State-of-the-Art Visualization Techniques for Gleaning Insights in Large Time-varying Volume Data

Kwan-Liu Ma
University of California at Davis

Presentations

• Recent research results in time-varying data visualization
• Production visualization
  – High-performance computing environment
  – Desktop solutions
Not included

- Commercial products
- Open source projects
- Non-volume data

Ask questions
From You

• New challenges
  – Data size, type, and complexity
  – Unique computing environments
  – New applications

• Basic requirements
  – Visualization operations
  – Process of data analysis and visualization

• Others

Schedule

08:30-08:45 Overview
08:45-09:20 High-performance visualization techniques
  Kwan-Liu Ma, UC Davis
09:20-10:00 Feature extraction and encoding
  Han-Wei Shen, Ohio State University
10:00-10:30 coffee break
10:30-11:00 Multimodal visualization process
  Dave Modl, LANL
11:00-11:30 Desktop techniques
  John Clyne, NCAR
11:30-12:00 Open Discussion
Time-Varying Data Visualization

Overview

Time-Varying Data

- Large
  - Several hundreds to thousands time steps
  - Tens of million points to several billion points (regular-grid data)
  - Tens of variables

- Data transport requirements
  - 100MB-10GB per time step per variable

- Desire to browse and explore the spatial, temporal, and variable domains

- Irregular mesh, dynamic mesh!
Solutions

• Data reduction
  – Subset
  – Compression
  – Feature extraction
• Parallel visualization
• Simulation-time visualization
Data Reduction

- **Value-based encoding**
  - Lossy?
  - Space, time, or both (4D)
    - Time [Shen and Johnson ’94][Shen et al. ’99][Lum et al. 2001]
    - Space [Westerman ’95][Schneider and Westermann 2003]
    - 4D [Wilhelms and Van Gelder ’94][Linsen et al. 2002]
  - Multivariate data [Fout et al. 2005]

- **Feature-based methods**
  - Physically based
  - A modeling problem
  - Intelligent system approach [Tzeng and Ma 2005]

- **Multiresolution, multiscale?**
- **Postprocessing or simulation-time?**

Parallel Visualization

- **Distributing both data and calculations**
- **Preprocessing**
  - Resampling and filtering
  - Derived properties
  - Statistical data
  - Feature extraction
  - Partitioning and packing
- **Rendering**
  - Geometry-based, cell-based, voxel-based
- **Scalability**
Simulation-Time Visualization

• Move data to a dedicated visualization server
  – Simple
  – Scale?
• Do not move the data
  – Shared data structures
  – Rendering cost?
  – Simulation cost?

Outline

• Parallel Pipelined Rendering
• A Parallel I/O Strategy
• Intelligent Feature Extraction and Tracking
A Parallel Pipelined Renderer

- A postprocessing visualization facility
- Parallel pipelining to maximize processor utilization and hide I/O cost
- Image compression to cut down image transport cost over a wide-area network

Parallel Pipelined Rendering

Intra-volume parallelism

- No pipeline effect
- Long inter-frame delay
- High rendering rates but the frame rates can be bad
Parallel Pipelined Rendering

Inter-volume parallelism

- Limited by local memory size
- Low rendering rates
- Out-of-order results

Parallel Pipelined Rendering

Inter-volume + Intra-volume parallelism

- Partition $P$ processors into $L$ groups
- Pipelined rendering of $L$ volumes
- Choose $L$ carefully to achieve the optimal frame rates
Parallel Pipelined Rendering

128 time steps $256^3$ data set, 256x256 pixels on RWCP SCore II
Efficient Image Transport

- Compression plays a significant role
- Lossy versus lossless compression
- Compression with reasonable speed
- Parallel compression vs. serial compression
- Frame-to-frame coherence?
- Rapid decompression

Discussion

- Optimal for batch mode but not for interactive browsing
- Alternative solutions?
  - High performance storage and network
    STORCLOUD 2005
  - Parallel I/O
    - MPI I/O
    - Data sieving for fetching nonconsecutive data
    - Collective I/O for merging I/O requests from processors
An I/O Strategy for Parallel Rendering
Time-Varying Volume Data

Problem Description

- Postprocessing visualization
- Parallel visualization of earthquake simulation
- Transmitting 400+MB per time step from disk to the parallel computer for the 100 million cells simulation
- Preprocessing calculations are needed
- Parallel file systems
- Software rendering
When data size is small:

- $T_f$
- $T_p$
- $T_s$
- $T_r$

When data size is large:

For 400 MB / timestep: $T_f + T_p > 20$ seconds

Using 128 Rendering Processors: $T_r < 1$ second
The best performance can be achieved, if one input processor can finish fetching and preprocessing during the other (m-1) input processors sending their data.

\[ T_f + T_p = T_s (m - 1) \rightarrow m = \frac{T_f + T_p}{T_s} + 1 \]

\[ \text{Rendering Processors} \]

\[ \text{Hide the I/O cost successfully.} \]
1D Input Processors (1DIP)

• When $T_r < T_s$, no matter how many input processors are used,

![Diagram showing the relationship between rendering and waiting times for 1D input processors.]

• Basic Solution

Use several Input Processors to send one timestep to reduce $T_s$

2D Input Processor (2DIP)

• Use $n$ groups of $m$ input processors

![Diagram showing the relationship between rendering and waiting times for 2D input processors.]

$m$ processors fetch, preprocess, and distribute a single time step
Test

LeMieux
• At Pittsburg Supercomputing Center (PSC)
• HP/Compaq AlphaServer with 3,000 processors

Data
• Hexahedral cells
• Scalar fields
  – 100 M cells
• Time-varying
  – 800 timesteps
  – 400 MB per timestep per variable

Test Result (1) – 1DIP

![1D input processor strategy for rendering 512*512 image](image)

64 rendering processors

Using the actual values of \( T_f \), \( T_p \), and \( T_r \), we obtain m:

\[
\frac{(12.32 + 9.06)}{2.0} + 1 = 11.69
\]
Test Result (2) – 1DIP

1D input process strategy for rendering 1024*1024 image

- $T_r \uparrow$, $m \downarrow$

Test Result (3) – 1DIP

Rendering 512*512 image using simple preprocessing

- $T_p \downarrow$, $m \downarrow$
Test Result (7) – 1DIP vs. 2DIP

128 rendering processors

Test Result (6) – 2DIP sending time

Number of Input Processors for Sending One Timestep

Sending time
Discussion

• I/O bottlenecks are effectively removed and preprocessing cost is hidden
• Rendering time dominates inter-frame delay and near-interactive visualization is achieved
• Relying on parallel file systems
• Limited by the maximum bandwidth
• Adaptive rendering and prefetching
Intelligent 4D Feature Extraction and Tracking

Feature Extraction Problems

- High dimensional transfer function!
- Data values of a feature of interest can vary over time
Transfer Function Space Extraction

- An intelligent system approach
Cumulative Histogram

Cumulative_Histogram(x) = \int_0^x \text{Histogram}(t) dt

Transfer Function Space Extraction

- An intelligent system approach
Artificial Neural Networks

- Mimic the processes in biological neural networks
- Can learn from a given set of data
- An trained ANN represents a high-dimensional non-linear function

![Artificial neural network diagram](image)

“Texture of the Nervous System of Man and the Vertebrates” by Santiago Ramón y Cajal

Results

<table>
<thead>
<tr>
<th>time</th>
<th>Transfer function 1</th>
<th>Transfer function 2</th>
<th>Transfer function 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>t=195</td>
<td><img src="image" alt="key frame" /></td>
<td>![image]</td>
<td>![image]</td>
</tr>
<tr>
<td>t=210</td>
<td>![image]</td>
<td>![image]</td>
<td>![image]</td>
</tr>
<tr>
<td>t=225</td>
<td><img src="image" alt="key frame" /></td>
<td>![image]</td>
<td>![image]</td>
</tr>
<tr>
<td>t=240</td>
<td>![image]</td>
<td>![image]</td>
<td>![image]</td>
</tr>
<tr>
<td>t=255</td>
<td>![image]</td>
<td>![image]</td>
<td>![image]</td>
</tr>
</tbody>
</table>
Feature Tracking

• Help the user to follow the temporal progression of a select set of features

Data Space Extraction

• Higher-dimensional feature extraction
  – Different data properties such as location, scale and shape
  – Multivariate data

• User interface
  – Provide full control to the user
  – Intuitive
Data Space Extraction

High-Dimensional Extraction

Multivariate data includes:
- **Mixture fraction** (the amount of fuel in the mixture),
- **Chi** (the mixing rate),
- **Y_OH** (mass fraction of the OH radical),
- **Temperature**, and **Vorticity magnitude**.
High-Dimensional Extraction

Extraction Based on Scale
Discussion

• We have applied machine learning to extract and track time-varying flow features

• The system allows more flexible specification of features of interest

• The machine learning model can be implemented in graphics hardware for interactivity

• Feasible and desirable to use PC cluster for the results

Summary

• Visualization helps glean insights in large data
• Visualization should be integrated into the overall scientific discovery or engineering design process
• We should exploit graphics hardware technologies and other advanced computing methods as much as possible for data understanding tasks
• A PC cluster can handle higher spatial resolution data. No limit on the temporal resolution!
• Multidimensional encoding could be feasible for some classes of data to achieve more ambitious data reduction and to support comparative visualization
• Machine intelligence should be exploited more
Acknowledgements

• NSF ITR
• DOE SciDAC
• DOE ASCI VIEWS

Kwan-Liu Ma
Department of Computer Science, UC Davis
http://www.cs.ucdavis.edu/~ma
ma@cs.ucdavis.edu