Progressive Simplification of Tetrahedral Meshes Preserving All Isosurface Topologies

Yi-Jen Chiang

Xiang Lu

Polytechnic University, NY, USA

(appeared in Eurographics, Sept. 2003)

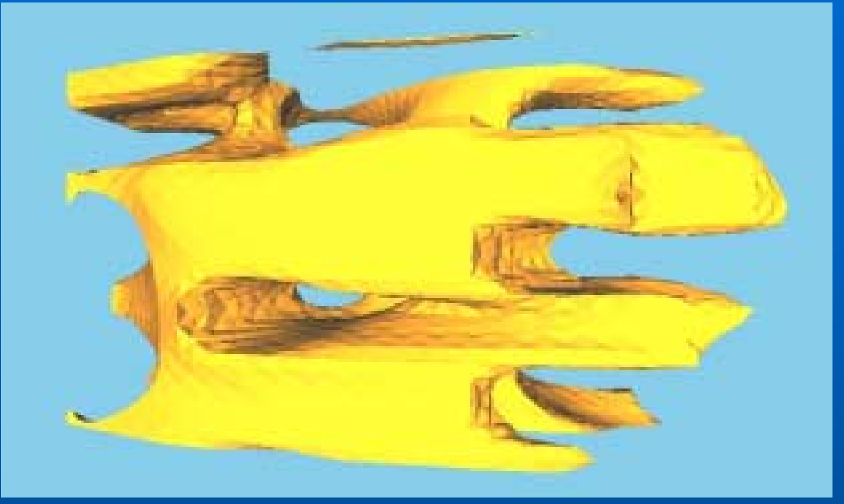
Problem Specification

- **Instance:**
  - Volume dataset: (x, F(x))

- **Task:**
  - For a user-specified error tolerance ε (in a given error metric), progressively simplify the volume such that we
  - Preserve the topologies of all isosurfaces embedded in the volume
  - Control the geometric accuracy of the volume (and the isosurfaces) by ε

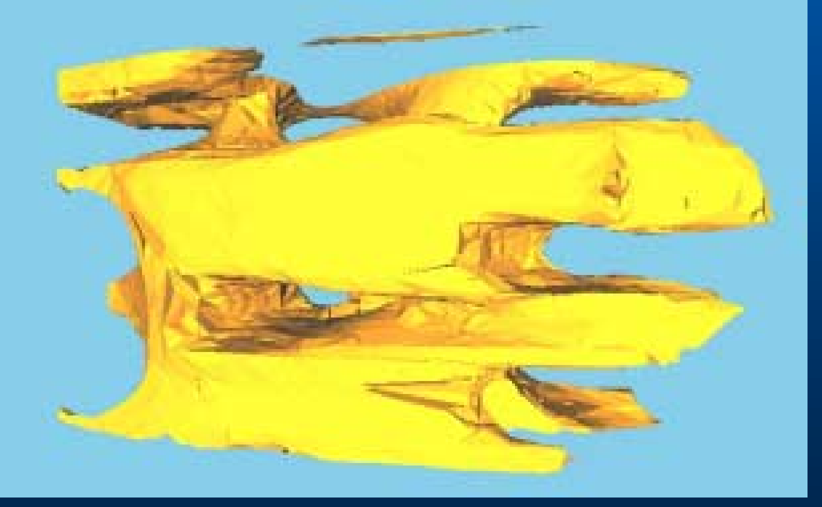
* Isosurface of q: C(q) = {p | F(p) = q}

Example: Combustion Chamber



Original volume isovalue = 0.251

Simplified, no isosurface topology guarantee; ε = 0.64



Simplified, with isosurface topology guarantee; ε = 0.64

Motivation

- **Displaying isosurfaces** is one of the most powerful visualization techniques for volume data
- For large datasets, **multi-resolution** methods are essential for efficient visualization
- During **simplification** to create multiple LODs, it is crucial to still **capture the features** of the original data
- One of the most critical features is the **topologies** of **all isosurfaces**

➔ We want **volume simplification** that **preserves all isosurface topologies**

Motivation (cont.)

- **Naïve approach:**
  - **Use simplification method without isosurface topology guarantee, and adjust ε back and forth to find the right ε to use**
  - **User visual inspection: No correctness guarantee; tedious & slow**
  - **Try & error (thousands of critical isovalues to test): Infeasible**

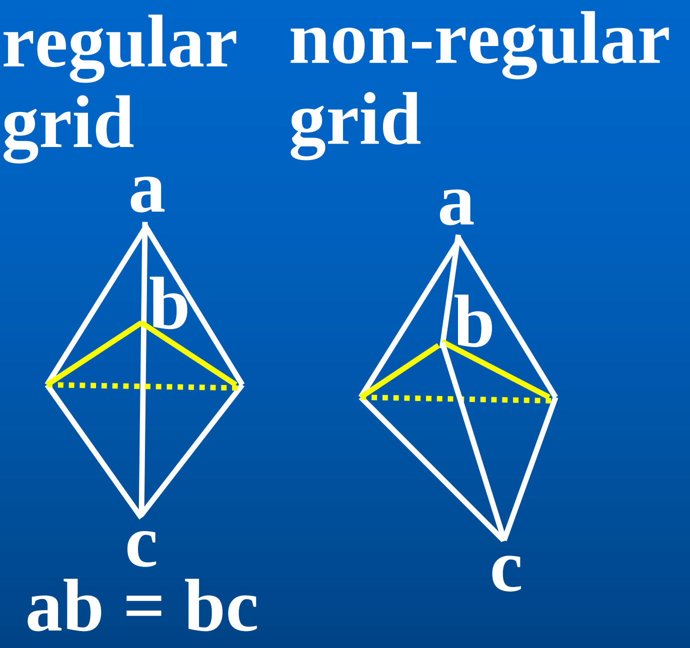
➔ **We need simplification algorithm that automatically guarantees the correctness for any value of ε**

Previous Work: Volume Simplification

- [Staadt-Gross 98], [Trotts et al 98], [Trotts et al 99]
  - Tetrahedral volume simplification by edge collapses
  - No topology guarantee on volume or isosurfaces
- [Dey et al 99], [Cignoni et al 00]
  - Tetrahedral volume simplification by edge collapses
  - Preserve the topology of the volume itself (rather than isosurface topologies)
- [Chopra-Meyer02]
  - Simplifies tetrahedral volume by collapsing tetrahedral cells (Fast)
  - No topology guarantee on volume or isosurfaces

Previous Work (cont.)

- [Gerstner-Pajarola 00]
  - Simplifies regular grids
  - Preserves all isosurface topologies
  - Tetrahedralize regular grid
  - Perform a reverse process of bisecting tetrahedral cells
  - Only works for regular grids



- [Wood et al 02]
  - Works on regular grid to simplify isosurface topology
  - Completely different problem: polygonal model acquisition (only one isosurface and one connected component)

  * We have an arbitrary no. of isosurfaces & components

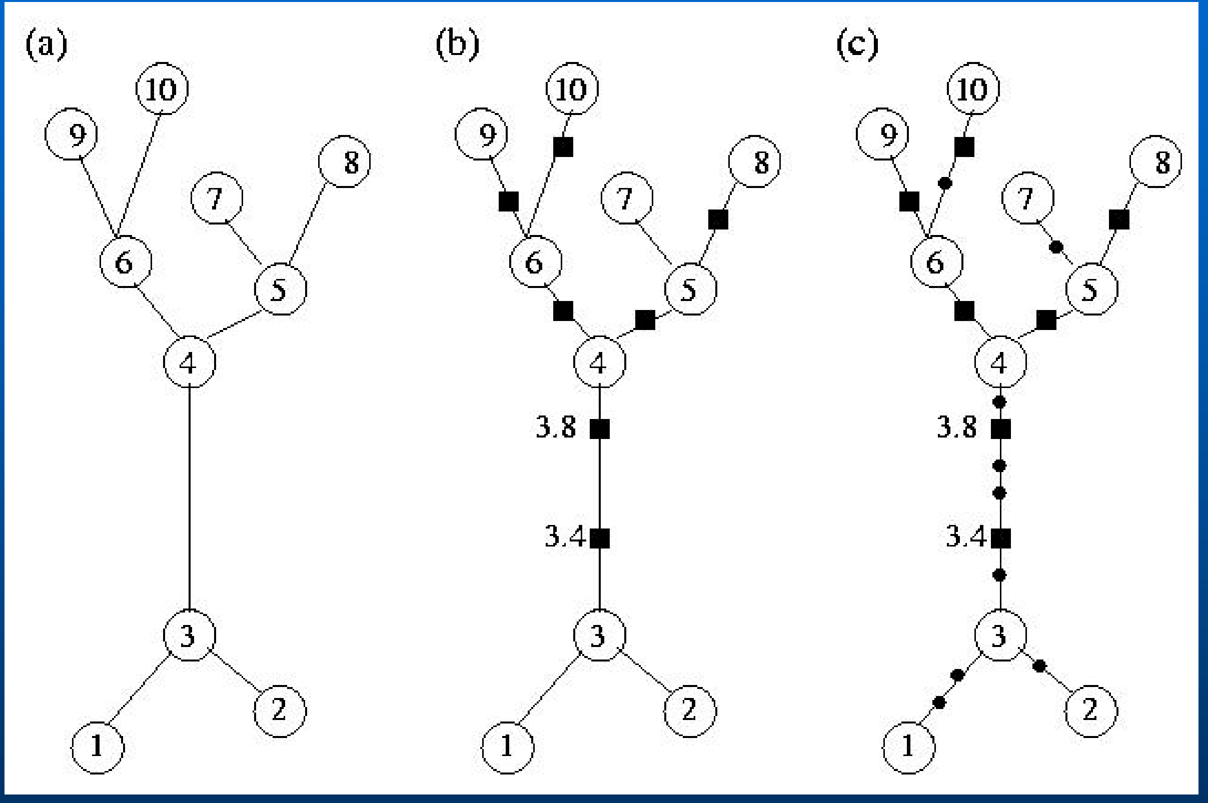
Our Contribution

- The first volume simplification algorithm that preserves all isosurface topologies for rectilinear, curvilinear, and irregular grids (represented as tetrahedral meshes)
- Also preserves the geometry of the volume boundary, and avoids the fold-over problem
- Develop a theoretical foundation; provide a theoretical guarantee for the correctness
- Achieve a nice data-reduction rate
- Algorithm runs competitively fast

Overview of Our Technique

- Based on (half-)edge collapses, but disallow the collapses if they cause an isosurface topology change
- Use Morse Theory [Banchoff 67]: critical points
  [Assume: scalar function is piece-wise linear]
  Key: Not enough to just preserve all critical points
- Two major classes of collapsibility tests:
  Check if collapsing an edge will
  (a) join two regions of different isosurface topologies
  (b) remove or create critical points
- Two phases in the algorithm:
  (1) Segmentation: identify top-eq regions, for (a)
  (2) Simplifcation

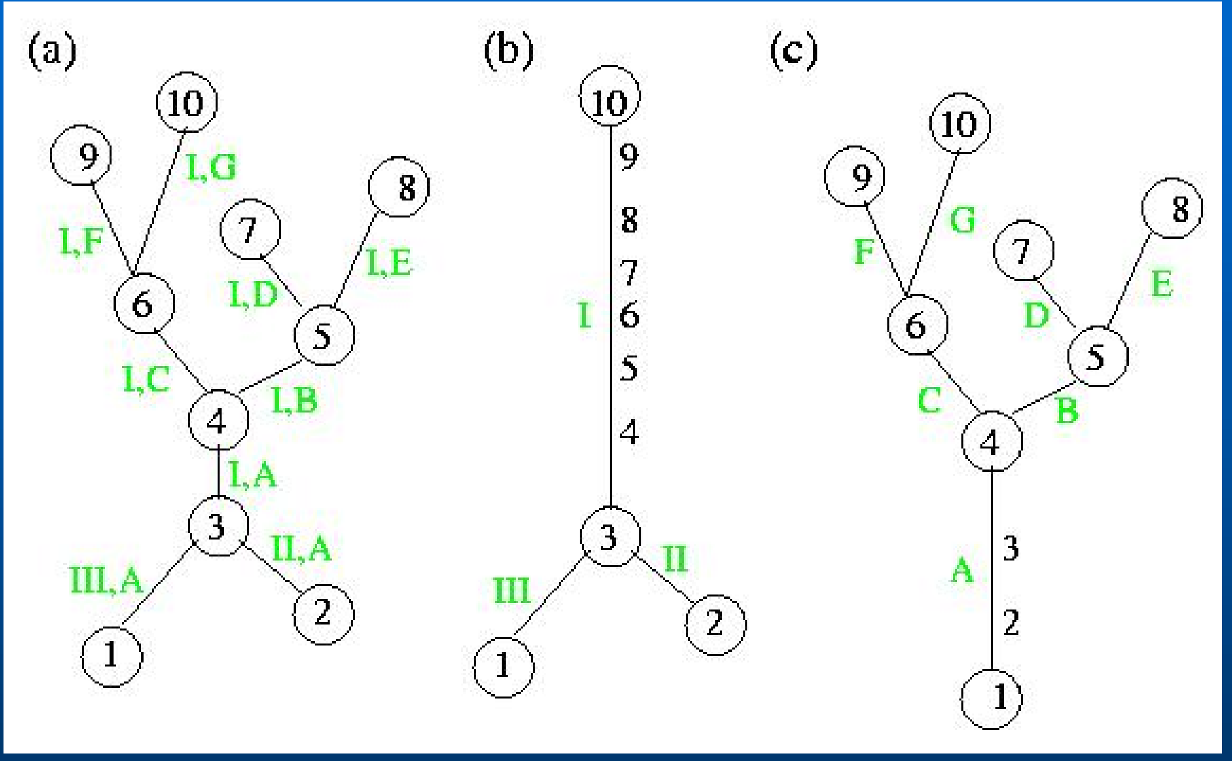
Segmentation Phase



- Goal: identify **top-eq** regions
- Use **fully augmented contour tree** (Ordinary contour tree [van Kreveld et al 97] does not capture **genus-change-only** events)

1. Classify all vertices as **critical** / **non-critical** [Chiang et al 03]
   * internal vertices * boundary vertices (**explicit method**)
2. Compute contour tree using [Carr et al 00]

Segmentation Phase (cont.)



2’. Compute contour tree **implicitly** by a **labeling scheme**
* join tree: (b)
* split tree: (c)
* **merge two trees implicitly by labeling**

3. Assign **non-critical vertices** & **cells** to **top-eq regions**
* non-critical vertex: label from fully aug. contour tree
* cell: assigned to **each top-eq region** of its **non-critical vertices**
-- **pure/impure** cell

Simplification Phase

- Simplify top-eq regions one by one, independently
- For each top-eq region, collapse edges from smallest to largest errors <= ε, subject to 6 types of collapsibility tests:

1. Critical edges: e has one or two critical endpoints
   → never collapse e (never put e to the priority queue Q)
2. Cross-region edges: endpoints of e are in 2 top-eq regions
   → never collapse e (by labels from Segmentation Phase)
3. Boundary-vertex edges: e = (v1, v2)
   * v1 & v2 are on boundary – never collapse e
   * v1 is on boundary but v2 is not – only allow v1← v2
   * Purposes:
   (a) preserve boundary geometry of the volume
   (b) make Type 4, 5 tests for boundary critical points easier

Simplification Phase (cont.)

4. **Critical-neighbor edges:** e has a critical point c as a neighbor
   → disallow collapsing e if it makes c non-critical (see paper for details)

5. **Criticality checking:** trying to collapse e = (v1,v2) to v
   → disallow the collapse if v or its non-critical neighbor becomes critical (prevent creating a new critical pt)

   * Naïve: collapse e to v & check criticality (full check)
   * Faster: Two-step checking or Easy-checking only

6. **Fold-over checking:**
   → disallow collapsing e if it produces a tetrahedral cell of negative volume (check each cell affected by the collapse)

Simplification Phase: Speed up

*** Avoid Expensive Tests:**

For criticality checking, **full checking** is slow
(e= (v1,v2) → v, check to avoid creating a **new critical pt**)

- **[Easy-checking Lemma]** Let F(v1) < F(v2). If **all neighbors** of v1 and v2 have **scalar values < F(v1) or > F(v2),** then v is **non-critical** & all non-critical neighbors stay non-critical

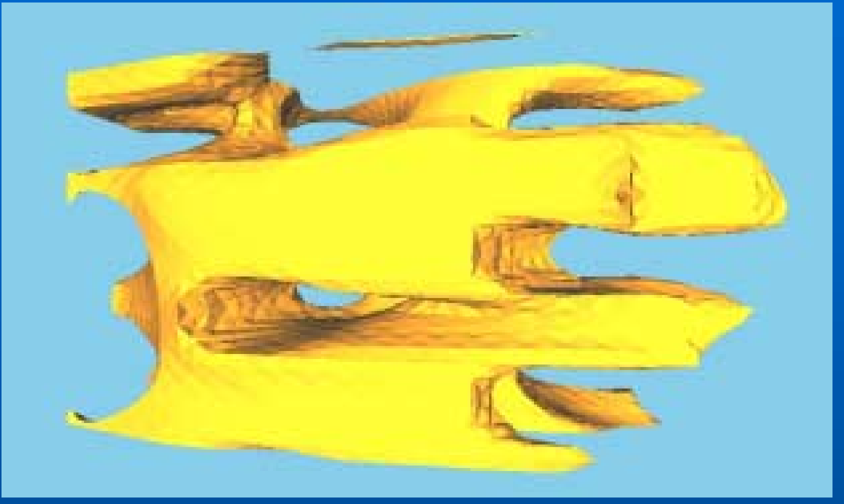
- **Two-step checking:**
  (1) Easy checking, pass if succeeds, else **(2) full checking**
  (**Easy checking only**: **correct**, faster, simplifying less)

*** Avoid Repeated Unsuccessful Tests** (while simplify more):

- If e fails in some test, then **put e aside**, and **later put e back to Q when** some other edge neighboring to e is collapsed that may **affect the checking result of e**

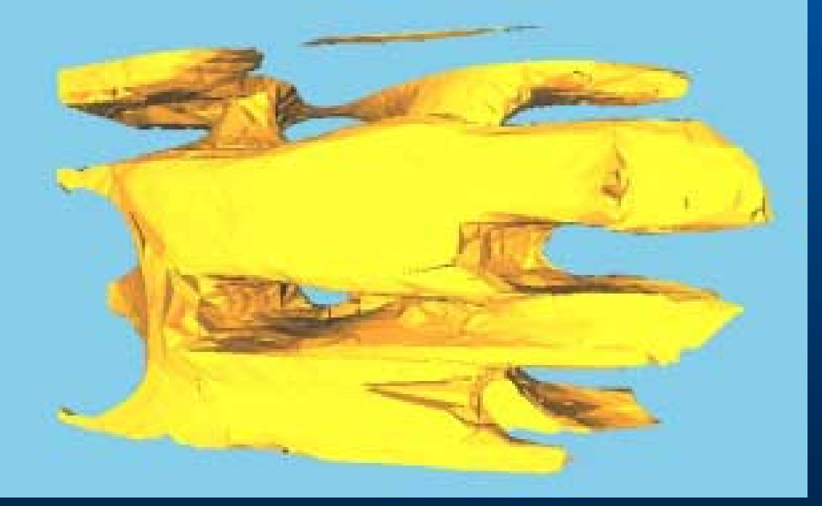
Results: Combustion Chamber

Error metrics:
g: edge length
s: scalar diff.
h: half-half



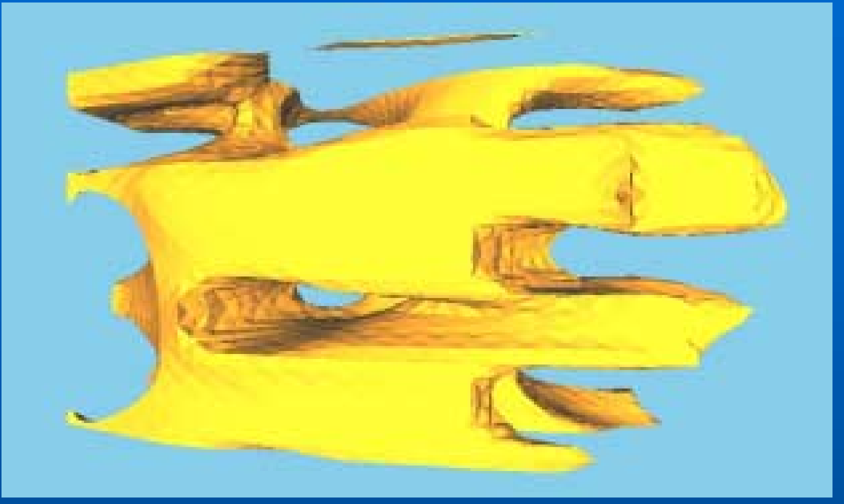
Original volume
isovalue = 0.251
38512 triangles

No-top; h, ε = 0.64; 7.74% cells left; 9944 tris



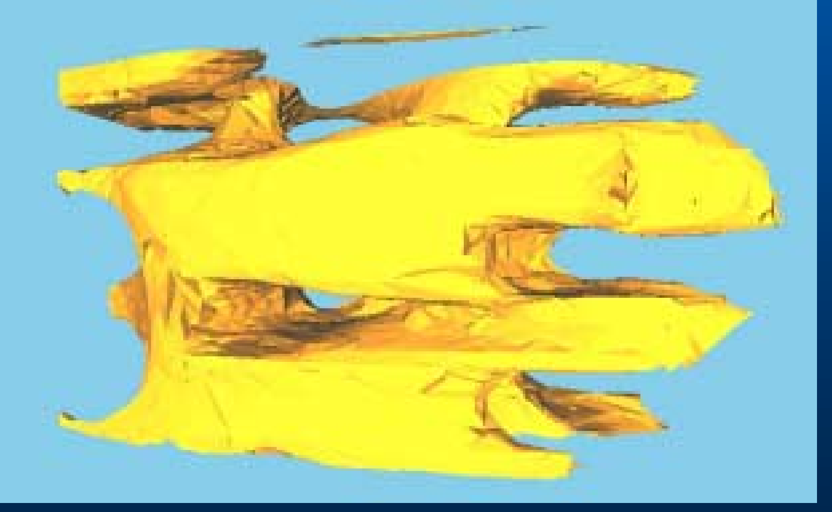
Our method; h, ε = 0.64; 62.05% cells left; 28986 tris

Results: Chamber (cont.)



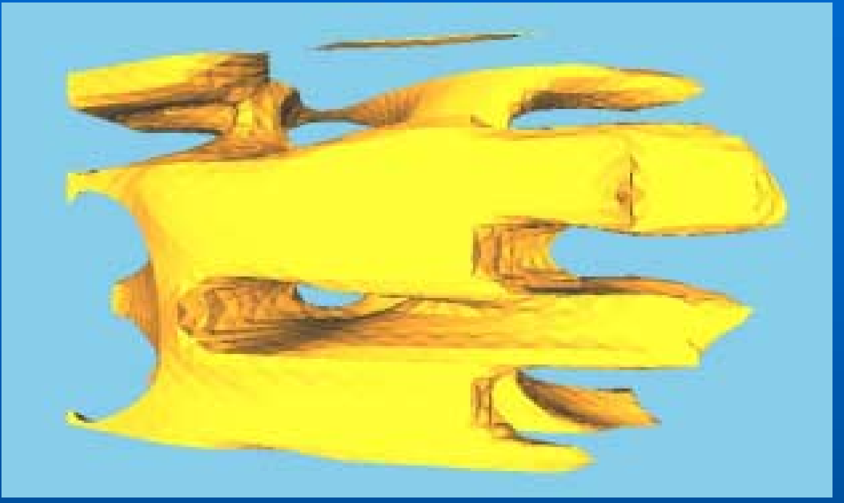
Original volume
isovalue = 0.251
38512 triangles

No-top; g, ε = 0.65; 21.36% cells left; 18085 tris



Our method; g, ε = 0.65; 63.73% cells left; 29276 tris

Results: Chamber (cont.)



Original volume
isovalue = 0.251
38512 triangles

Our method; h, ε = infinity;
51.82% cells left; 27099 tris

Our method; g, ε = infinity;
52.15% cells left; 27078 tris

Results: spx

Original; isovalue = 1.25;
850 triangles

No-top; g, ε = 0.61;
23.64% cells left; 161 tris

Our method; g, ε = 0.61;
85.48% cells left; 827 tris

Our method; s, ε = infinity;
82.01% cells left; 799 tris

Results: Statistics Summary

- Datasets: # vertices: 20108—211680; # cells: 12936—1005675
- Sun Blade 1000, dual 750MHz UltraSPARC III CPU

- Typically: < 5% critical vertices; 5—25% pure cells; thousands of top-eq regions (different isosurface topologies)
- Segmentation phase is very fast (0.81—68.96 seconds)

- Nice data-reduction rate (% removed cells): 48—89% when ε = infinity

- When using two-step checking, easy checking accounts for > 99.5% of successful criticality checkings

- When using easy checking only and omitting full checking, simplification phase runs 2—3 times as fast (17.41—1013.3 seconds for ε = infinity; 2.7 times as fast for the three longest runs), with almost the same data-reduction rate

Conclusions

- **The first volume simplification algorithm that preserves all isosurface topologies for rectilinear, curvilinear, and irregular grids**
- **Also preserves the geometry of the volume boundary, and avoids the fold-over problem**
- **Theoretical foundation; nice data-reduction rate; overall algorithm runs competitively fast**

Future Work:

- **Topological noises?**
- **Multi-resolution volume hierarchy for run-time isosurface extraction**

Acknowledgments

- Boris Aronov, Gunter Rote for discussions on critical points
- Han-Wei Shen for the Marching Tetrahedra code
- Han-Wei Shen, Claudio Silva, Peter Williams, Prashant Chopra, NASA, Vtk for the test datasets

- National Science Foundation (CAREER CCR-0093373, ACI-0118915, ITR CCR-0081964)