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
The graph illustrates the growth of information storage and human communication over time.

- **All Disk Storage**: Steady increase with a slope indicating consistent growth.
- **All Human Documents (40k Yrs)**: A horizontal line, suggesting a stable or slowly increasing number of documents.
- **Amount Can Store In Human Minds in 1 Yr**: A horizontal line at a lower level than disk storage, indicating limited human cognitive capacity.
- **All Info/Yr** and **Unique Info/Yr**: Both show exponential growth, with Unique Info/Yr growing even faster than All Info/Yr.

Sources: Lesk, Berkeley SIMS, Landauer, EMC
Data Exploration through Visualization

Data \rightarrow Visualization \rightarrow Image \rightarrow Perception & Cognition \rightarrow Knowledge

Data \rightarrow Specification \rightarrow Exploration

J. van Wijk, IEEE Vis 2005

New York University

Polytechnic Institute of New York University

Monday, November 28, 11
... 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
- Ocean circulation
- Short waves

Forcings
- River discharges
- In-situ data (CORIE, ...)
- Remote sensing
- Bathymetry

Codes

Data assimilation

Simulations of circulation

Quality Controls
- Sediments
- Lower food web
- ...

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

Typical Δt: 1.5 min

Daily forecasts

Data products
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
Fresh Water Plume
Visualization Infrastructure

- High-quality rendering algorithms

Maximum Intensity Projection (MIP)  Full Volume Rendering
• Improved visualization algorithms

Unstructured Volume Rendering Algorithms vs. Hardware
Log Scale

- Projected Tetrahedra
- Incremental Slicing
- GATOR
- HW Ray Casting
- HAVS

Tetrahedra/Second

Year


Algorithms
Hardware
Visualization Infrastructure

- Visualization systems

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

- Provenance Capture

VisTrails Plugin for ParaView
Coupling EEG and Cognitive Load

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

- Data
- Visualization
- Specification
- Image
- Perception & Cognition
- Exploration
- Knowledge

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?

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 $f: \mathbb{R}^3 \rightarrow \mathbb{R}$), 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 $5$. 

Figure 3: Different approaches to verification. Left figure illustrates the state exploration approach used in CS. The middle figure shows the systematic grid refinement used in CS&E. The right figure shows both geometry and topology verification procedures for isosurface extraction algorithms. In order to verify geometry, systematic refinement is used in the same spirit as in CS&E. On the other hand, topology verification resembles the state exploration approaches used in CS because it requires exhaustive state exploration of all possible cases of a trilinear interpolant inside a given cell and its neighbors for Marching Cubes algorithms. Rightmost figure was adapted from [35].

Kirby and Silva suggest for “Verifiable Visualization” [109].
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:

\[ E_h = \| u^h (x) - u(x) \| = ch^\alpha \]

\[ \log(E_h) = \log(c) + \alpha \log(h) \]
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

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

\( h \)

- 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 \((\alpha=1.02)\)

Monday, November 28, 11
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

![Graph showing the observed order of accuracy for algebraic distance with different methods: 2nd order slope, Macet ($\alpha=0.98$), and Dual Contouring ($\alpha=1.02$).]
Observed order of accuracy Algebraic distance

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

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

\(h\)

(Fixed)
Visual Analysis -- User
Observed order of accuracy Normal

(Bug)

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

\[ h \]

- 1st order slope
- Macet \((\alpha = -0.12)\)
- Dual Contouring \((\alpha = -0.11)\)
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</td>
<td>0</td>
<td>27.6</td>
</tr>
<tr>
<td>Afront</td>
<td>0</td>
<td>35.9</td>
</tr>
<tr>
<td>Macet</td>
<td>0</td>
<td>54.3</td>
</tr>
</tbody>
</table>
The idea of verification through manufactured solutions is a fundamental tool to ensure the correctness of visualization algorithms. It involves generating a known, theoretically correct result and comparing it with the output of the algorithm to be verified.

**Concrete Example -- VTK MC**

The diagram illustrates a specific example related to VTK (Visualization Toolkit) MC (Marching Cubes) algorithm. This algorithm is used for isosurface extraction, which is a method for visualizing 3D data by extracting surfaces that correspond to a particular value of a field function.

In this context, the diagram shows a cube and a series of points connected by dotted lines, possibly representing a process of extracting an isosurface. The cube likely symbolizes the 3D data grid, and the points and lines are part of the isosurface extraction algorithm, demonstrating how the algorithm works in practice.

**Conclusion and Future Work**

Theoretical guarantees and empirical evidence suggest that verification is a useful tool for researchers and developers to ensure the reliability and correctness of their visualization algorithms. Further investigation and development of scientific visualization infrastructure can benefit from adopting verification as an integral part of the process.
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