Topology Verification for Isosurface Extraction

Tiago Etiene
Luis Gustavo Nonato
Carlos Scheidegger
Julien Tierny
Valerio Pascucci
Tom Peters
Mike Kirby
Cláudio Silva

The University of Utah
“The need for verifiable visualization”

INTRODUCTION
Introduction

• Visualization is critical part of the scientific pipeline

• How can the visualization community build confidence on **Algorithms & Implementations** being used?
How to build confidence

• Consider the problem of Isosurface extraction. Common correctness tests:
  – Expert analysis
  – Visual comparisons
  – Benchmark suites
  – Comparison with “trustworthy” codes
  – Simple vs. Real world data
  – ....
Related work

• Visualization:
  – “The Need for Verifiable Visualization” [Kirby and Silva, 2008]

• Computational Simulation Community:
  – Validation & Verification [I. Babuska and J. Oden, 2004]
Traditional Scientific Simulation Pipeline

Validation – Does the mathematical model represent the physical phenomena correctly?

Verification – Does the computational model and its implementation represent the mathematical model accurately?
Validation  – Does the mathematical model represent the physical phenomena correctly?

Verification – Does the computational model and its implementation represent the mathematical model accurately?
Goal

Build tools for Topology Verification of Isosurfaces that can be implemented by anyone willing to verify their codes
Building confidence through verification

VERIFICATION & VISUALIZATION
Why verify topology?

• Do people care?
  – Medical data analysis
  – Numerical simulation

• Stacking complexity. Ex MC:
  – “Pure MC” – fairly simple (but people still get it wrong)
  – Asymptotic Decider – fairly simple + epsilon
  – Chernyaev MC33 – ok, now it's hard!
  – [Your method goes here]
The problem of verifying topology

• We can mathematically describe algorithm behavior
  – What are algorithm limits?
  – How to test limits?
  – What is the proper implementation behavior?
Algorithm behavior

• Example: VTK Marching Cubes
• Scalar field is a distance from a curve
Algorithm behavior

- What are the limits of the implementation being used?

Some configurations, as shown above, has more than one way of triangulating the square or cube (green and red components). VTK always chose the red component for faces and cubes and thus it cannot reproduce a tube.)
The problem of verifying topology

• Implementation: What can be tested?
  – Consistency:
    • Informally: “no holes”
    • Disk property
  – Correctness:
    • Informally: reproduces an underlying function/interpolant
    • Topology invariants
Consistency Verification

• Locally, every single vertex should have a link that is a disk

Good!  Bad!  Bad!
Correctness Verification

• Verification scheme
  – Compare the *expected* against the *obtained* topology invariant

• Euler characteristics

• Betti numbers

• Contour trees are powerful
  – But, is there an easier way to test?
Correctness Verification

- Two possible level-sets of a trilinear field...

- ... inside a single cube!
Verification using Stratified Morse Theory

- Euler characteristics using critical points

Given a volume date, we know all its critical points and thus we can compute the Euler characteristics.
Verification using Digital Surfaces

• Step 1 – User selects a grid size
Verification using Digital Surfaces

• Step 2 – Randomly create a scalar field
Verification using Digital Surfaces

- Step 3 – Create $S'$, a refined version of $S$.

No ambiguities allowed in $S'$. If such refinement is not possible, restart the process.
Verification using Digital Surfaces

• Step 4 – Mark digital surface
Verification using Digital Surfaces

• Step 5 – Extract Betti Numbers

Compute Betti numbers of Digital Surfaces

Extract mesh and compute Betti Numbers
Problem

- The grid should be refined until no *ambiguity* appears in the underlying grid. If such refinement is not possible, restart the process.

- Do we systematically remove some important cases?
  - Complicated cases may never be reached
  - Fortunately, this does not happen!
Problem

- Random Sampling vs. Our method
Observed implementation behavior

PRACTICAL RESULTS
Results

- Two techniques under verification:
  - Lewiner’s Marching Cubes 33
  - Tamal Dey and Josh’s Dellso

- Number of randomly generated scalar fields:
  - 10000 for Lewiner’s MC
  - 1507 for Dellso

<table>
<thead>
<tr>
<th></th>
<th>Failed Betti 0</th>
<th>Failed Betti 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lewiner’s MC</td>
<td>1.14%</td>
<td>1.61%</td>
</tr>
<tr>
<td>Dellso</td>
<td>27.0%</td>
<td>5.84%</td>
</tr>
</tbody>
</table>
Results

• Thomas Lewiner MC33:

Output from Isosurfacing  Expected Output  Trilinear surface
Results

• Tamal Dey and Josh’s Dellso:
  – Connected components
Results

- Tamal Dey and Josh’s Dellso:
  - Loops
Results

- Tamal Dey and Josh’s Dellso:
  - Holes
What we’ve learned

CONCLUSION & FUTURE WORK
Conclusion & Future Work

• Implementation of MMS:
  • Easy to code test: implementation is a black box

• Future work:
  – Isosurface extraction:
    • Build a more complete set of manufactured solutions
  – MMS in Visualization
    • Streamlines computation
    • Volume rendering
    • Mesh simplification
Thank you. Questions?
References


References


