Interactive Visualization Techniques for Complex Data Analysis

Claudio T. Silva

Visualization and Geometric Computing
Computer Science & Engineering
Polytechnic Institute of New York University
(joined July 1st, 2011)

Funded by grants/gifts from NSF, NIH, DOE, IBM, Sandia, LLNL, and ExxonMobil
Data Exploration through Visualization

It is hard to make sense out of large volumes of raw data, e.g., sensor feeds, simulations, MRI scans. Insightful visualizations help analyze and validate various hypotheses. But creating a visualization is a complex, iterative process.

J. van Wijk, IEEE Vis 2005
Ocean, Coastal, and Coastal-Margin Observatories...

... are EOFS for marine and estuarine systems

- Initiated in 1996, CORIE is a coastal margin observatory designed as a multi-purpose scientific and regional infrastructure for the Columbia River

- CORIE crosses traditional boundaries extending from freshwater to deep ocean

- CORIE has diversified user base and impact - consistent with regional needs

- NOAA Fisheries (NWFSC) has been a major partner

Recent perspectives:
- NEPTUNE
- PNW observatory/SATURN
- Coastal Storms Initiative
CORIE: Modeling System

Forcings
- Atmospheric forcings (wind, pressure, heat exchange)
- Eastern North Pacific tides
- River discharges
- Ocean circulation

Codes
- In-situ data (CORIE, ...)
- Remote sensing
- Bathymetry

Data assimilation

Simulations of circulation
- Typical Δt: 1.5 min
- Sediments
- Lower food web

Quality Controls

Circulation database
- Contemporary (Goal: 90-91, 96+)
- Pre-development
- Scenarios
- Calibration

Daily forecasts

Data products

Monday, November 28, 11
CORIE: Modeling System

Integrates key regions of interest
- Freshwater to deep ocean
- Emphasis on estuary and plume

Grid characteristics
- unstructured in horizontal
  - 30,001 nodes, 55,880 elements
- Z-coordinates in vertical
  - 62 vertical layers
- ~2-million total prism faces

Computational infrastructure
- 20 Intel dual-CPU nodes (2.4 Ghz, 4 Gb)
- 16TB primary storage
- Code parallelization in progress

Noteworthy
- ~3x faster than real time in a single CPU Intel processor, for \( \Delta t=1.5 \) min
- ~0.5 TB of storage for one-year simulation @ 15 min sampling
Salinity Intrusion
• High-quality rendering algorithms

Maximum Intensity Projection (MIP)  Full Volume Rendering
Visualization Infrastructure

- Improved visualization algorithms

![Graph showing unstructured volume rendering algorithms vs. hardware](image)

- Log Scale

- Tetrahedra/Second

- Year


- Algorithms
- Hardware
Visualization Infrastructure

• Visualization systems

VisTrails -- http://www.vistrails.org
Visualization Infrastructure

• Provenance Capture

VisTrails Plugin for ParaView
Systems: BirdVIS

• Collaboration with the Cornell Lab of Ornithology
• Collaboration with Marta Heilburn (Radiology, Utah)
• Visualizing trails in the Electronic Health Record with Timed Word Trees, a pancreas cancer use case
Data Exploration through Visualization

Insightful visualizations help analyze and validate various hypotheses. However, creating a visualization is a complex, iterative process.

Data Visualization User

J. van Wijk, IEEE Vis 2005
How to verify visualizations? algorithms?

- Two ingredients:
  - **Expected behavior**: mathematical description of algorithm properties
  - **Observed behavior**: output of the visualization technique
- Compare **expected** and **observed** behavior
  - If a mismatch occurs, something is wrong.
Application: Subject-Specific Modeling
How to verify visualizations?

How to verify visualizations?

3 Preliminary work: Verifying Isosurface Extraction Algorithms

We provide an instance of the verification process applied to isosurface extraction, a fundamental process in scientific visualization. Our rationale for selecting isosurface extraction for a case study is that these techniques are used daily for analyzing data ranging from medicine to engineering for making critical commerce-sensitive and sometimes life-altering decisions. These surfaces are defined as the set of all points in a scalar field $f$ (typically $f : \mathbb{R}^3 \to \mathbb{R}$), sharing a common value $k$: $f(x, y, z) = k$. Isosurfaces are used both as a visualization tool and as a preprocessing step in other techniques. Despite their fundamental importance, relatively little attention has been paid to determining the correctness of isosurface extraction codes. We describe next tools for verification of the geometry and topology correctness of isosurface extraction tools. For geometry verification, we borrowed ideas from CS&E while topology verification use ideas from CS community.

3.1 Geometry verification

Recently, we presented a verification methodology for isosurface extraction inspired by work in the CS&E community [67]. This technique is a step toward assessing the correctness of several publicly available isosurface extraction codes. We build our framework on top of the order of accuracy $5$. Monday, November 28, 11
Geometry verification

• Given an isosurface $f = k$ and assuming linear interpolation
  
  • Algebraic distance:
    \[
    \left\| f^h (\bar{x}) - f (\bar{x}) \right\| = O(h^2)
    \]
  
  • Normal (cross product of edges):
    \[
    \left\| \nabla f^h (\bar{x}) - \nabla f (\bar{x}) \right\| = O(h)
    \]
  
  • Gaussian curvature (angle deficit method):
    \[
    \left\| \kappa (f^h) - \kappa (f) \right\| = O(1)
    \]
  
  • Surface area:
    \[
    \left\| \text{area}(f^h) - \text{area}(f) \right\| = ?
    \]

• In practice:  
\[ E_h = \left\| u^h(x) - u(x) \right\| = ch^\alpha \]
\[ \log(E_h) = \log(c) + \alpha \log(h) \]
Observed order of accuracy Algebraic distance

\[ \log(L_\infty \text{ norm}) \]

- 2nd order slope
- Macet ($\alpha=0.98$)
- Dual Contouring ($\alpha=1.02$)

$h$
Observed order of accuracy Algebraic distance

\[ \log(L_\infty \text{ norm}) \]

\[ h \]

- 2nd order slope
- Macet \((\alpha=0.98)\)
- Dual Contouring \((\alpha=1.02)\)
Observed order of accuracy Algebraic distance

\[ \log(L_\infty \text{ norm}) \]

\[ h \]

- 2nd order slope
- Macet (\(\alpha=0.98\))
- Dual Contouring (\(\alpha=1.02\))
Observed order of accuracy Algebraic distance

\[ \log(L_{\infty} \text{ norm}) \]

- 2nd order slope
- Macet ($\alpha=0.98$)
- Dual Contouring ($\alpha=1.02$)

$h$

Monday, November 28, 11
Observed order of accuracy Algebraic distance
Observed order of accuracy Algebraic distance

\[ \log(L_{\infty} \text{ norm}) \]

\[ h \]

2nd order slope
Macet \((\alpha=0.98)\)
Dual Contouring \((\alpha=1.02)\)
Observed order of accuracy Algebraic distance

Fixed

\( \log(L_\infty \text{ norm}) \)

\( h \)

2nd order slope
Macet (\( \alpha=0.03 \))
Dual Contouring (\( \alpha=1.96 \))
Visual Analysis -- User

Before

After

Before $h$, $h/2$, $h/4$, $h/8$
Observed order of accuracy Normal

![Graph showing log(L_infinity norm) vs. h for different methods.](image)

- **1st order slope**
- **Macet (\(\alpha=-0.12\))**
- **Dual Contouring (\(\alpha=-0.11\))**

Monday, November 28, 11
Observed order of accuracy Normal

\[
\log(L_{\infty} \text{ norm})
\]

\[
\begin{aligned}
&1 \\
&0.1 \\
&0.01 \\
&0.001 \\
&\text{h}
\end{aligned}
\]

- 1st order slope
- Macet (\(\alpha=0.75\))
- Dual Contouring (\(\alpha=0.96\))
Topology verification

TO BE PRESENTED AT 4:30pm THIS AFTERNOON
Results

- Rate of mismatch for topology invariants for algorithms without topological guarantees

<table>
<thead>
<tr>
<th>Consistency (%)</th>
<th>Digital Topology (%)</th>
<th>SMT (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Betti 0</td>
<td>Betti 1</td>
</tr>
<tr>
<td>VTKMC 0</td>
<td>27.6</td>
<td>23.2</td>
</tr>
<tr>
<td>Afront 0</td>
<td>35.9</td>
<td>22.8</td>
</tr>
<tr>
<td>Macet 0</td>
<td>54.3</td>
<td>20.9</td>
</tr>
</tbody>
</table>
In this paper, we are interested in how to effectively verify topological properties of codes which employ trilinear interpolation. The verification of these properties is critical for ensuring the correctness of visualizations derived from scientific data. We have developed and implemented verification tools that utilize techniques from digital topology and stratified Morse theory to detect implementation problems.

The isosurface evolution of level-sets of simply connected domains is a common task in scientific visualization. Contour trees are powerful structures to describe the topology of these surfaces. They naturally provide a checkable certificate of correctness for the isosurfaces.

However, the construction of contour trees can be computationally expensive, especially for high-dimensional data. Our approach is based on regular grids, which allows for a divide-and-conquer strategy. The process involves walking through diagonals of adjacent cubes to determine the digital topology. In the case of regular-grids, this is straightforward. However, for general grids, the process is more complex due to the presence of ambiguous cubes.

We have compared our approach with other methods that use SMT and DT for topological verification. While these methods are capable of finding unexpected behavior and coding mistakes, our approach is simpler and more effective in practice. For instance, our method is more likely to be adopted during development of visualization software.

We have conducted experiments to compare the number of mismatches found by each approach. Even when other tools, such as contour trees, are available, our approach is likely to be used along with them. Our experiments have shown that the theoretical guarantees supporting our method are strong, and we expect it to be used as an integral part of the verification cycle.

Acknowledgments
We would like to acknowledge the support of the National Science Foundation for this research. We also thank our colleagues for their contributions to this project.
Results

- Rate of mismatch for topology invariants for "topologically correct" algorithms

<table>
<thead>
<tr>
<th>Consistency (%)</th>
<th>Digital Topology (%)</th>
<th>SMT (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Betti 0</td>
<td>Betti 1</td>
</tr>
<tr>
<td>MC33</td>
<td>0</td>
<td>2.4</td>
</tr>
<tr>
<td>Dellso</td>
<td>19.1</td>
<td>24.4</td>
</tr>
<tr>
<td>MCFlow</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
“Efficient implementation of Marching Cubes with topological guarantees” -- missing case in original theorem
Lessons

• Debugging, even “visual debugging” is hard, and incomplete

• Visual inspection is required but not sufficient

• Important codes and algorithms are still “incomplete” despite 20+ years of development

• Formal methods (of verification) used with parameter studies can greatly improve the quality of codes and techniques

• Each visualization & analysis technique needs their own analysis