahedron vertices $p_i=(x_i,y_i,z_i)$, $i=1,2,3,4$, α is barycentric coordinates of $p=(x,y,z)$:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix}$$

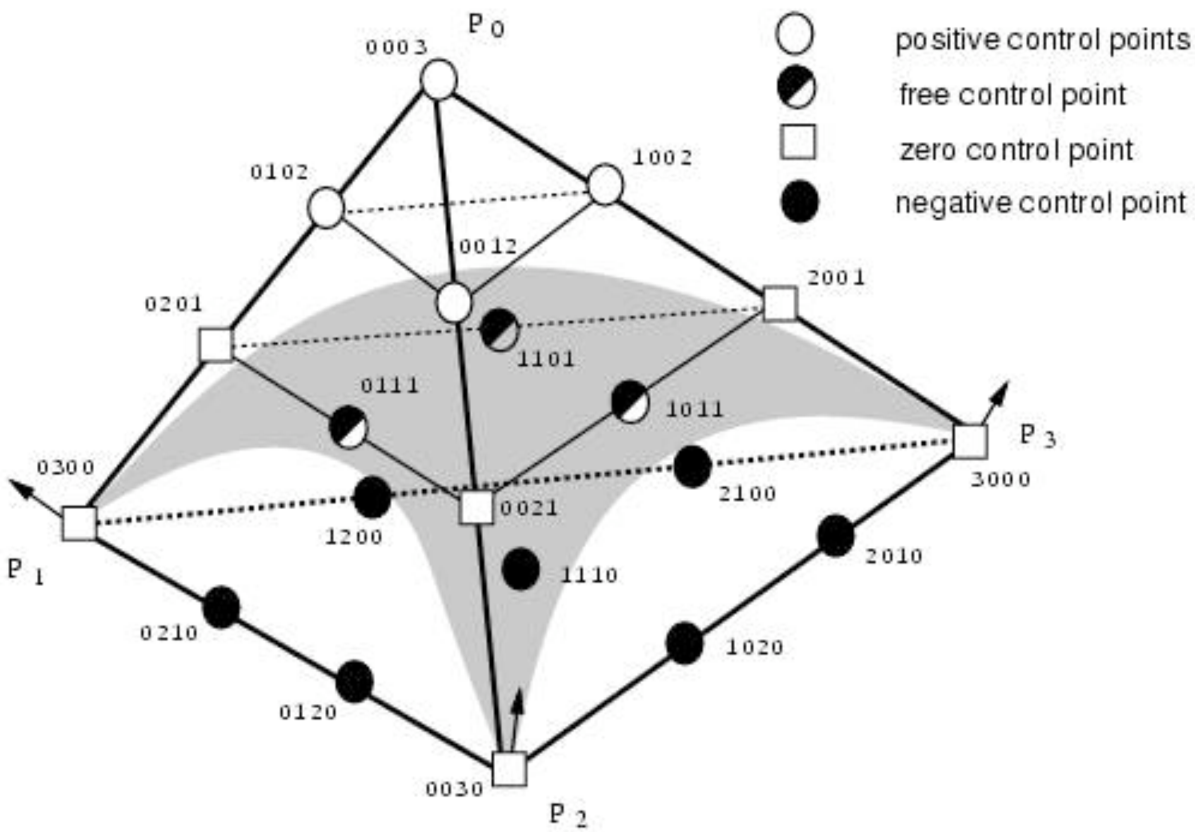


- function $f(p)$ of degree n can be expressed in Bernstein-Bezier form :

$$f(p) = \sum_{|\lambda|=n} b_\lambda \, B^n_\lambda(\alpha), \; \lambda \in \mathcal{Z}^4_+ \qquad B^n_\lambda(\alpha) = \frac{n!}{\lambda_1!\lambda_2!\lambda_3!\lambda_4!} \, \alpha_1^{\lambda_1}\alpha_2^{\lambda_2}\alpha_3^{\lambda_3}\alpha_4^{\lambda_4}$$

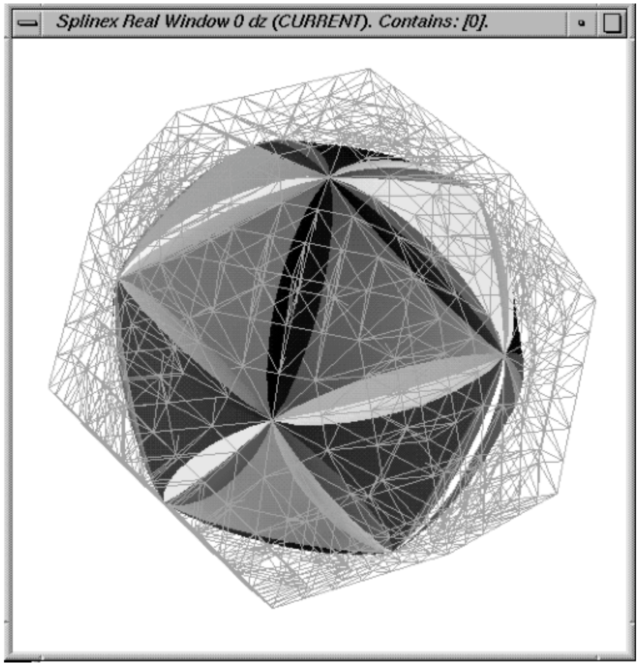
- Algebraic surface patch(A-patch) within the tet is defined as $f(p)=0$.

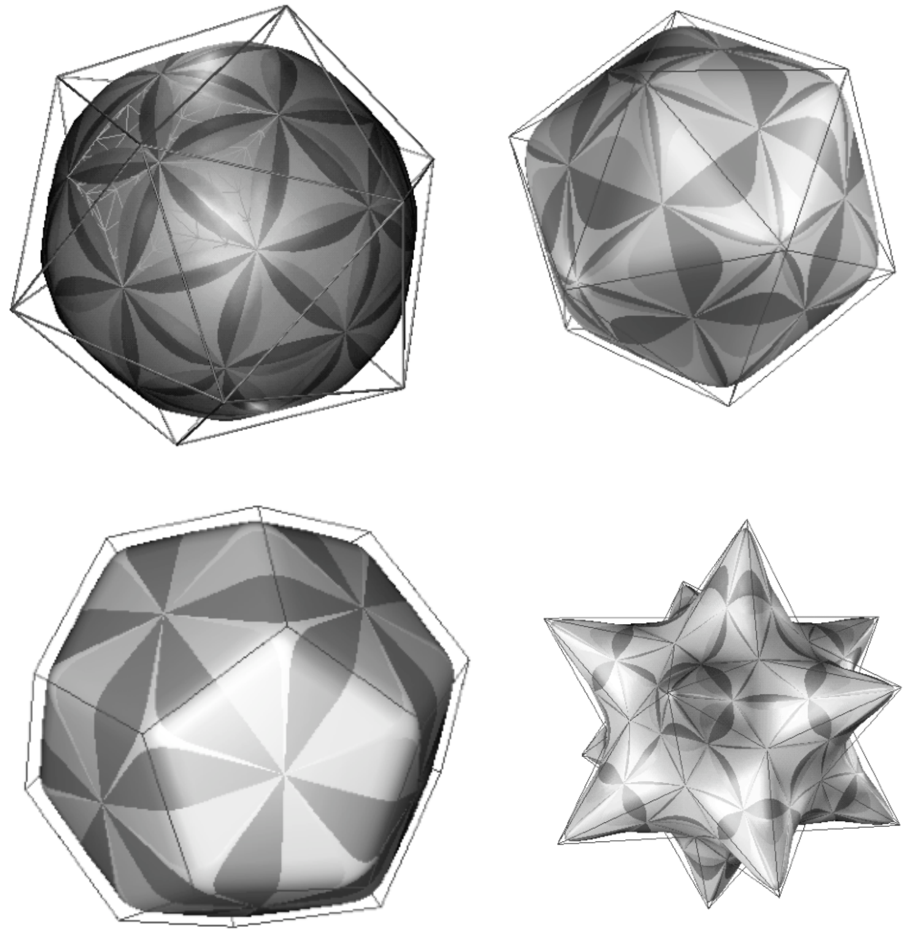
A-patch Surface (C^1) Interpolant

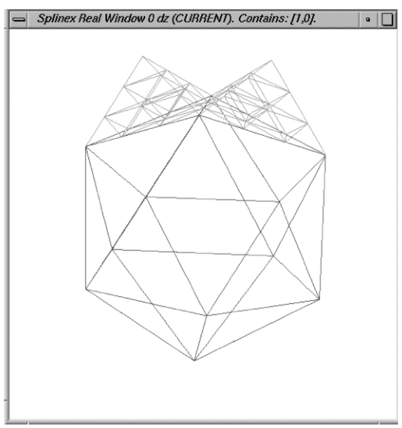


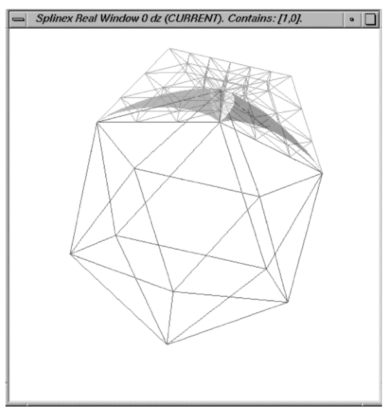
- An implicit single-sheeted interpolant over a tetrahedron

A-patch Contouring

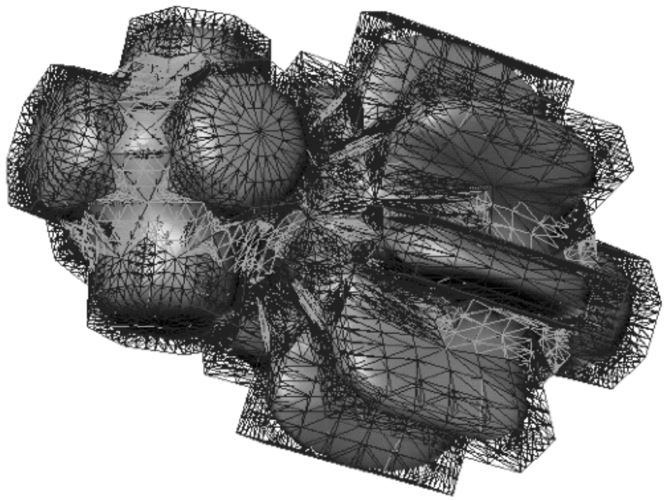


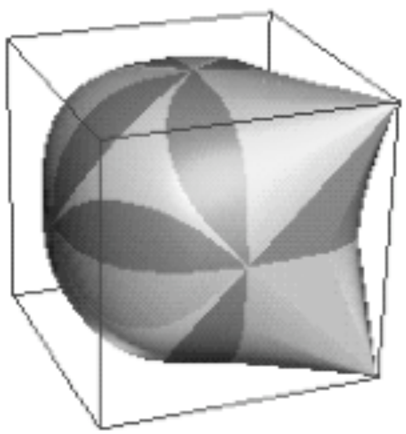


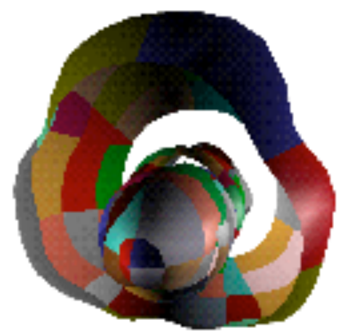


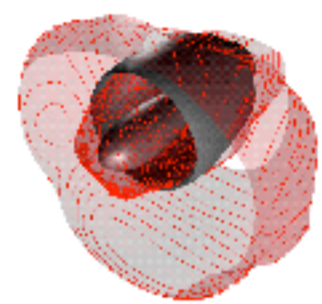


A-patch Contouring

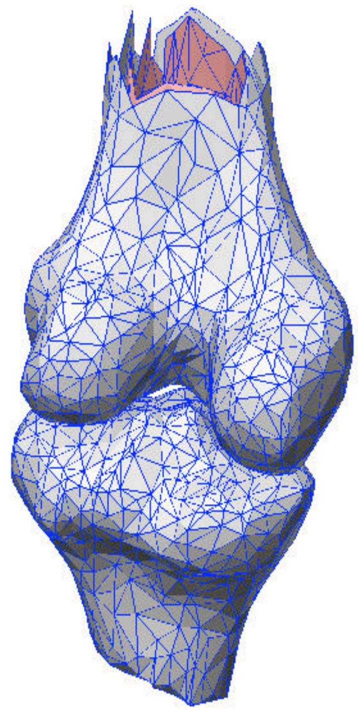




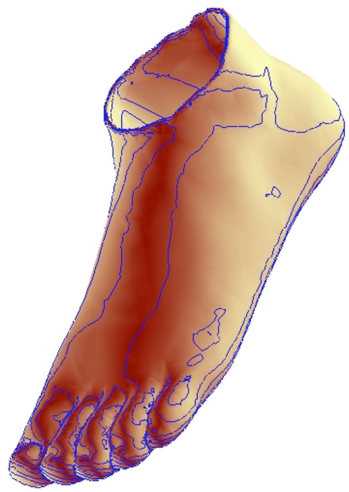


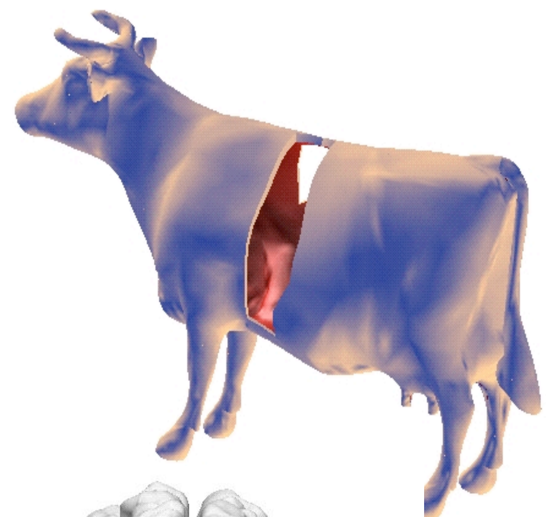


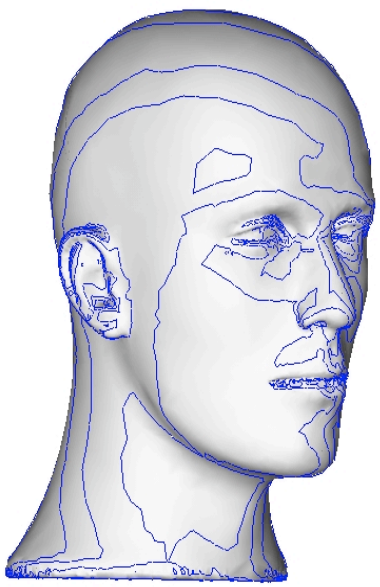
Finite Elements from Images

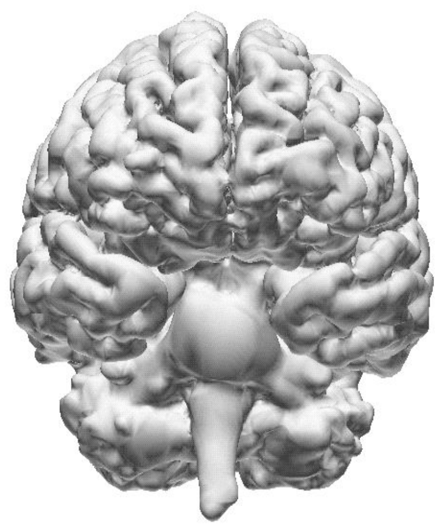


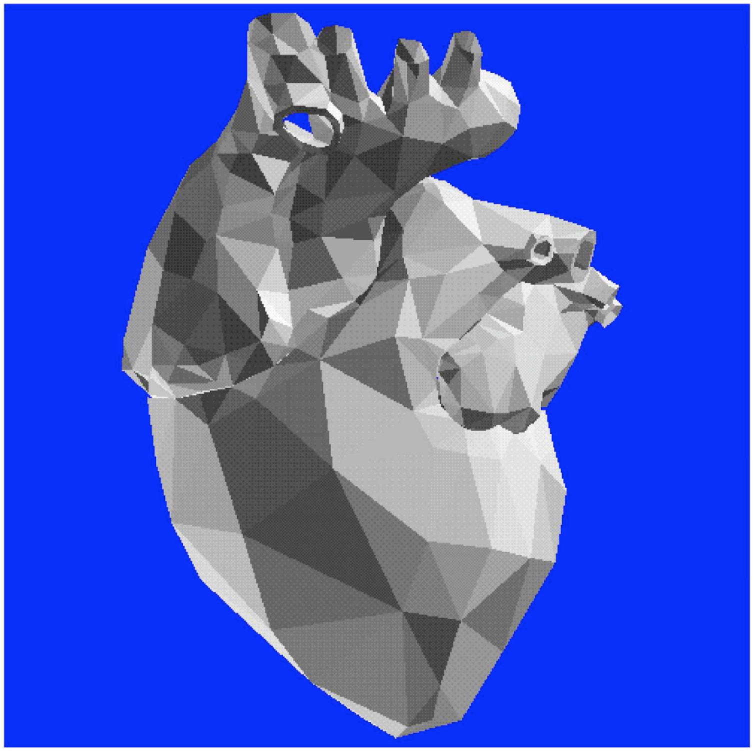
Examples with Shell Finite Elements

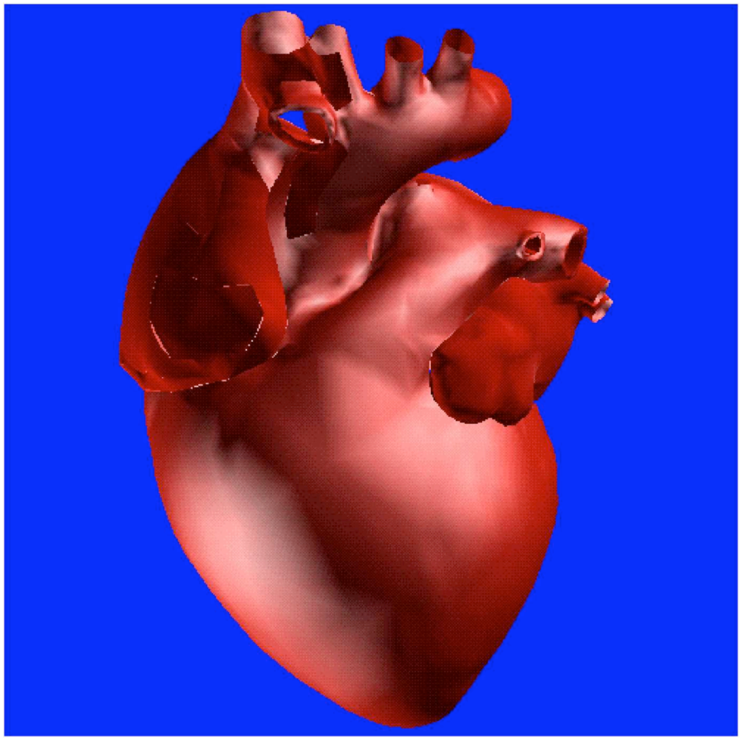




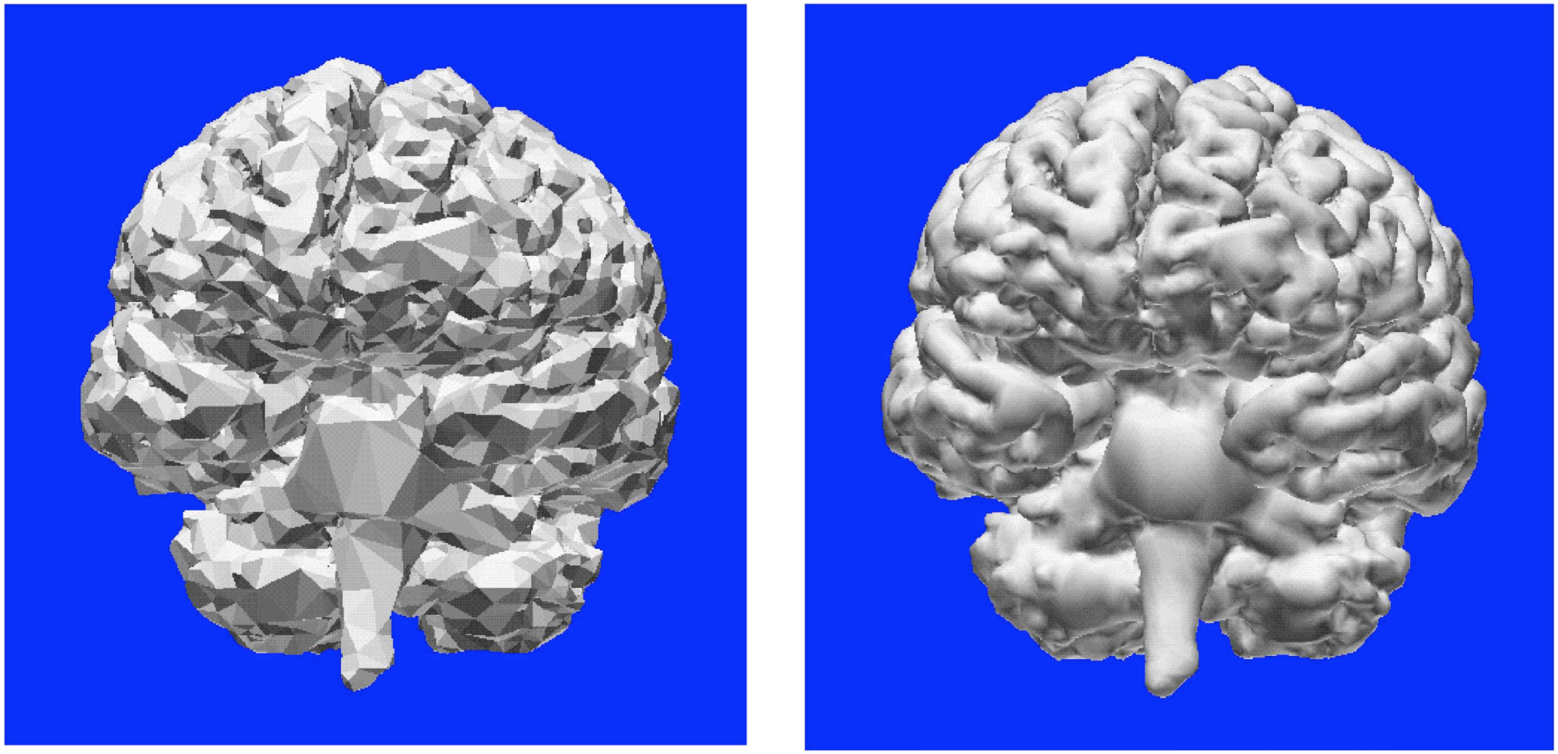




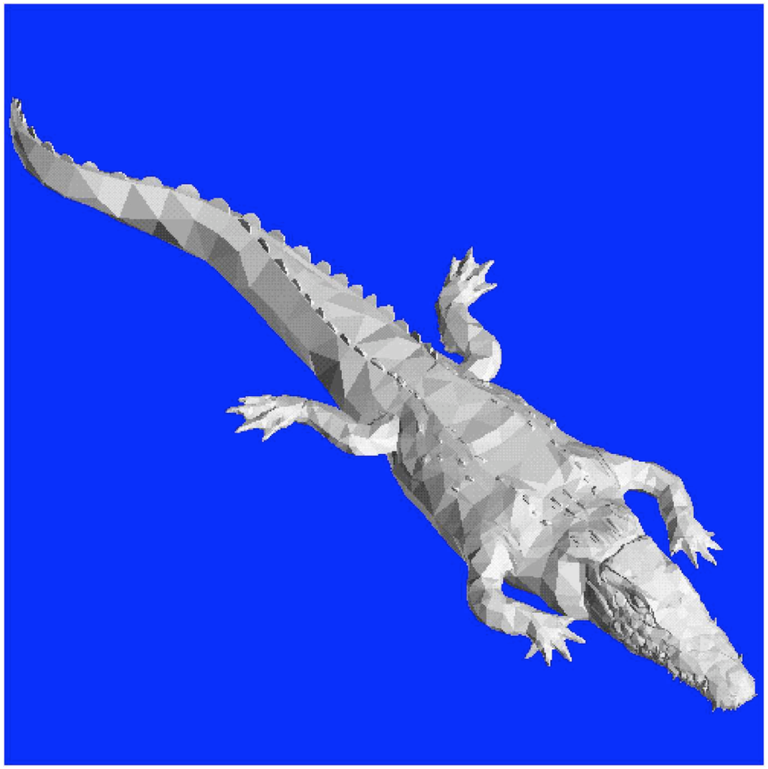


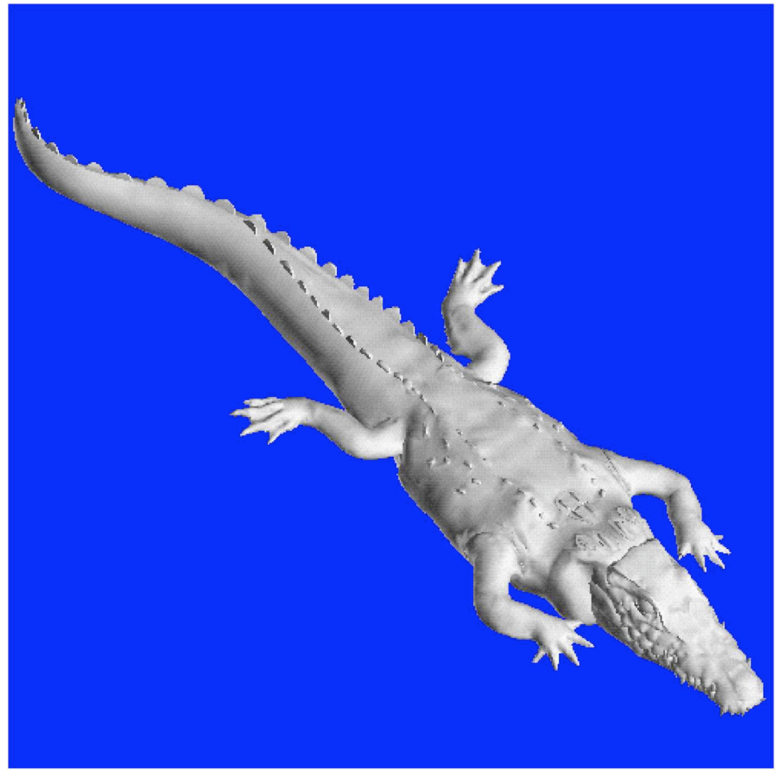


Adaptive feature of the reconstruction: The flat parts use less patches than the curved parts



Adaptive feature of the reconstruction: The flat parts use less patches than the curved parts



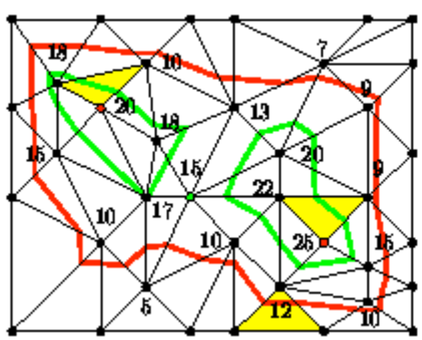


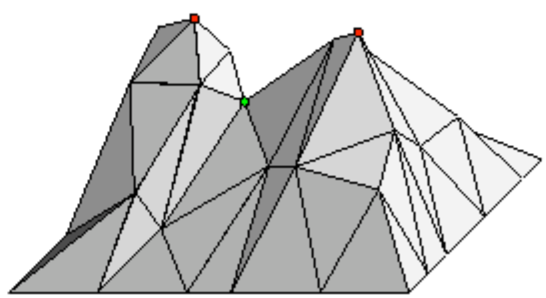
Capturing detail structures.

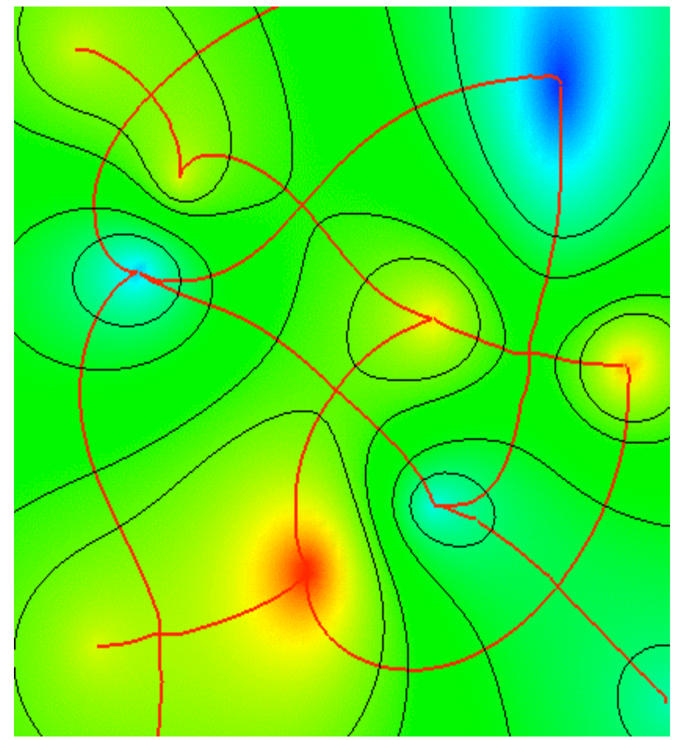
Interactive Isocontour Queries

– Input:
  - Scalar Field *F* defined on a mesh
  - Multiple Isovalues *w* in unpredictable order

– Output (for each isovalue *w*):
  Contour *C(w)* = {*x* | **F**(*x*) = *w*}







Center for Computational Visualization
Institute of Computational and Engineering Sciences
Department of Computer Sciences
University of Texas at Austin

Related Work

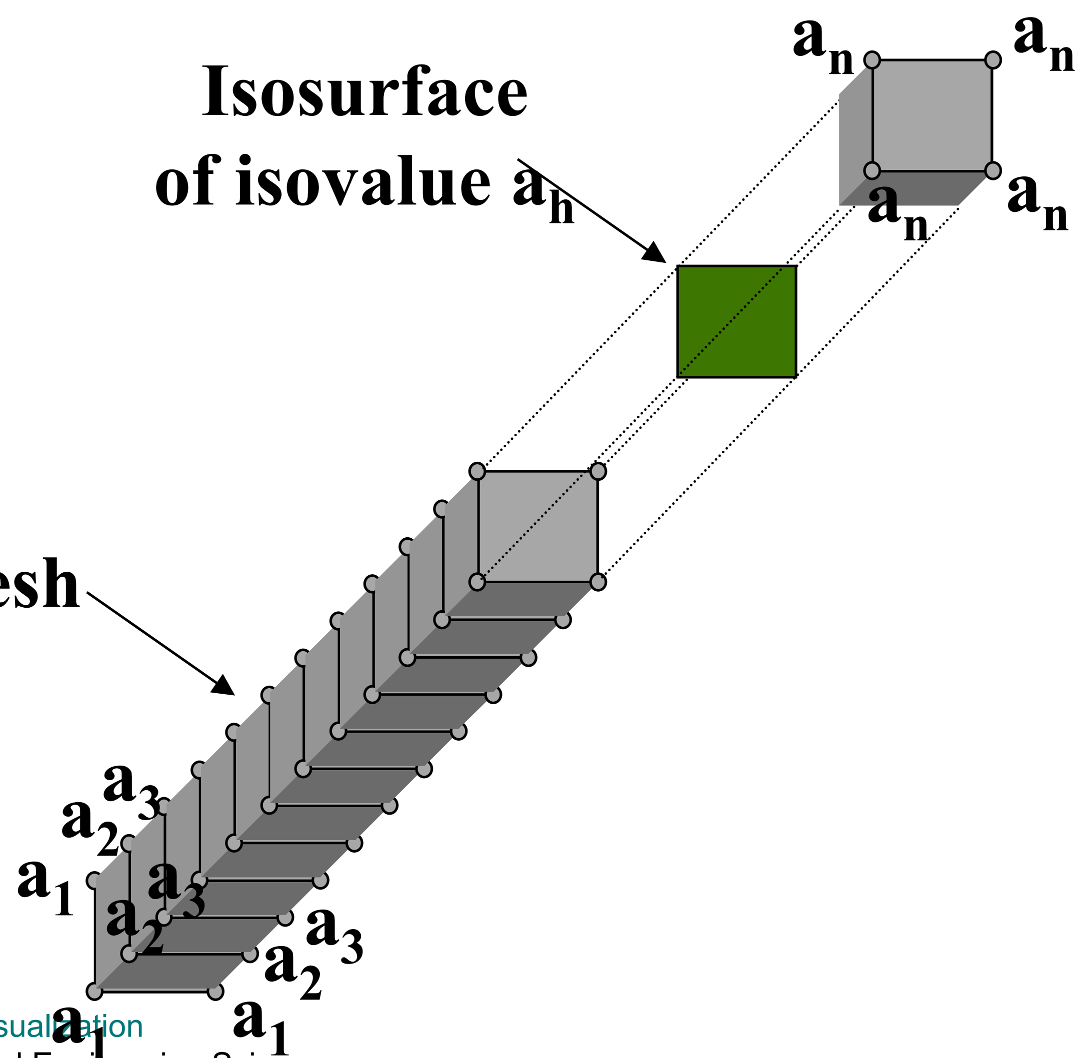
		Search Space	
		Geometric	**Value**
Contouring Strategy	Cell by Cell	Lorenson/Cline (Marching Cubes) Wilhelms/Van Gelder (octree)	Giles/Haimes (min-sorted ranges) Shen/Livnat/Johnson/Hansen (LxL lattice) Gallagher(span decomposed into backets) Shen/Johnson (hierachical min-max ranges) Cignoni/Montani/Puppo/Scopigno Livnat/Shen/Johnson (kd-tree)
	Mesh Propagation	Howie/Blake(propagation) Itoh/Koyamada (extrema graph) Itoh/Yamaguchi/Koyamada (volume thinnig)	van Kreveld Bajaj/Pascucci/Schikore van Kreveld /van Oostrum/Bajaj/ Pascucci/Schikore

Isocontour Query Problem

Lower Bound

Input size n } $m+\log(n)$

Output size m

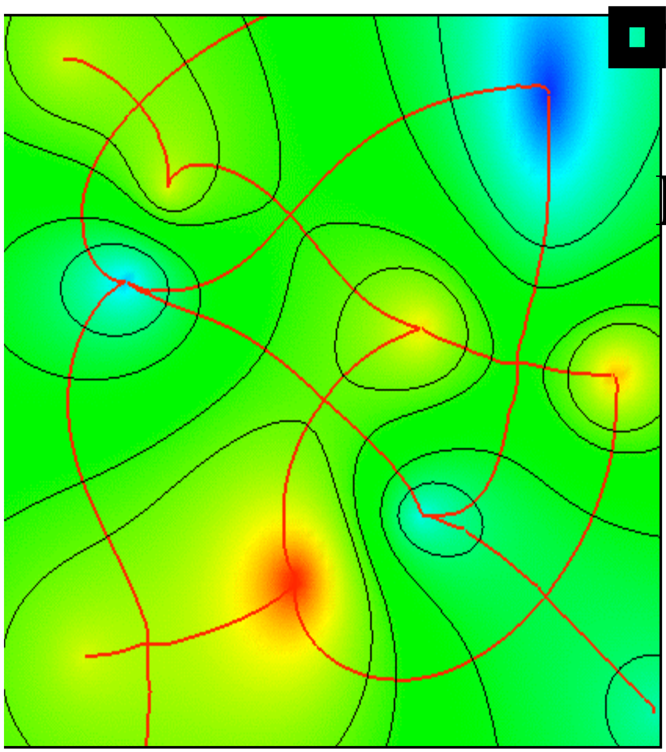


The search for a_h takes at least $\log(n)$

$a_1, a_2, a_3, a_4, \ldots, a_n$

Optimal Single-Resolution Isocontouring

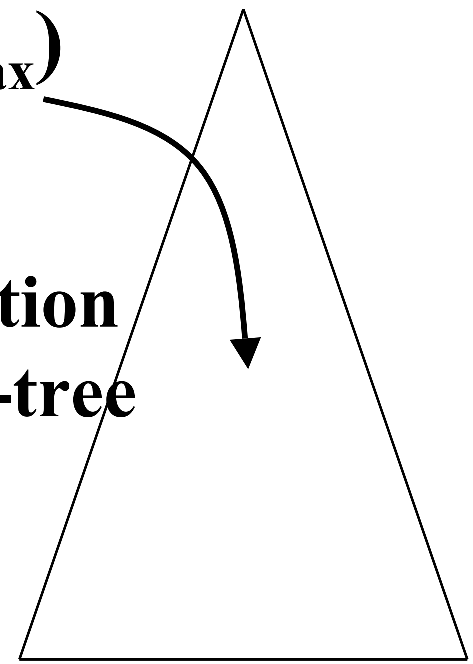
The basic scheme



Preprocessing: (f_{min}, f_{max})

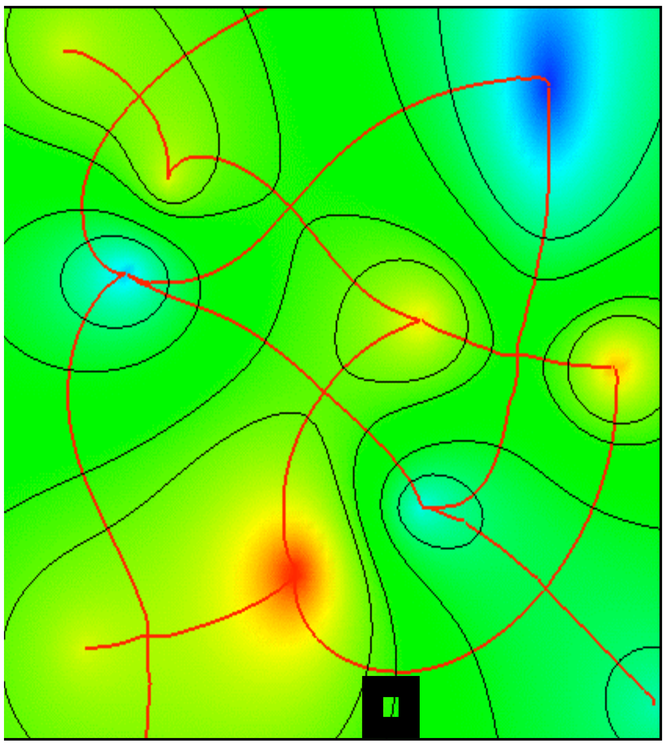
For each cell c in *M*

Enter its range of function values into an interval-tree



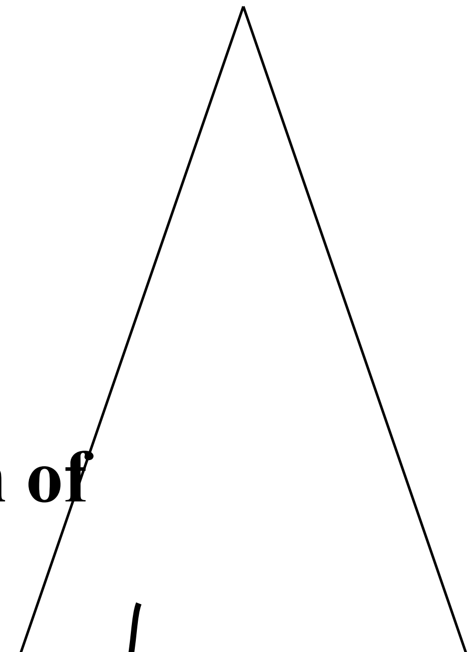
Optimal Single-Resolution Isocontouring

<u>The basic scheme</u>



Isocontour query *W*

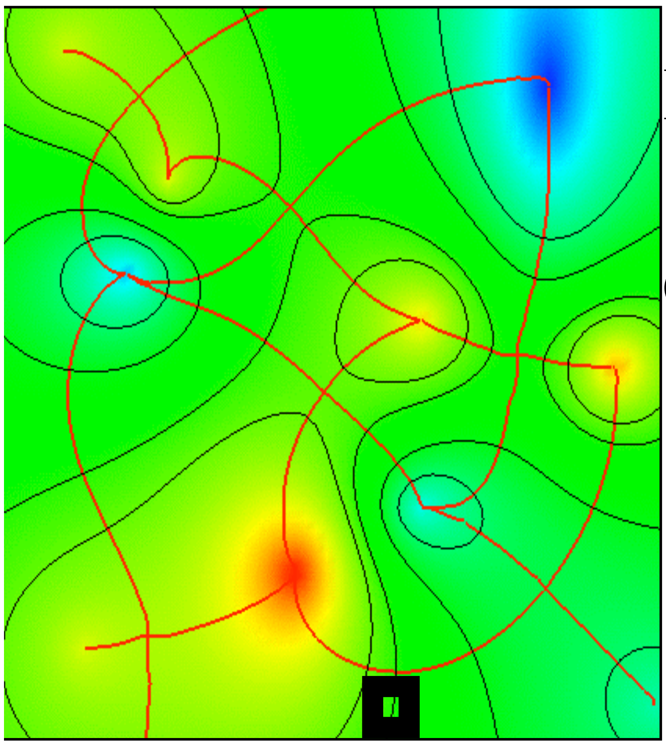
For each interval containing *W*
Compute the portion of isocontour in the corresponding cell



(f_{min}, f_{max})

Optimal Single-Resolution Isocontouring

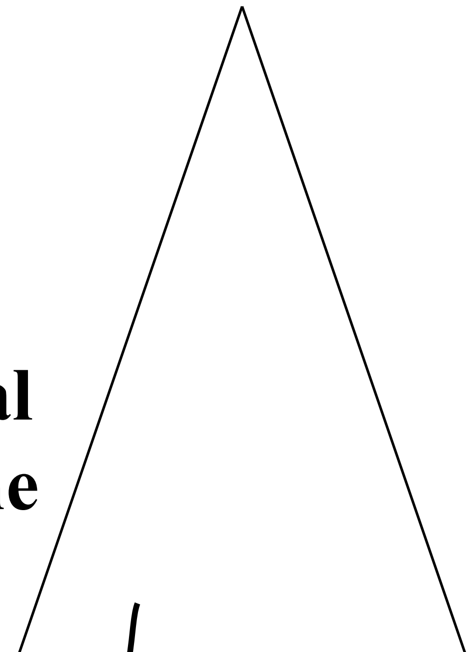
The basic scheme



Isocontour query *W*

Complexity: $m + \log(n)$

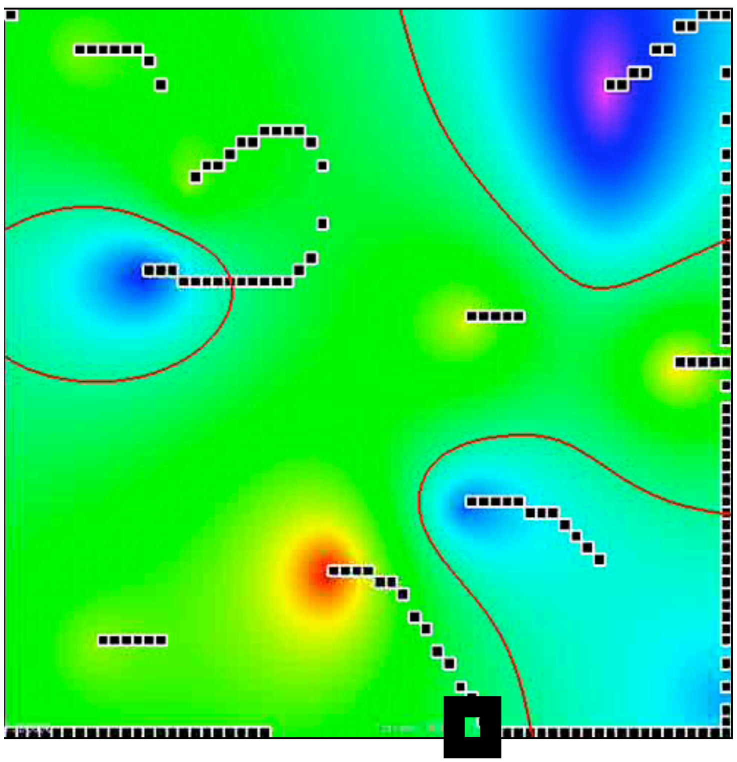
Optimal but impractical because of the size of the interval-tree



(f_{min}, f_{max})

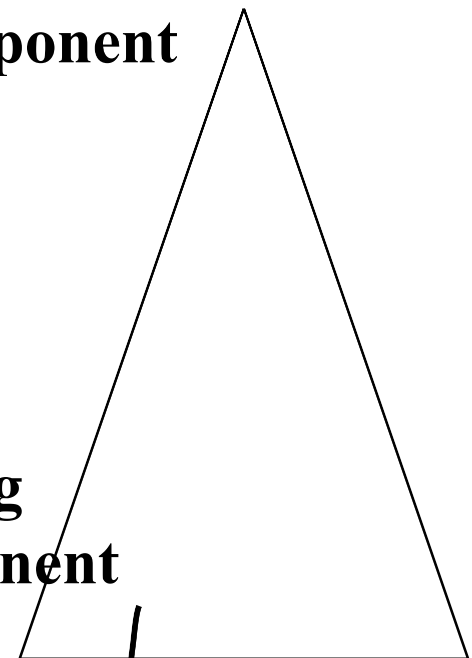
Optimal Single-Resolution Isocontouring

Seed Set Optimization



For each connected component we need only one cell (and then propagate by adjacency in the mesh)

Seed Set: a set of cells intersecting every connected component of every isocontour

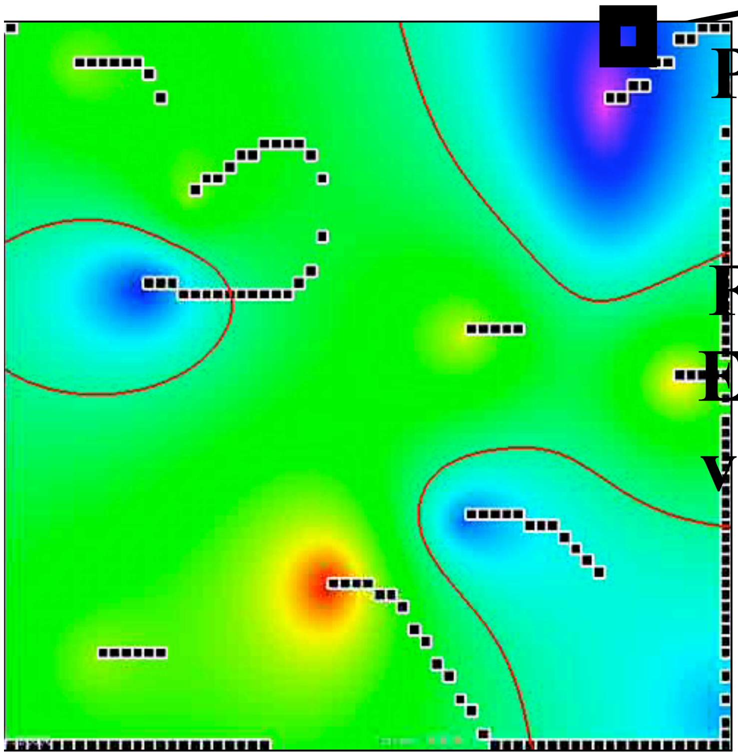


$$(f_{\min}, f_{\max})$$

Center for Computational Visualization
Institute of Computational and Engineering Sciences
Department of Computer Sciences
University of Texas at Austin

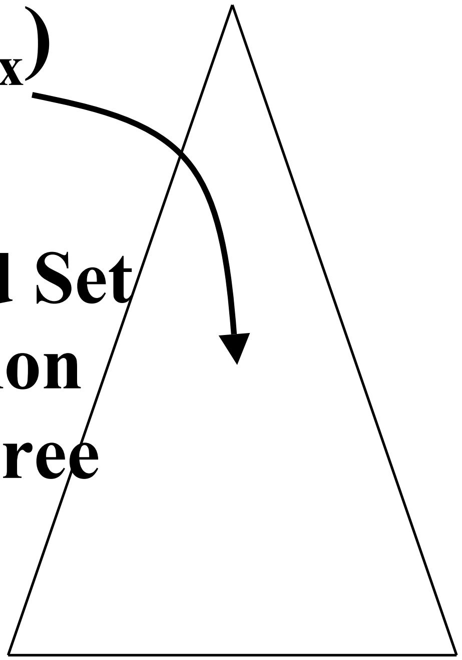
Optimal Single-Resolution Isocontouring

The basic scheme



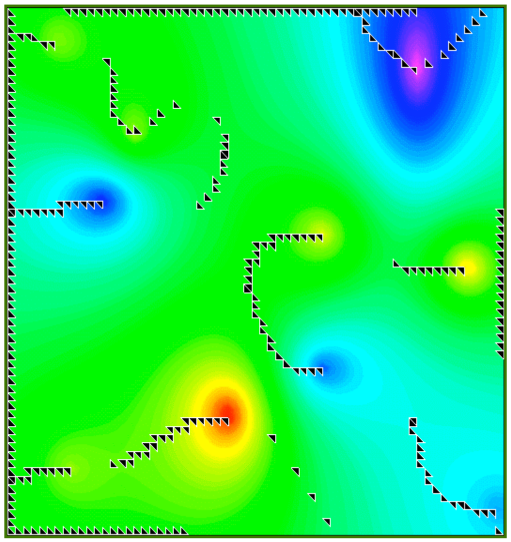
Preprocessing (f_{min}, f_{max}) **(revised):**

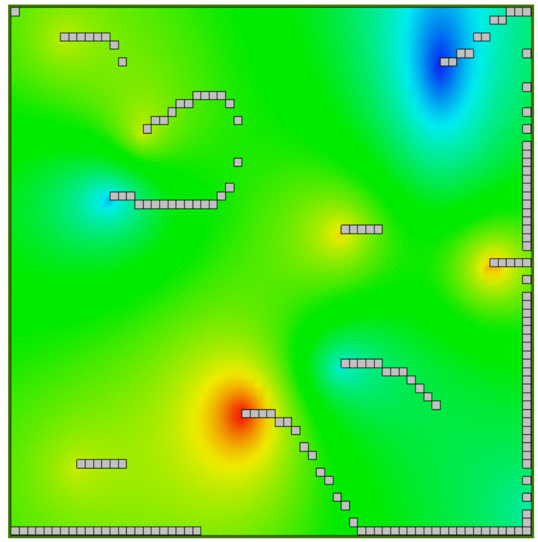
For each cell c in a Seed Set Enter its range of function values into an interval-tree

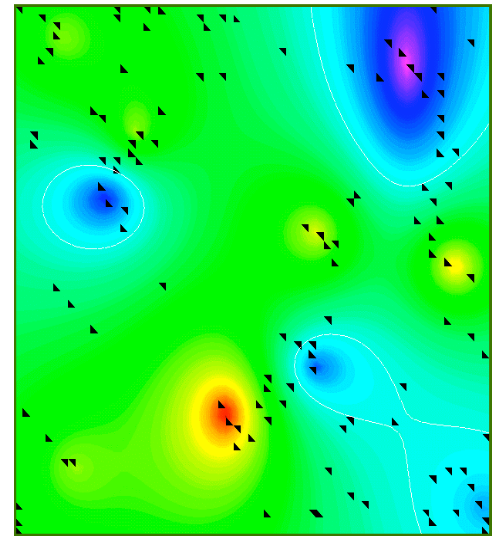


Optimal Single-Resolution Isocontouring

Seed Set Generation (*k* seeds from *n* cells)


Domain Sweep

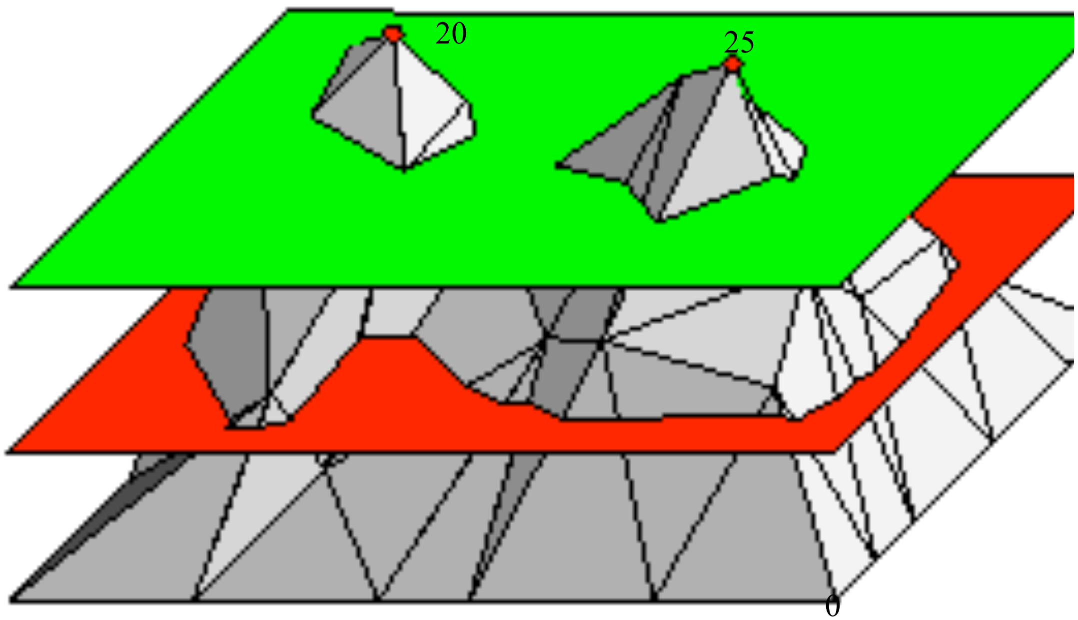

Responsibility Propagation

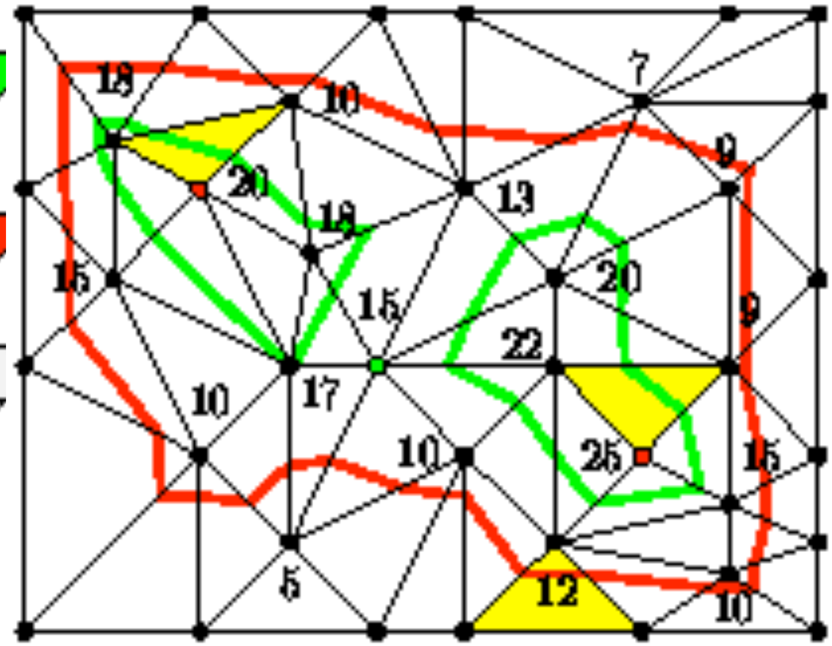

Range Sweep

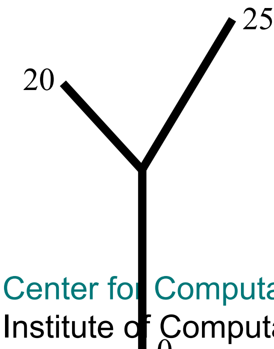
	Domain Sweep	Responsibility Propagation	Range Sweep
Time	**O(*n*)**	**O(*n*)**	**O(*n* log *n*)**
Space	**O(*k*)**	**O(*k*)**	**O(*n*)**
***k* =**	**?**	**?**	**$2\,k_{min}$**
Test	238 seed cells 0.01 seconds	177 seed cells 0.05 seconds	59 seed cells 1.02 seconds

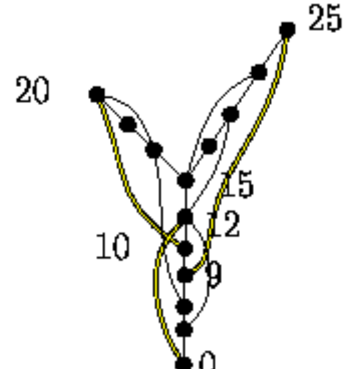
Optimal Single-Resolution Isocontouring

Contour tree

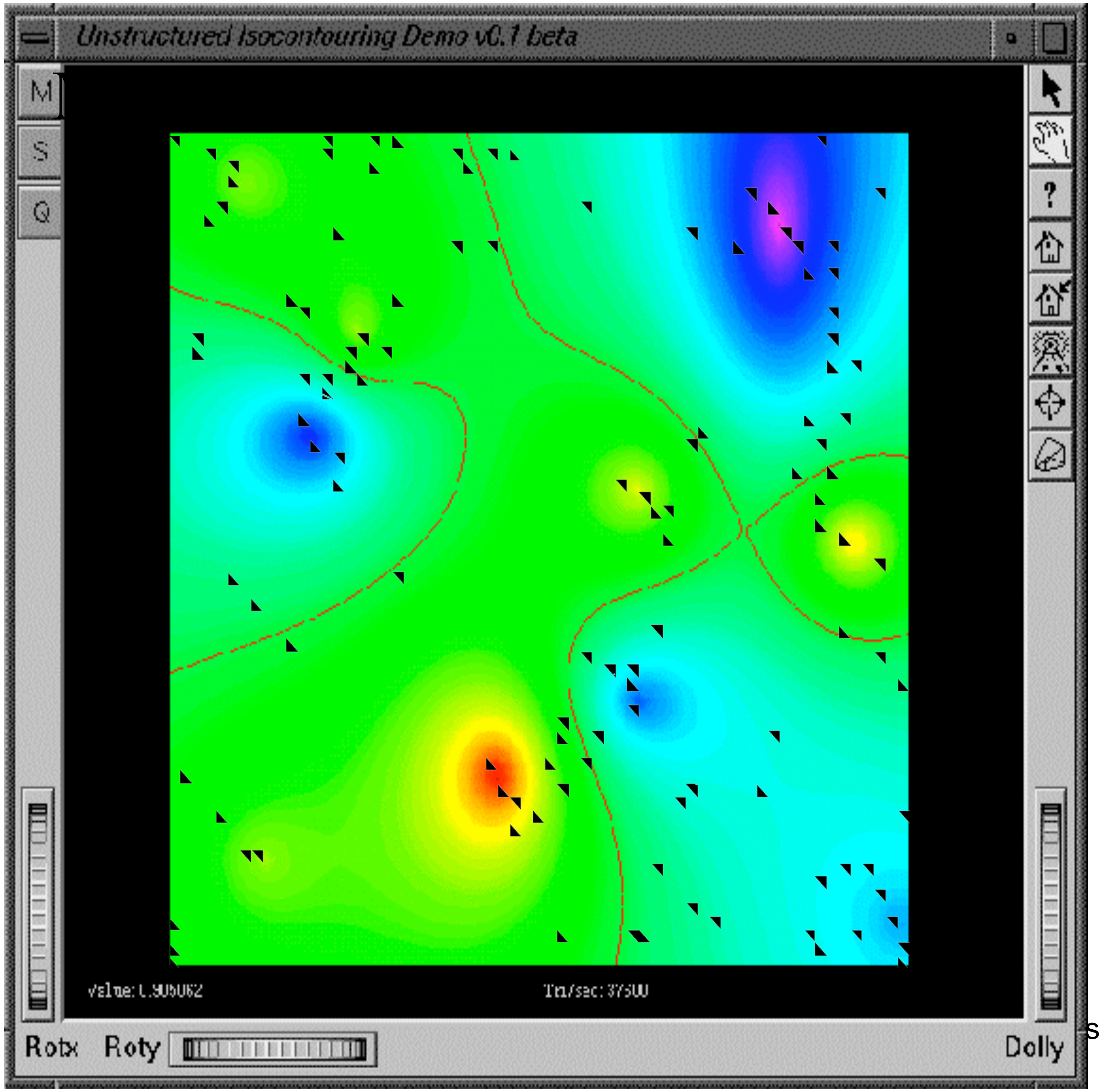




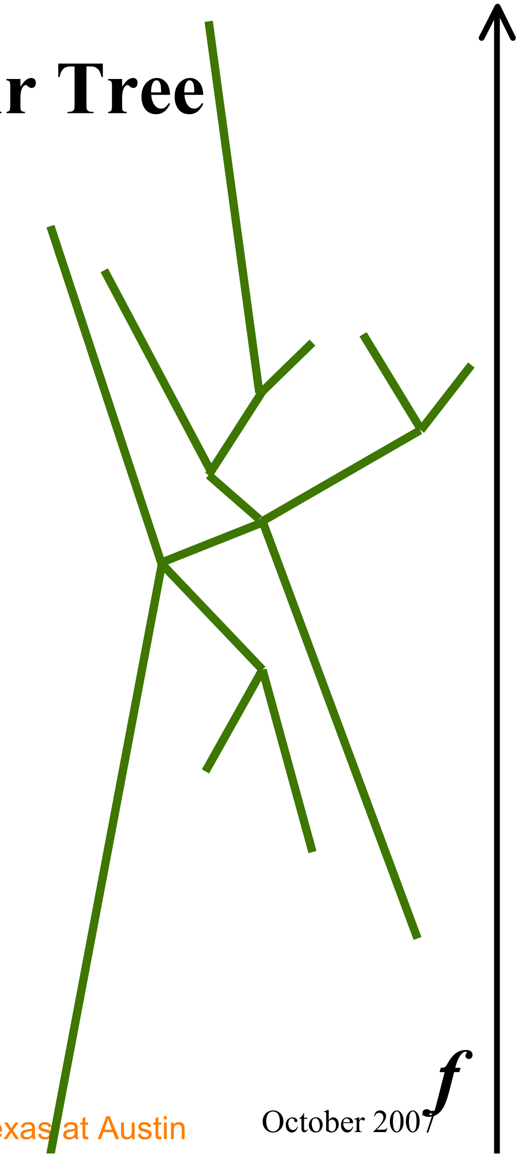




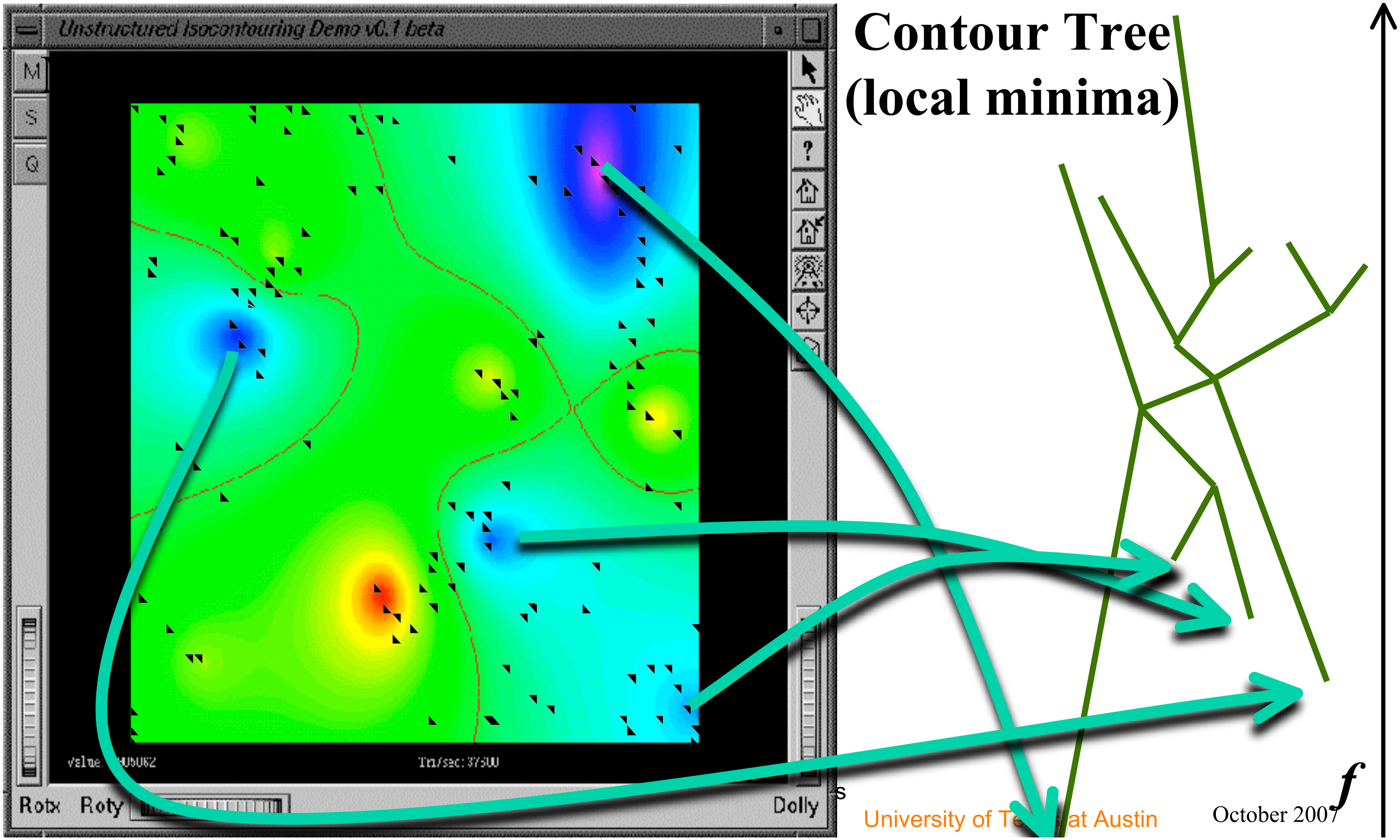
Optimal Single-Resolution Isocontouring



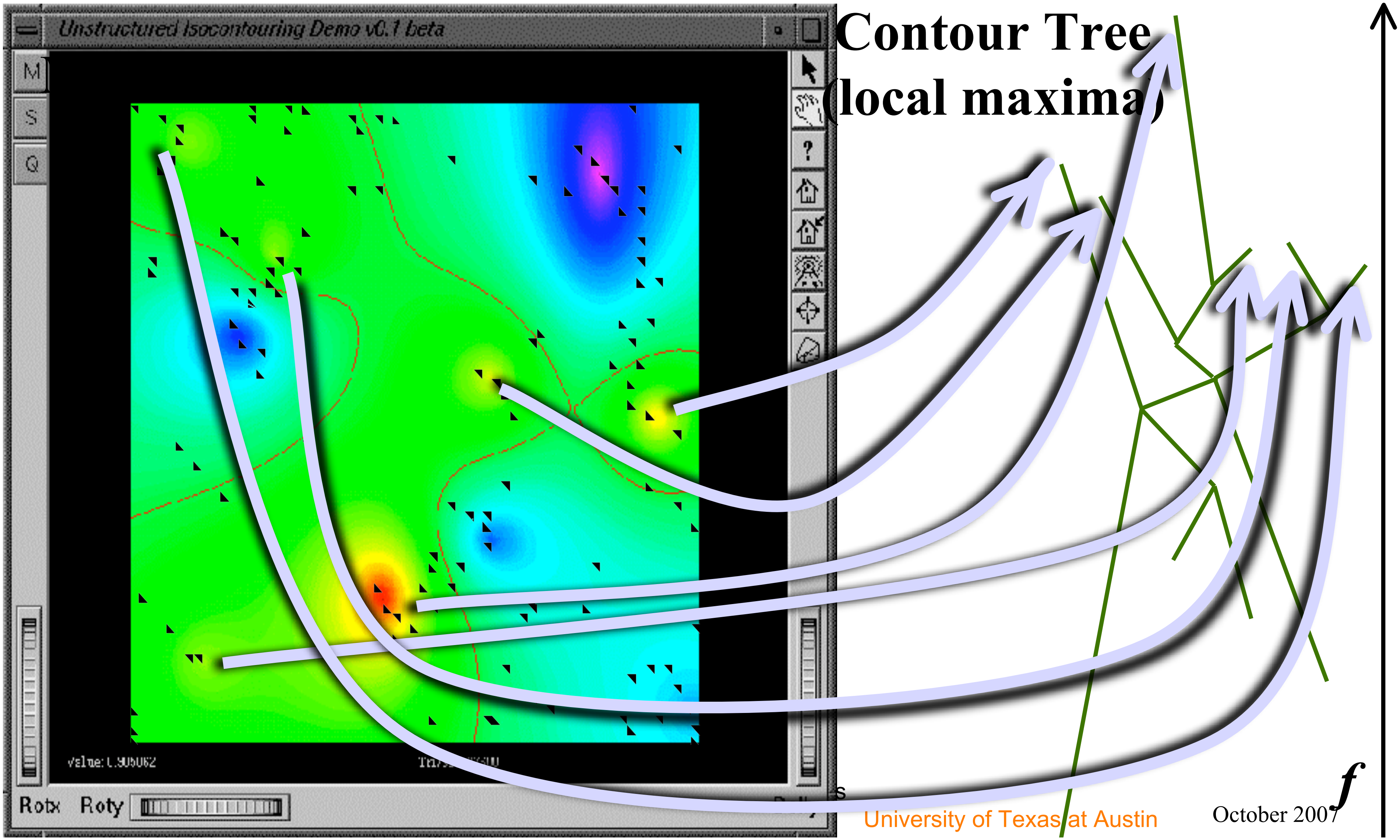
Contour Tree



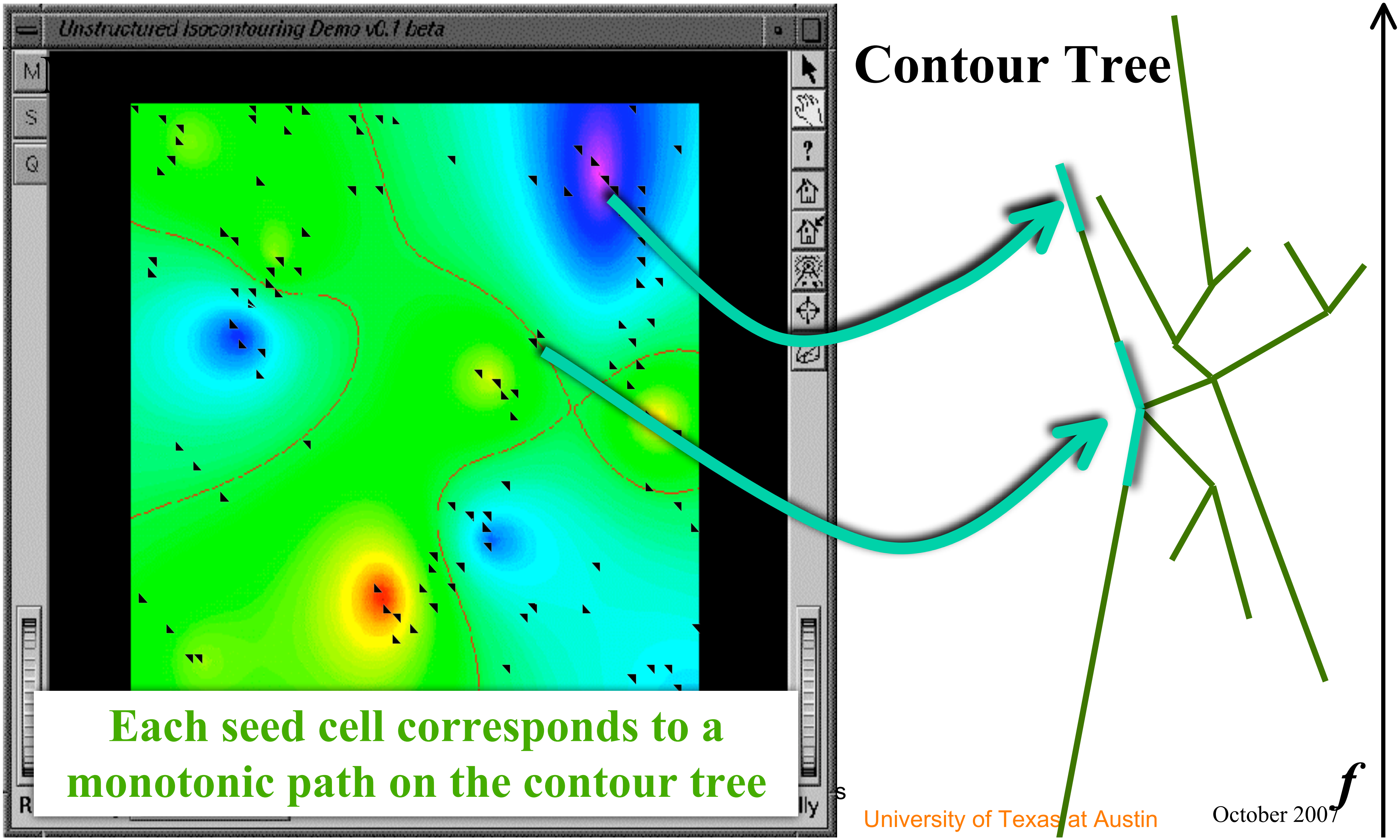
Optimal Single-Resolution Isocontouring



Optimal Single-Resolution Isocontouring

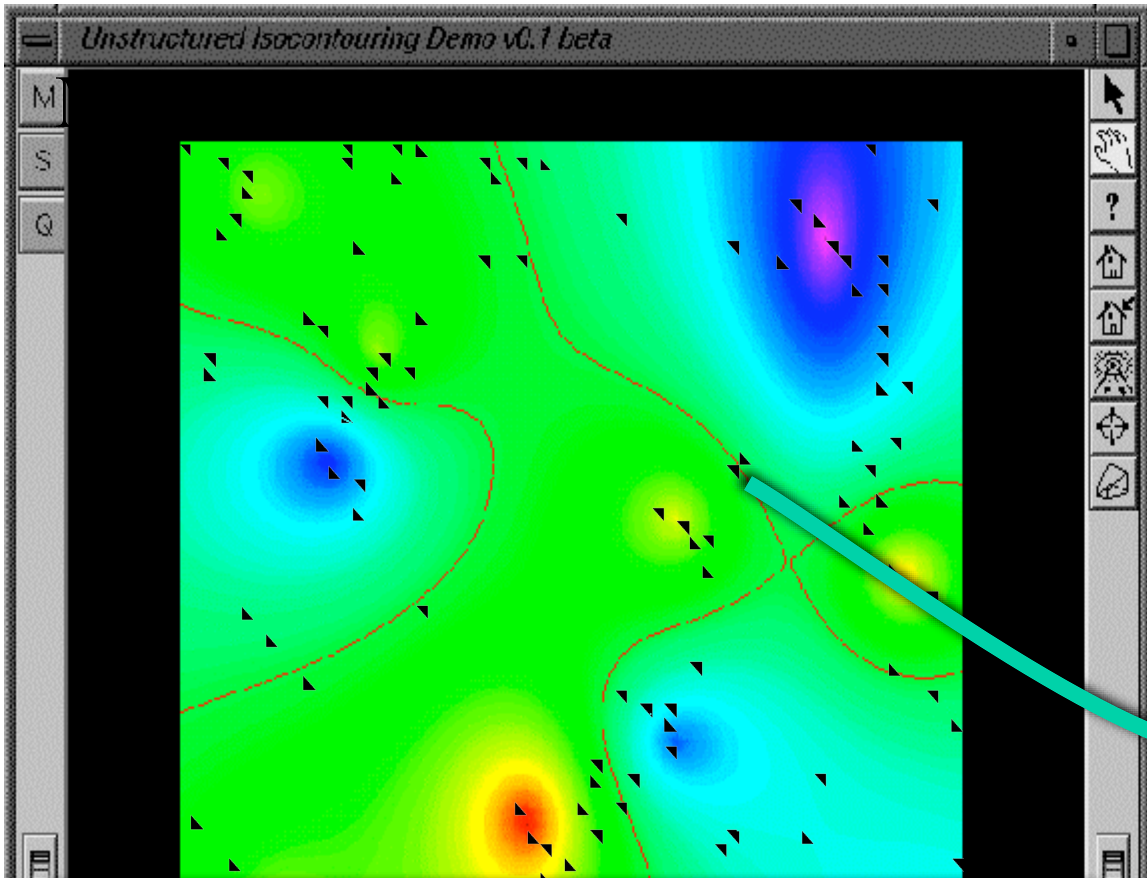


Optimal Single-Resolution Isocontouring



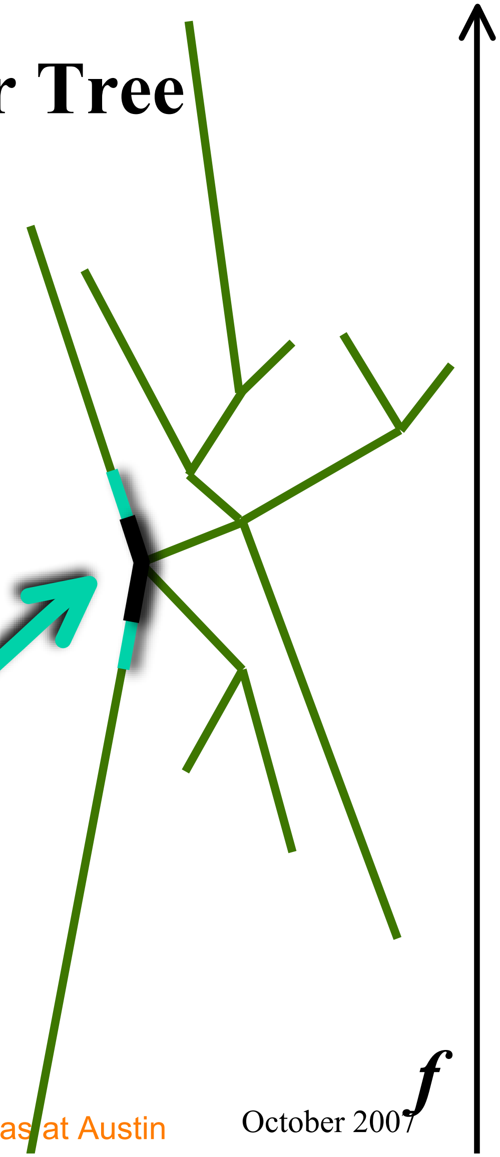
Each seed cell corresponds to a monotonic path on the contour tree

Optimal Single-Resolution Isocontouring

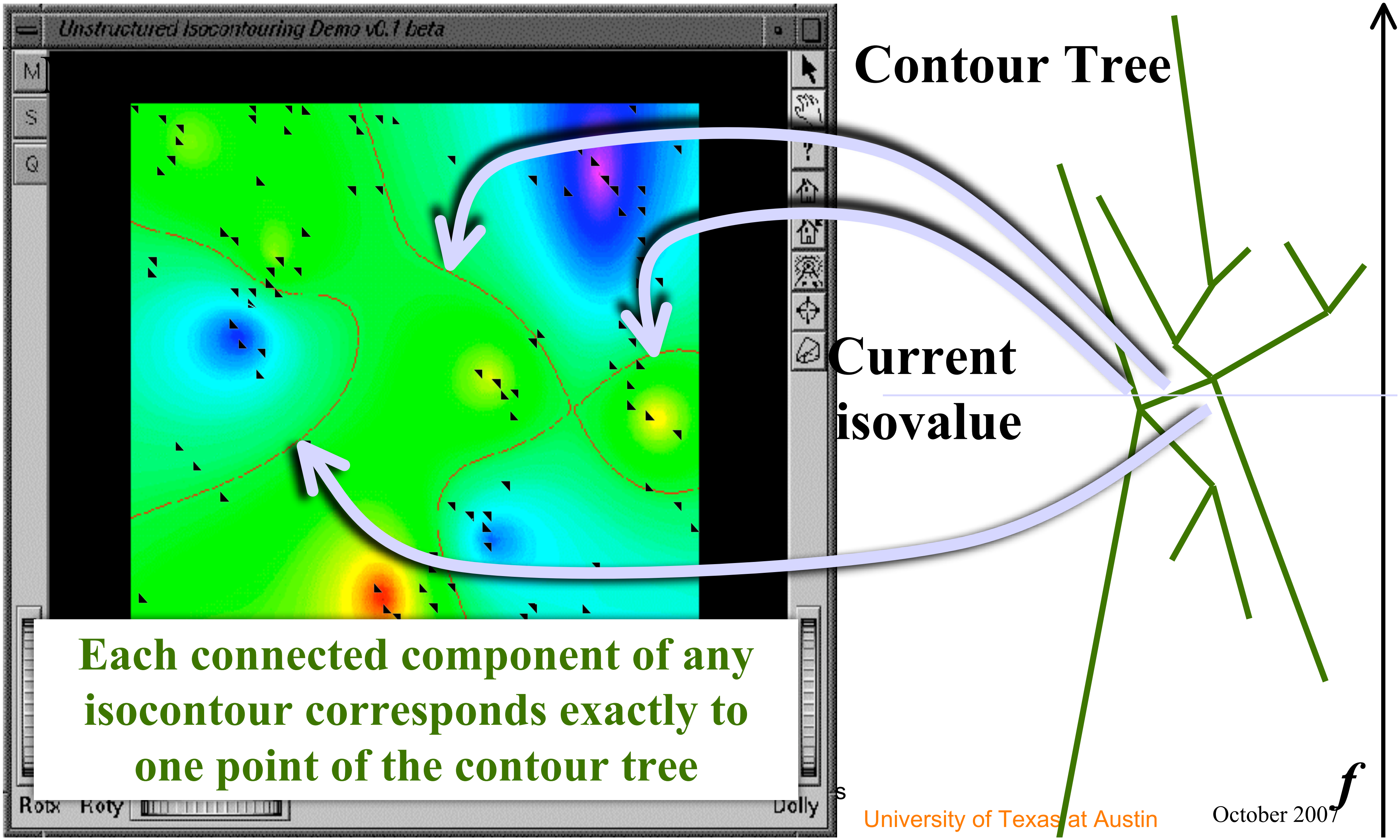


For a minimal seed set each seed cell corresponds to a path that is not covered by any over seed cell

Contour Tree

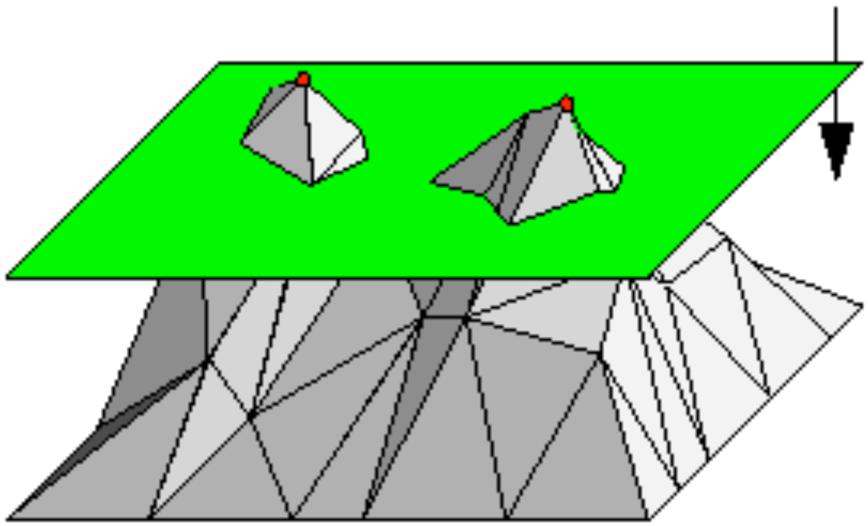


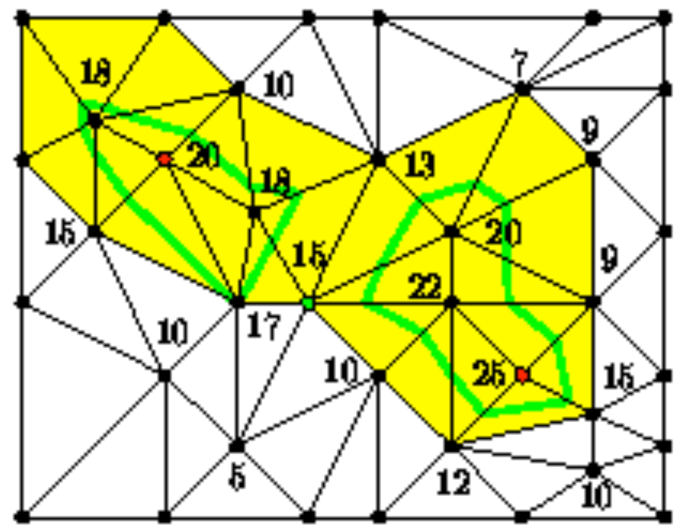
Optimal Single-Resolution Isocontouring

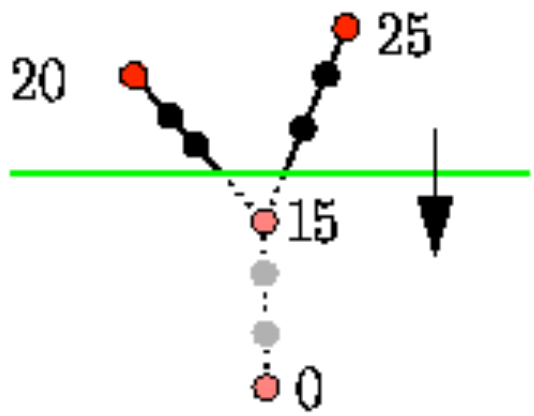


Each connected component of any isocontour corresponds exactly to one point of the contour tree

Optimal Single-Resolution Isocontouring

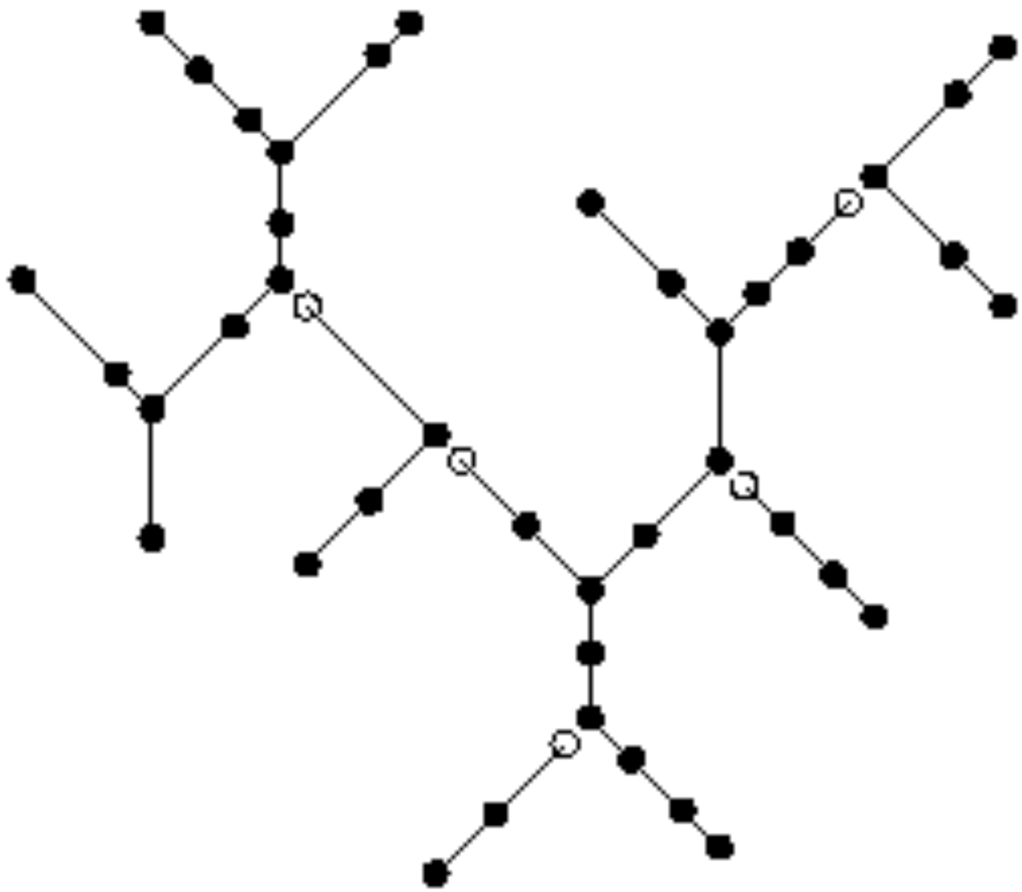






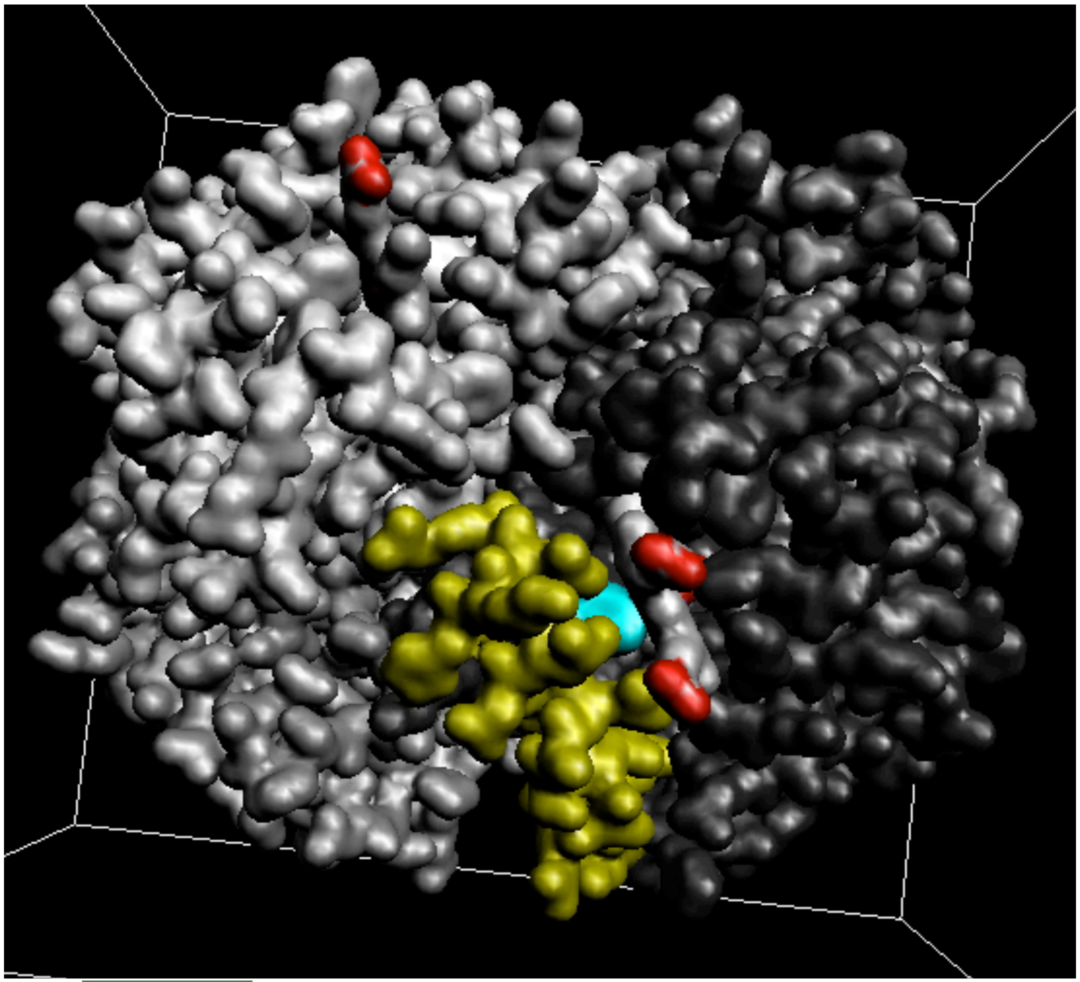
Optimal Single-Resolution Isocontouring

- The number of seeds selected is the minimum plus the number of local minima.



Structural Analysis

Contour Spectrum and Contour Tree on Hemoglobin Dynamics



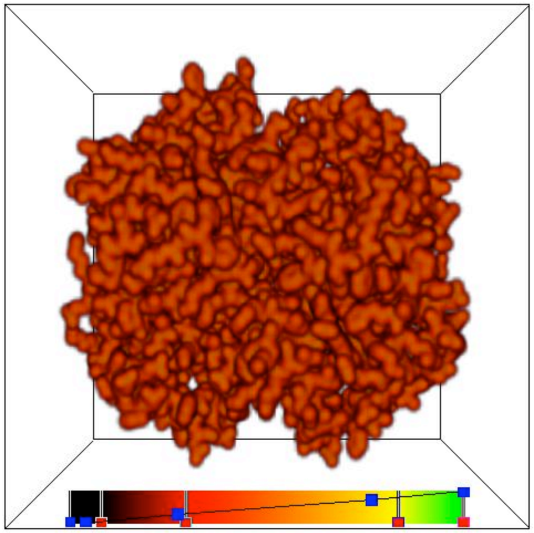
Subunit A
Subunit B
Subunit C
Subunit D

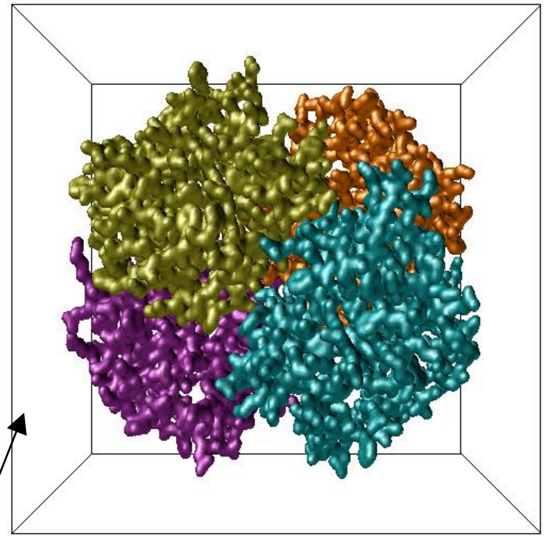
Within Subunit A

F helix
Histidine Ligand(HIS87)
O_2

- Oxy process : O2 binds to the Fe2+ ion on the opposite side of the histidine ligand. F helix shifts position through the oxy-deoxy cycle.

Topological Analysis & Visualization



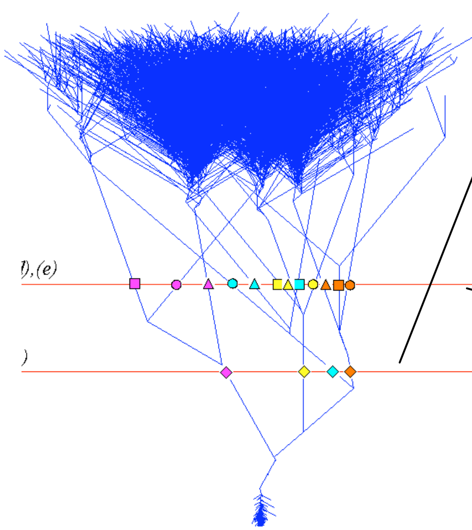


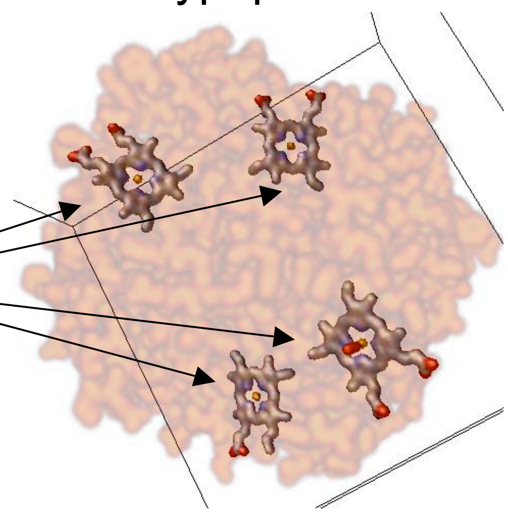
Four Polypeptide chains

Contour Tree
of
Electron Density Map

↓

3D chemical bonding
structures
with different levels





Functional groups

Atoms belonging to the same contour have stronger linkage

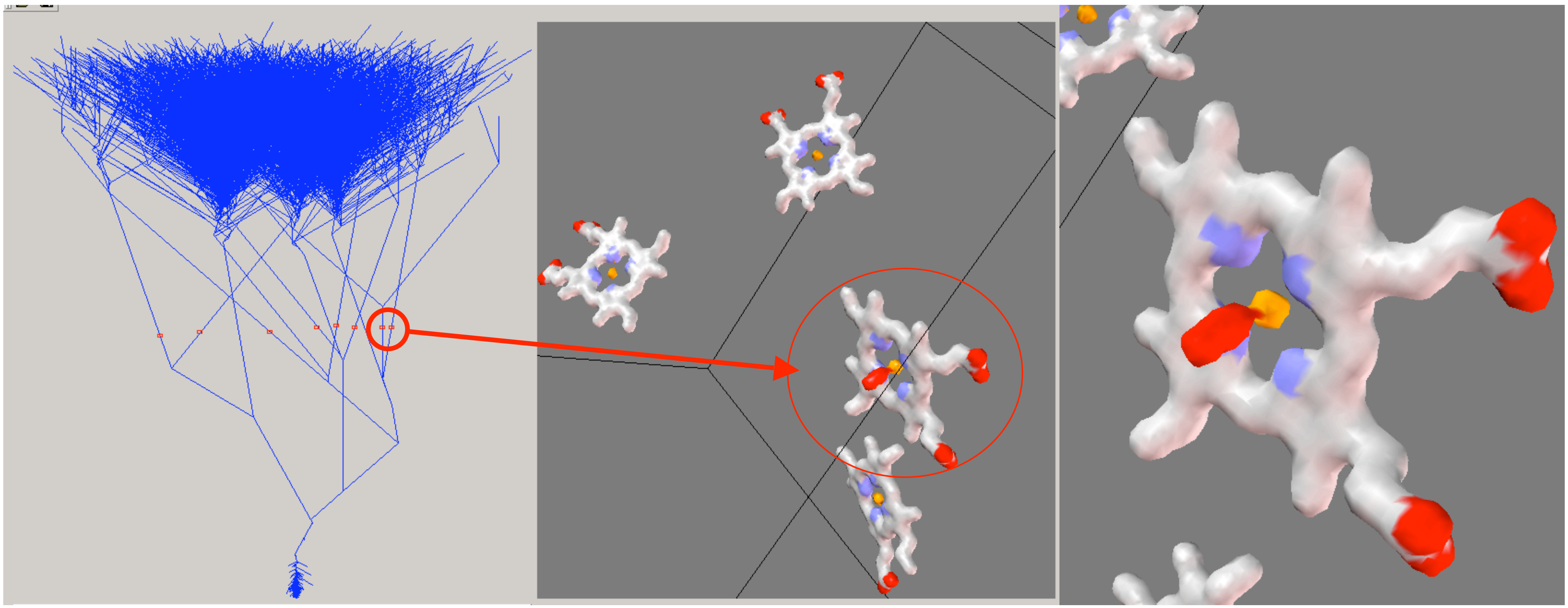
Each chain consists of heme, iron, and globin

•M. van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. Schikore, *Chap5, pg 71 - 86, 2004 ed. by S. Rana, John Wiley & Sons, Ltd, 2004*

•C. Bajaj, V.Pascucci, and D.Schikore, *Proceedings of the 1997 IEEE Visualization Conference,167-173, October 1997 Phoeniz, Arizona*

Topological Analysis using the CONTOUR TREE

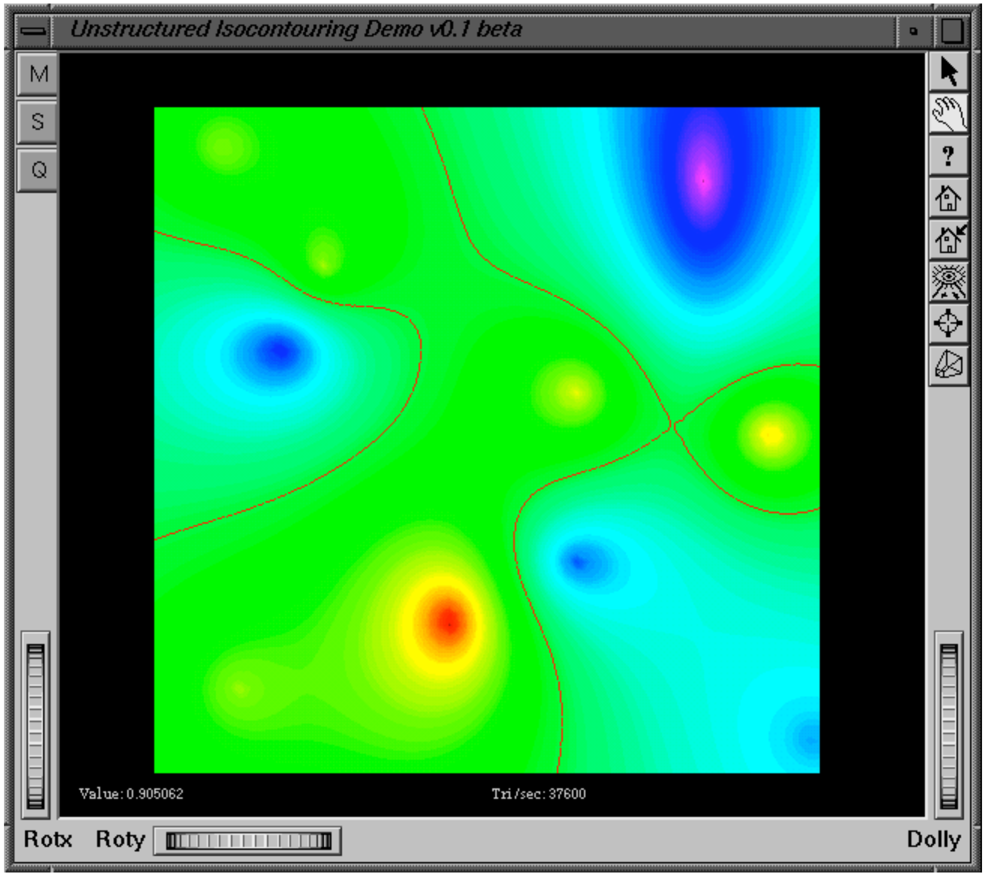
- Oxygenated Hemoglobin (T=1)



<isovalue = 31>

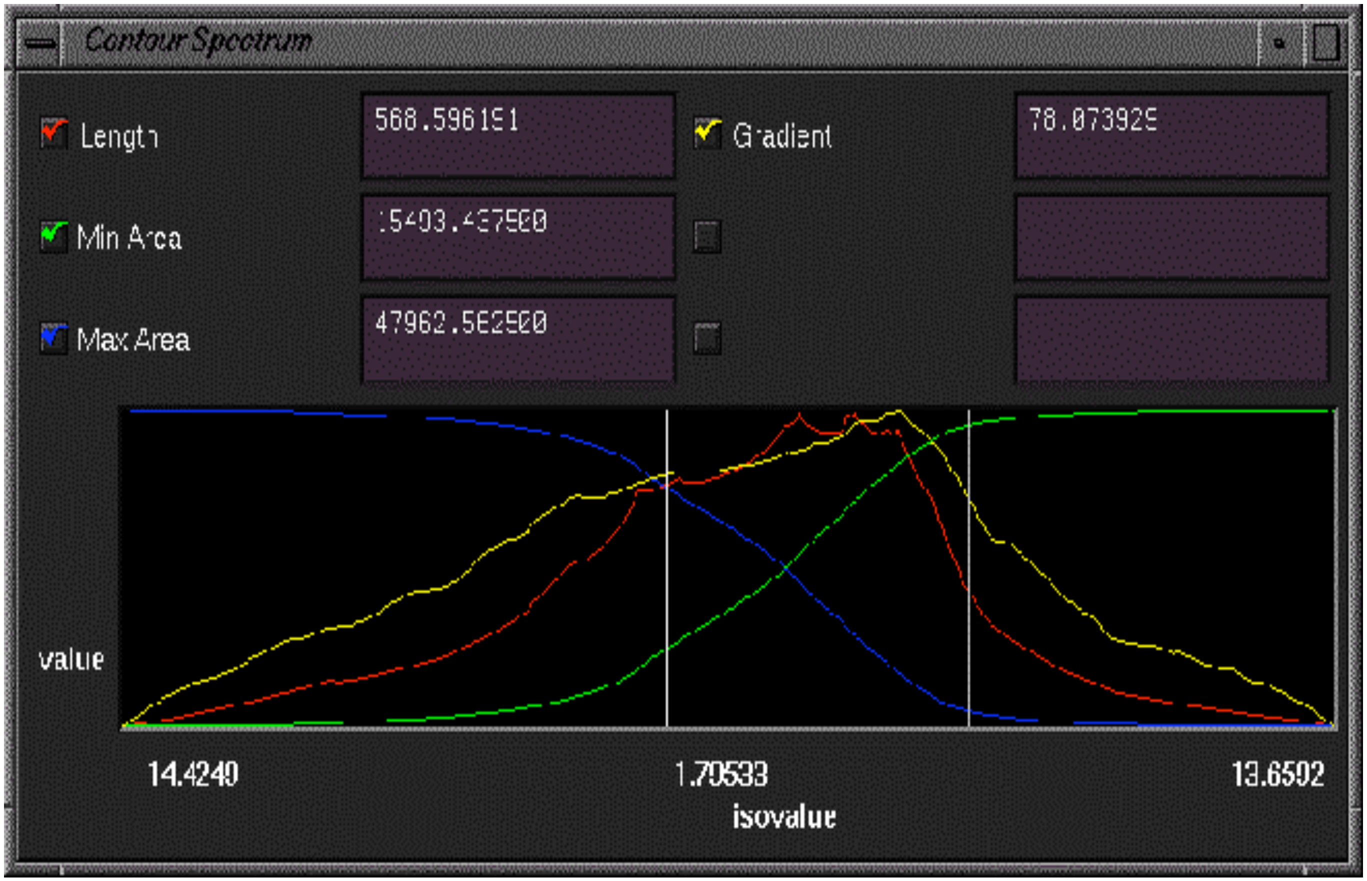
Spectral Analysis

- Consider a terrain of which you want to compute the length of each isocontour and the area contained inside each isocontour.



Spectral Analysis

Graphical User Interface for Static Data



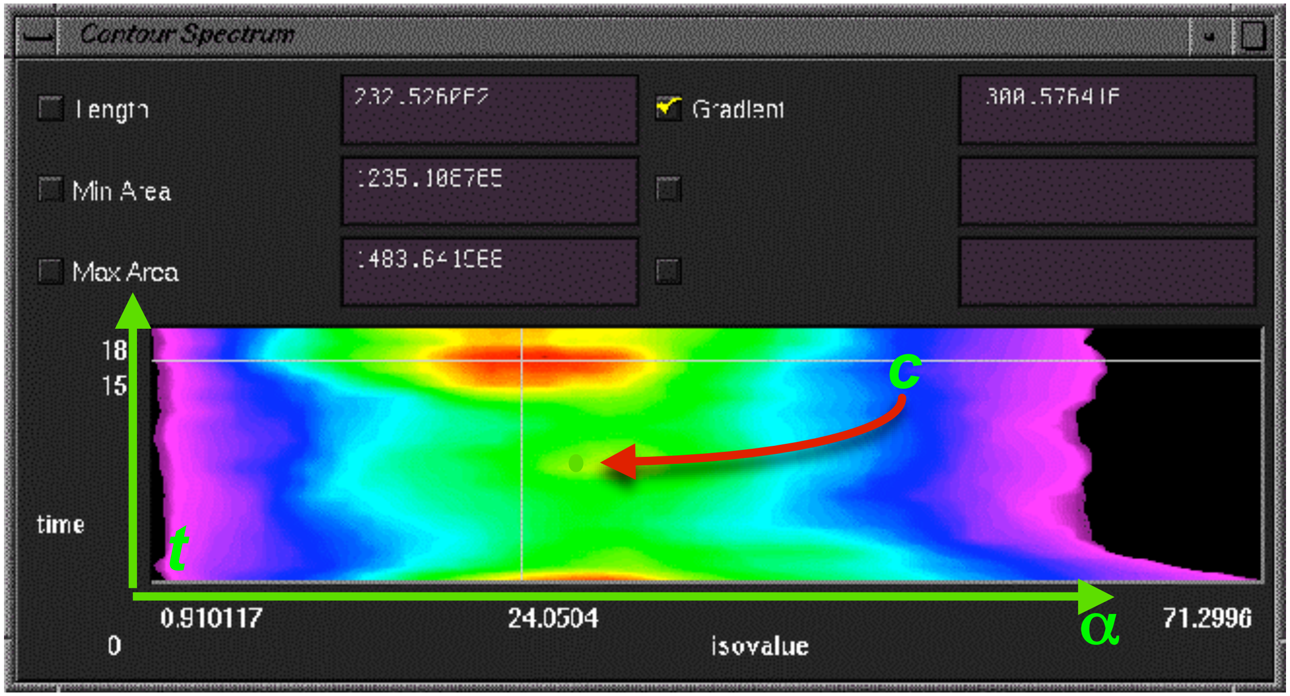
– The horizontal axis spans the scalar values α.

– Plot of a set of signatures (length, area, gradient ...) as functions of the scalar value α.

- Vertical axis spans normalized ranges of each signature.

Spectral Analysis

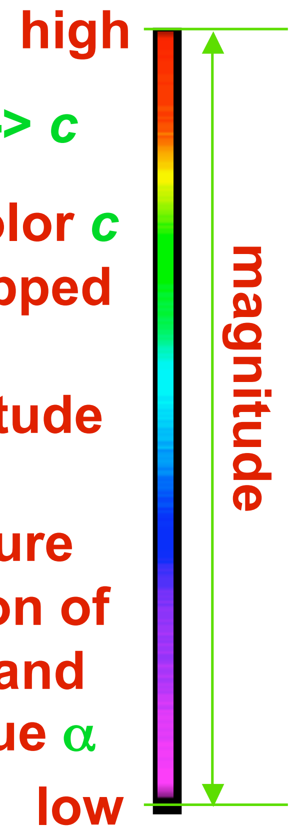
Graphical User Interface for time varying data



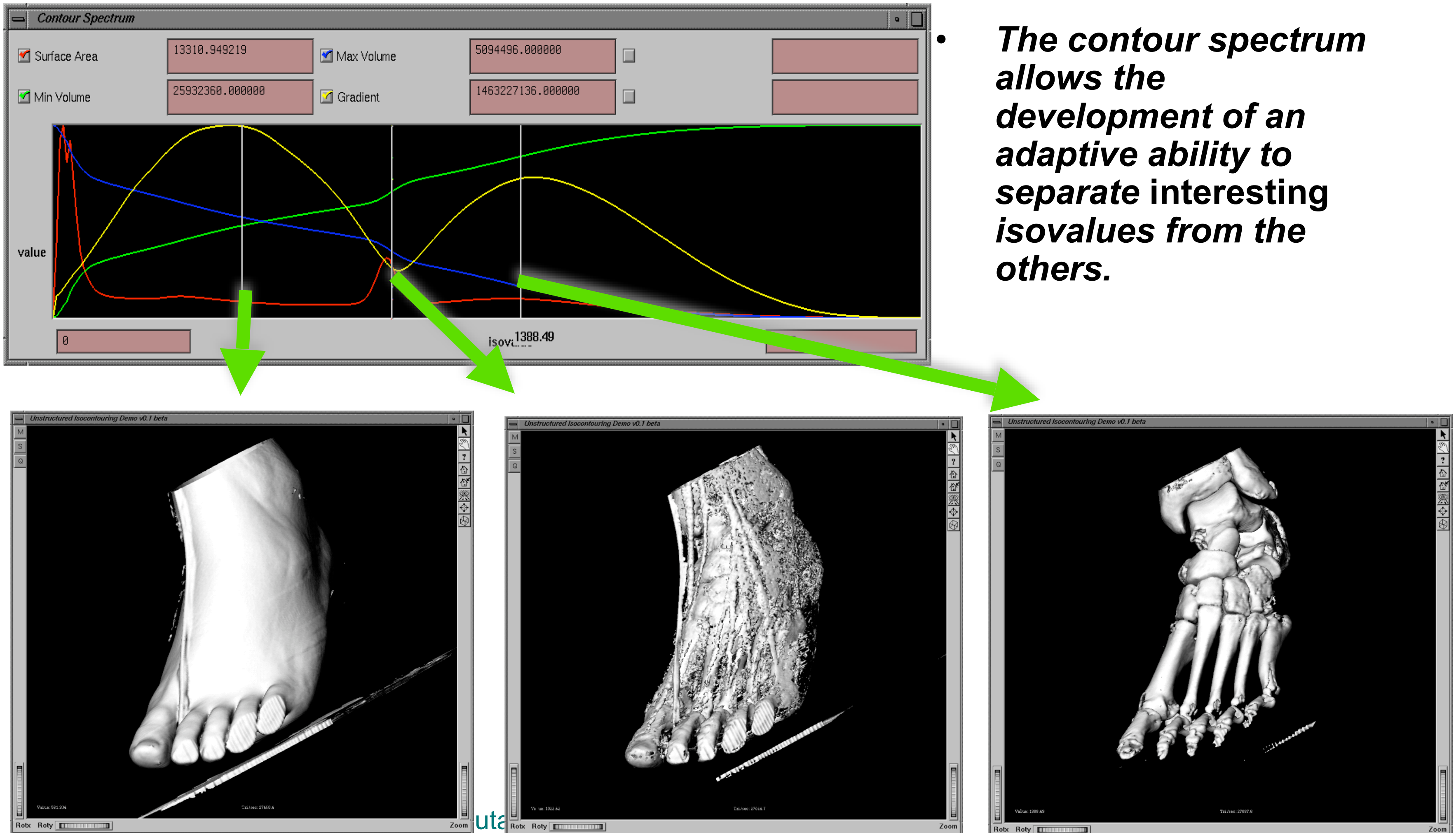
The horizontal axis spans the scalar value dimension α
The vertical axis spans the time dimension t

(α,t) --> c

The color c is mapped to the magnitude of a signature function of time t and isovalue α

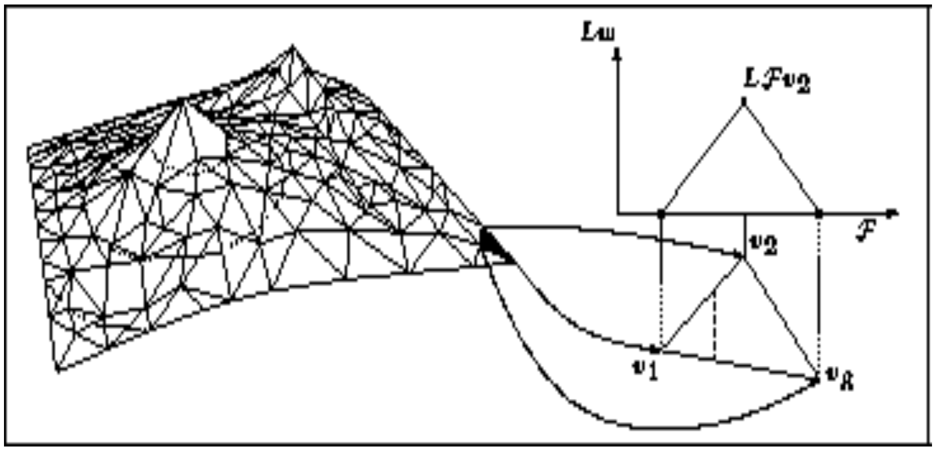


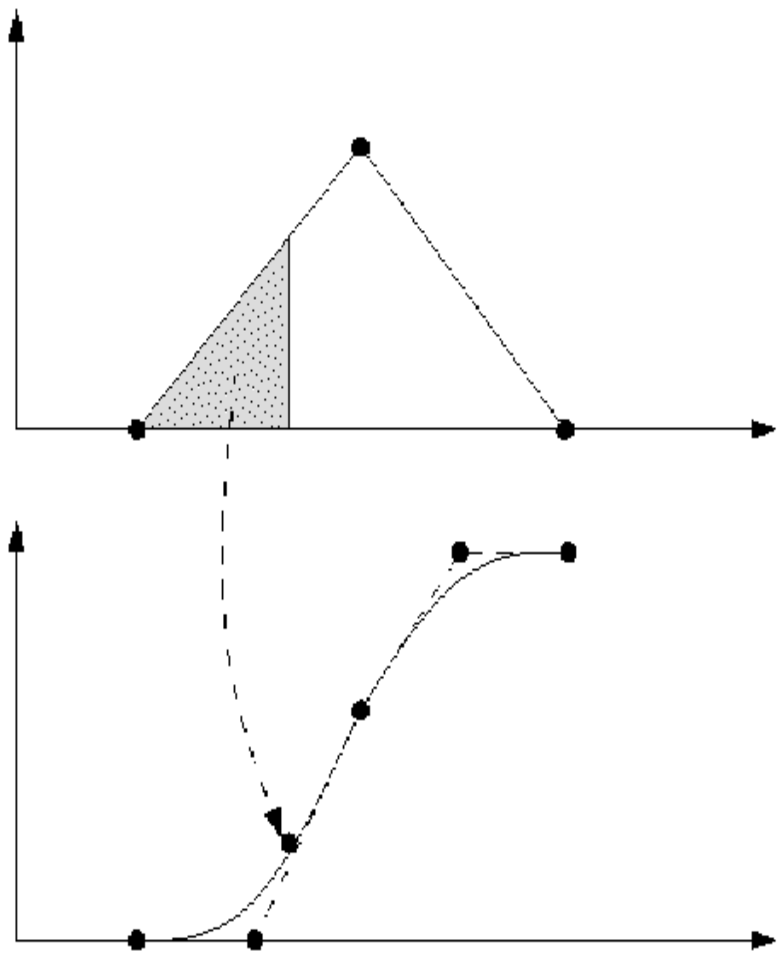
Contouring based Selection



- ***The contour spectrum allows the development of an adaptive ability to separate interesting isovalues from the others.***

Spectral Analysis (signature computation)



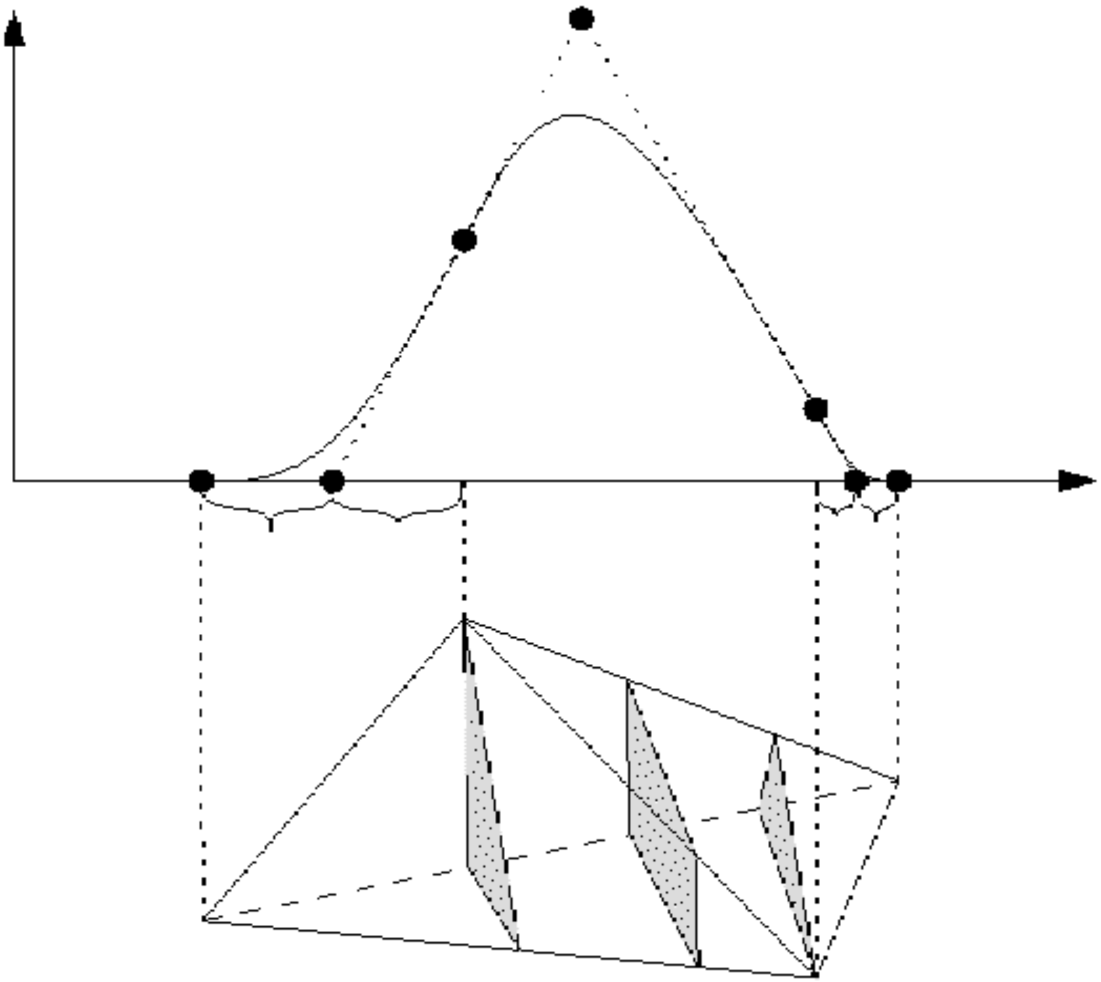


– **The length of each contour is a C^0 spline function.**

The area inside/outside each isocontour is a C^1 spline function.

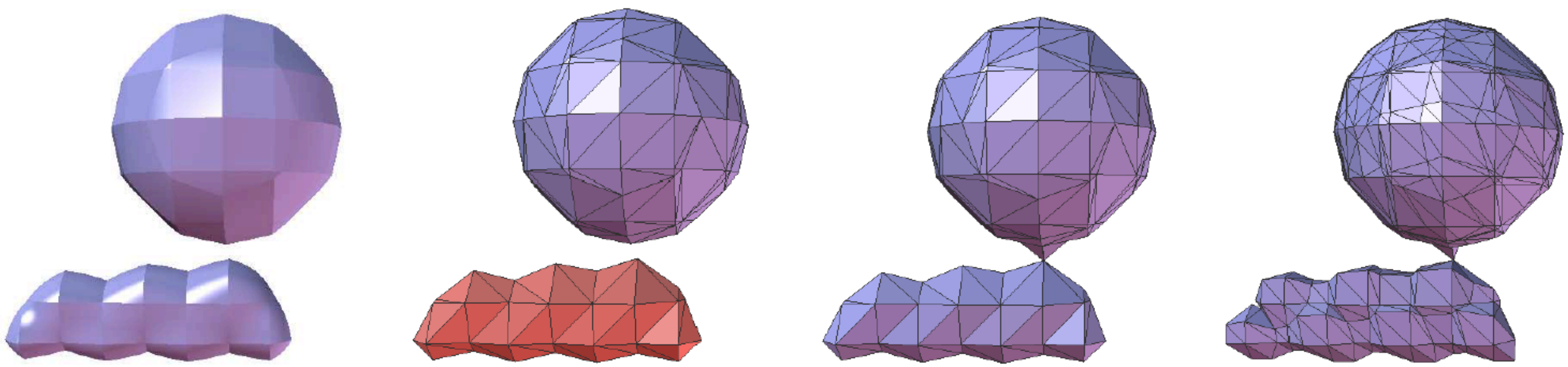
Spectral Analysis (signature computation)

- In general the size of each isocontour of a scalar field of dimension d is a spline function of d-2 continuity.
- The size of the region inside/outside is given by a spline function of d-1 continuity



Applications
[Contour Tree Based Visualization]

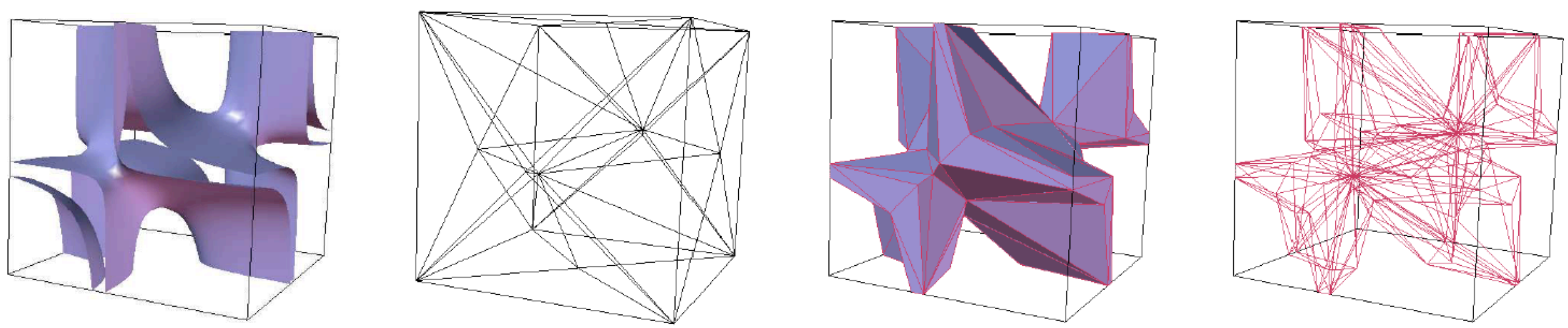
- Perfrom Tetrahedral Decomposition of Rectilinear Data (Trilinear Isosurface Topology is preserved)
- Apply Contour Tree and Seed Set computation, and Contour Propagation for Isosurface Component Segmentation



(a) Trilinear (b) Cell Decomposition (c) Marching Cubes (d) Marching Tetra

Applications
[Trilinear Interval Volume Tetrahedrization]

- Perform topology preserving tetrahedral decomposition method
- Apply interval volume tetrahedrization to each tetrahedra generated from our method



(a) boundary isosurfaces (b) tetrahedral decomposition (c) interval volume (d) wireframe

Further Reading

- C. Bajaj (ed) "DataVisualization Techniques", John Wiley & Sons 1998
- C. Bajaj, V. Pascucci, D. Schikore, "Contour Spectrum" IEEE Viz,1997
- M. van Kreveld, van Oostrum, C. Bajaj,V. Pascucci, D. Schikore "Contour Trees & Small Seed Sets" ACM SoCG 1997, also book chap in 2004
- B. Sohn, C. Bajaj. "Topology Preserving Tetrahedral Decomposition of Trilinear Cell", CS/ICES Tech. Rep. TR2004.
- S.Goswami, A. Gillette, C. Bajaj "Efficient Delaunay Mesh Generation from Sampled Scalar Functions", 16h IMR, 2007
- J. Bloomenthal, C. Bajaj, J. Blinn, M. Gascuel, A. Rockwood, B. Wyvill, G. Wyvill **Introduction to Implicit Surfaces** *Morgan Kaufman Publishers Inc., (1997).*