Programming Language Interfaces for Time-Varying Multivariate Visualization

Jian Huang (presented by Wesley Kendall)
The University of Tennessee, Knoxville
SciDAC Institute of Ultrascale Visualization
What is a feature?
What can you show me?

• A critical gap:
  – What do you want to see?
  – Show me what ever you find then.

• Too many variables to look at side by side
• Too many time steps to examine one by one
• Too many models/run – to compare/contrast
What can you show me?

• A critical gap:
  – Often scientists know what they want to see
  – But cannot provide a formal quantitative description
Extreme Complexity $\rightarrow$ Ultrascale

- Scientific visualization faces problems more complex than ever before by orders of magnitude
  - Complexity: carbon, biogeochemical, evolution, coupling
  - Number of variables: $>100$
  - Temporal resolution + span: every 3 min, 1000 years ($1.75e8$)
  - Spatial resolution: 22km $\rightarrow$ 1km
  - Size of ensemble runs: 50 $\rightarrow$ 1000
User Concepts

- Qualitative user concepts:
  - When does the growing season start?

- Domain specific programming language methods
  - Specify events in an expressive, concise and powerful way

- Any persistent trends of event changes
  - Has the beginning of the growing season shifted in time in recent decades? How are different locations affected?
Concept-Driven Visualization

- As visual summaries, the benefits are:
  - Data reduction
  - Semantic meaning
  - Focus
  - Easily multivariate and temporal
  - Iteratively refined and recorded
Relational Patterns in Local Distribution

1. Define neighborhood
2. Establish relevant data ranges
3. Draw up clauses

STEP 1: define neighborhood

STEP 2: select distribution intervals

STEP 3: formulate clauses

- mean(A) > mean(B)
- skewness(B) < -100
- covar(A, B) > 0
Spatial Neighborhood Query

With typical 1D transfer function

Neighborhood Query
freq(nonbackground) > freq(background)
Temporal Neighborhood Query

Positive and negative covariance between two timesteps
Fuzzy Matching

We want to show locations that:
– match to a degree (score $\rightarrow$ opacity)
– match a subset of inequalities (combination $\rightarrow$ color)
Evaluating a Query

- For each location
  - For each clause
    - If TRUE
      \[ \text{score} = 1 \]
      \[ \text{tag bit set} \]
    - Otherwise
      \[ \text{score} = f(\text{distance}) < 1 \]
Visualizing a Query

- *All* locations scored and the rendered
- Sum of clause scores $\rightarrow$ opacity
- Clause bitfield $\rightarrow$ color
  - Bitfield indexes into colormap on the GPU
  - $2^{|\text{clauses}|}$ possible bitfield configurations

![Color Map](image)

clause 0 and 2 met
mean temperature and precipitation between decades 2000-2009 and 2090-2099
Specifying Temporal features

- User concepts about temporal events are often “story” like

- Uncertainty expressed via regular expression
  - 

- Modeled after *regex*, but need to answers where and when an event occurs
TimeMarks

For example: $[-.4, .4] \times T[[.4, \max]]?\times$

- For each location, find time step $T$ sandwiched between zero or more changes in $[-.4, .4]$ and at least one change of more than .4

- $T$ – TimeMark: when event occurs
- Automatic expansion into substantiated queries
- Combine primitives in time sequence
“Green-Up”: Northern Hemisphere colored by month of event in variable ELAI.

\([-.4, .4]*T[.4, \text{max}]*\)
“First Snow”: Northern Hemisphere colored by month of event in variable FSNO

???[min, 0.7]*T[0.7, max]*
Another look in parallel coordinates

“Green-Up” in 2050
Northern Hemisphere colored by month of event in variable ELAI:
[-0.4, 0.4] * T[0.4, max]?*
Complexity of Meta Queries

- Many cases could lead to exponential problem spaces
- Fortunately, the data access patterns are not random (except in rare cases)

<table>
<thead>
<tr>
<th>ID</th>
<th>Query</th>
<th># Queries generated</th>
<th># Data blocks accessed</th>
<th># Voxels returned</th>
<th>Running time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[0-10][0]?[0-1e10]*</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>3.168</td>
</tr>
<tr>
<td>2</td>
<td>[0-99][0]?[0-1e10]*</td>
<td>1</td>
<td>41</td>
<td>3</td>
<td>26.522</td>
</tr>
<tr>
<td>3</td>
<td>[40-60]??[0-1e10]*</td>
<td>1</td>
<td>10</td>
<td>342</td>
<td>6.067</td>
</tr>
<tr>
<td>4</td>
<td>[50]?*</td>
<td>71</td>
<td>1</td>
<td>3,615,888</td>
<td>7.454</td>
</tr>
<tr>
<td>5</td>
<td>[70]?[-5-1e10]<em>[5--1e10]</em></td>
<td>71</td>
<td>1</td>
<td>6,584</td>
<td>7.485</td>
</tr>
<tr>
<td>6</td>
<td>[0-20]??[0-1e10]<em>?</em></td>
<td>71</td>
<td>10</td>
<td>3,593,696</td>
<td>159.97</td>
</tr>
<tr>
<td>7</td>
<td>[80]?[-100-1e10]<em>[100--1e10]</em>[-100-1e10]*</td>
<td>2,556</td>
<td>1</td>
<td>16,994,091</td>
<td>248.87</td>
</tr>
<tr>
<td>8</td>
<td>[80-82]?[-100-1e10]<em>[100--1e10]</em>[-100-1e10]*</td>
<td>2,556</td>
<td>3</td>
<td>50,565,859</td>
<td>757</td>
</tr>
<tr>
<td>9</td>
<td>[0-99]?[-100-1e10]<em>[100--1e10]</em>[-100-1e10]*</td>
<td>2,556</td>
<td>≈117,500</td>
<td>≈1,685,529,000</td>
<td>≈72,500</td>
</tr>
</tbody>
</table>
Concept-Driven Visualization

• Required infrastructural support:
  – Parallelism
  – Scalable data structures
  – Optimal use of parallel I/O
Backend Technical Requirements

• Underlying data structures and management need to be optimized for common data types in scientific research.
  – Time-varying, multi-dimensional, multi-variate, potentially non-uniform grids.

• Data management systems (DMS) for massive data sets must ...
  – incur small storage costs,
  – provide ad hoc query support,
  – provide support for application-native scientific data
  – exhibit reasonable latency and throughput performance.

• Implications of these requirements are ...
  – no unnecessary data duplication,
  – a transparent, self-explanatory query structure,
  – use of sophisticated underlying data structures and algorithms.
Backend Technical Requirements

- Simplistic queries are not sufficient to describe features / subsets.
- Many features can generally be described as local events, i.e. spatially and temporally limited regions with characteristic properties in value space.
- Scientists know what they are looking for in their data, but may be unable to formally or definitively describe their concept, especially when based on partially substantiated knowledge.
- Scientists need to query and extract such features or events directly without having to rewrite their hypothesis into an inadequately simple query language.
- A more sophisticated feature-oriented query language is required.
Related Work

• Large data management in visualization
  – Data partitioning (blocks, “bricks”)
  – Efficient searching using tree-based data structures:
    • Interval tree, k-d tree quad-tree, octree, etc.
    • Bitmap indexing
  – Relational Database Management Systems (RDBMS)

• (Programming) languages in visualization:
  – More versatile and flexible compared to GUls.
  – Alter GPU shader programs on the fly: “Scout”
  – VTK provides Tcl/Tk and Python bindings.
Data Organization

- Large data sets need to be partitioned for data distribution and load-balancing.

- Break up data set into data items containing
  - spatial and temporal location \((x,y,z,t)\),
  - a value for each data variable.

  e.g. \(\{x=1; \ y=2; \ z=3; \ t=10; \ \text{density}=2.7; \ \text{entropy}=.7\}\)

- Implications
  - Yields increase in total data size!
  - Number of data items can be enormous!
  - **But:** Load-balancing can be applied on the level of data items.
Query-Driven Visualization

- **Load-balancing** by breaking up locality within the dataset.
- Optimized data access by using a **B-tree like structure** to skip irrelevant data items on top of a linear search.
- **Discard unwanted data items** upon distribution (data items are independent of any structural meta-information)
- **Compress blocks of data items** to trade memory space vs. access time, decompress on access.
Data Selection

- Each data server hosts a portion of the data set as data items in a *sorted* list.

- On top, a complete M-ary search tree of depth $N << M$ (e.g. $M = 256$, $N = 3$) indexes into the list of data items.

- Search: Find first matching data item and initialize a *linear search* from it. Use search tree to skip irrelevant groups of data items.
Enhancements

• **Observation**: Data items are independent of any structural meta-information (e.g. a grid).
  – Unwanted data items can be deleted before distribution to data servers.
  – This counterbalances the increase of data set size.

• **Compress the linear list of data items**.
  – Trade-off: memory space vs. access time
  – Blocks of data items are decompressed on the fly.
  – Since linear list is sorted, high compression rates (20:1) are possible in many cases.
Scalability Tests: the data

- NASA's Moderate Resolution Imaging Spectroradiometer (MODIS) database, continuously updated
- Use 417 timesteps, 8-day interval, 02/2000 to 02/2009
- 500 meter resolution sampling of North and South America, creating a 31,200x21,600 grid
- Compute variables from 7 wavelength bands
- Use MRT toolkit to reproject from sinusoidal grid to equirectangular grid
- Total data used for scalability tests amount to 1.1TB
Scalability Tests: the machine

- Jaguar, ORNL
- Cray XT4 consisting of 7,832 quad-core 2.1 GHz AMD Opteron processors with 8 GB of memory.
- 31,328 cores with over 60 TB of main memory.
- Lustre parallel file system. One meta data server (MDS), 72 OSSs (I/O nodes), 144 OSTs (physical disk systems)
Infrastructural Diagram
Measured I/O Bandwidth
Distribution and Query Overheads

**Distribution Overhead**

**Average Time Per Query**
Climate Science - Drought Assessment

- NDVI and NDWI can be good indicators of drought, and NDDI (Normalized Difference Drought Index) can be computed by
  \[ \text{NDDI} = \frac{(\text{NDVI} - \text{NDWI})}{(\text{NDVI} + \text{NDWI})} \]

- We use a similar method of drought assessment by querying for:
  \[ \text{NDVI} < 0.5 \text{ and } \text{NDWI} < 0.3 \]

- After queries are issued, result is sorted in spatial order

- Temporal overlap can then be computed
  - look for \( 0.5 < \text{NDDI} < 1 \) for at least 4 timesteps (1 month) in a row

- We also placed one more restriction and throw out the areas where the event happened more than one time, finding only the areas where abnormal drought conditions occur
Drought Assessment

Figure 6: Drought analysis of NDVI < 0.5, NDWI < 0.3, and NDDI between 0.5 and 10. These are periods of drought that lasted for at least a month and occurred up to two years for any given region. Several regions are marked where drought has happened, most notably the 2006 Mexico drought. The image was colored based on the year that the longest drought occurred.

<table>
<thead>
<tr>
<th>Cores</th>
<th>Read</th>
<th>Write</th>
<th>Filter</th>
<th>Distribution</th>
<th>Query</th>
<th>Sort</th>
<th>Analysis</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1K</td>
<td>85.23</td>
<td>0.68</td>
<td>47.11</td>
<td>22.40</td>
<td>1.01</td>
<td>14.17</td>
<td>0.22</td>
<td>170.82</td>
</tr>
<tr>
<td>2K</td>
<td>48.62</td>
<td>0.30</td>
<td>24.49</td>
<td>14.19</td>
<td>0.71</td>
<td>11.31</td>
<td>0.21</td>
<td>99.83</td>
</tr>
<tr>
<td>4K</td>
<td>41.27</td>
<td>0.31</td>
<td>11.80</td>
<td>8.78</td>
<td>0.34</td>
<td>9.18</td>
<td>0.10</td>
<td>71.78</td>
</tr>
<tr>
<td>8K</td>
<td>48.78</td>
<td>0.32</td>
<td>6.30</td>
<td>5.60</td>
<td>0.16</td>
<td>6.50</td>
<td>0.08</td>
<td>67.74</td>
</tr>
<tr>
<td>16K</td>
<td>46.56</td>
<td>0.30</td>
<td>3.45</td>
<td>3.02</td>
<td>0.08</td>
<td>4.20</td>
<td>0.07</td>
<td>57.68</td>
</tr>
</tbody>
</table>

Table 1: Timing results (in seconds) of the drought application.
Climate Science: Time Lag Analysis

- Studying time lag is important for obtaining a better understanding of how variables like NDVI are affected by other conditions.
- Studying past and present droughts in relation to these conditions could enhance the capability to develop early warning systems.
- In our example, we compute the time lag between when NDWI first occurs in the 0.7 - 0.9 range (first snow) and when NDVI first occurs in the 0.4 - 0.6 range (vegetation green up).

![Time-Lag of First Snowfall and Start of Green-Up](image)

Example of time lag of first snowfall and vegetation green up.
Time Lag Analysis

Figure 8: Time-lag between the first occurrence of $0.7 < \text{NDWI} < 0.9$ and the first occurrence of $0.4 < \text{NDVI} < 0.6$ in 2006. The color represents the length of the time-lag. Calculating this allowed us to assess the duration from the first snow to the beginning of vegetation green-up.

<table>
<thead>
<tr>
<th>Cores</th>
<th>Read</th>
<th>Write</th>
<th>Filter</th>
<th>Distribution</th>
<th>Query</th>
<th>Sort</th>
<th>Analysis</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1K</td>
<td>80.63</td>
<td>1.96</td>
<td>47.90</td>
<td>24.20</td>
<td>6.11</td>
<td>26.21</td>
<td>0.25</td>
<td>187.26</td>
</tr>
<tr>
<td>2K</td>
<td>51.54</td>
<td>2.03</td>
<td>22.22</td>
<td>18.90</td>
<td>3.01</td>
<td>20.25</td>
<td>0.15</td>
<td>118.10</td>
</tr>
<tr>
<td>4K</td>
<td>45.48</td>
<td>2.05</td>
<td>12.60</td>
<td>12.56</td>
<td>1.64</td>
<td>15.30</td>
<td>0.08</td>
<td>89.71</td>
</tr>
<tr>
<td>8K</td>
<td>47.43</td>
<td>2.04</td>
<td>7.00</td>
<td>7.14</td>
<td>0.90</td>
<td>8.90</td>
<td>0.05</td>
<td>73.46</td>
</tr>
<tr>
<td>16K</td>
<td>46.34</td>
<td>2.06</td>
<td>3.65</td>
<td>5.02</td>
<td>0.51</td>
<td>4.80</td>
<td>0.05</td>
<td>62.43</td>
</tr>
</tbody>
</table>

Table 2: Timing results (in seconds) of the time-lag application.
Conclusion

• Creating single images as summarizing visualization of a high-level event to study climate change

• Feature specification that empowers “eye-balling” should be studied in depth, in addition to feature extraction and rendering

• Programming language type of methods have offered encouraging results

• It is crucial to have truly scalable parallel infrastructure for visualizing terascale data and beyond
References

• Wesley Kendall, Markus Glatter, Jian Huang, Tom Peterka, Robert Latham and Robert Ross, *Terascale Data Organization for Discovering Multivariate Climatic Trends*, SC'09, November 2009, Portland, OR.


Acknowledgements

• Current Students:
  – Wesley Kendall

• Graduated students:
  – Dr. Markus Glatter, Dr. C. Ryan Johnson, Dr. Rob Sisneros, Dr. Josh New
  – Brandon Langley, Colin Mollenhour

• Collaborators DOE SciDAC Ultravis Institute (www.ultravis.org)
  – Rob Ross, Tom Peterka, Kwan-Liu Ma, Han-Wei Shen, Ken Moreland, John Owens

• Collaborators at Oak Ridge National Laboratory
  – Sean Ahern, Forrest Hoffman, David Erickson.

• Our funding were provided by DOE SciDAC, DOE Early Career PI Award, NSF ACI and CNS programs.