A Fast Volume Rendering Algorithm for Time-varying Fields using a Time-space Partitioning (TSP) Tree

Han-Wei Shen, Ling-Jen Chiang, Kwan-Liu Ma

Vis 1999
How do we deal with large dataset?

• Subdivision
  – Break big pieces into smaller ones
Spatial Subdivision

• Hierarchy -- Octree
Add Time...

• One Octree per timestep ?

\[ t = 0 \quad t = 1 \quad t = 2 \ldots \]
Add a Dimension...

• 4D tree Octree (8-tree) -> 16 tree ?
Time-Space Partition Tree

- Two Level Hierarchical Subdivision
- $1^{st}$ level Spatial subdivision -- Octree
Time-Space Partition Tree

• Temporal Subdivision

4 time steps
Rendering

- Image composition remains the same as an Octree
Temporal Coherence

• Images in the octree nodes are cached for nodes with high temporal coherence
Time-Varying Volume Rendering

- Approximate reconstruction from the TSP tree

Error = 0

E = 0.05 (3.4% image diff.)
11.2 speedup
Results

- Shock wave: 1024 x 128 x 128, 40 time steps
- Minimum brick size 32 x 32 x 32
- Temporal error tolerance = 0.02

<table>
<thead>
<tr>
<th>Time Step</th>
<th>0</th>
<th>10</th>
<th>20</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td># Bricks Loaded</td>
<td>561</td>
<td>73</td>
<td>75</td>
<td>72</td>
</tr>
<tr>
<td>Percentage</td>
<td>100 %</td>
<td>13.0 %</td>
<td>13.3</td>
<td>12.8 %</td>
</tr>
</tbody>
</table>
Summary

• TSP Tree -- Extend Octree to include temporal information
• Render with standard Octree image composition
• Temporally coherent images are cached to reduce loads
• Allow approximated volume rendering animation via the hierarchy
Importance-Driven Time-Varying Data Visualization

Chaoli Wang, Hongfeng Yu, and Kwan-Liu Ma

Vis 2008
Importance Driven Volume Rendering

- Given a segmentation
- Emphasis important segments (works for medical data)

- How about time varying data?
Time Varying Scientific Data

• No temporal segmentation
• Measure importance
• Focus on analysis

• How to capture the important aspect of data?
  – Importance – amount of change, or “unusualness”
• How to utilize the importance measure?
  – Data classification
  – Abnormality detection
  – Time budget allocation
  – Time step selection
Importance

- Consider data as feature vectors \([X, Y]\)

- Blockwise importance measurement

- Entropy based
  - Mutual Information
    \[
    I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}
    \]
  - Conditional Entropy
    \[
    H(X | Y) = H(X) - I(X;Y)
    \]
• Consider a time window for neighboring blocks

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>4</th>
<th>0</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
</table>

• Importance of a data block $X_j$ at time step $t$:

$$A_{X_{j,t}} = \sum_{i=1..M} w_i \cdot H(X_{j,t} | Y_{j,i})$$

• Importance of time step $t$:

$$A_t = \sum_{j=1..N} A_{X_{j,t}}$$
Examples

Earthquake – Regular

Vortex – Turbulence

Climate -- Periodic

$T$
Cluster the Curves (k Means)

- 599 time steps
- 50 segments

- 1200 time steps
- 120 segments

- 90 time steps
- 90 segments
Results Highlights

- Earthquake
Time Budgeting

- Allocate rendering time base on importance

$$\omega_t = \Omega * \frac{A_t^\gamma}{\sum_{i=1}^{T} A_i^\gamma}$$
Time Step Selection

- Select the first time step
- Partition the rest of time steps into \((K-1)\) segments
- In each time segment, select one time step:
- Maximize the joint entropy

\[
t = \arg \max_{\tau} H(\tau | t')
\]
Summary

• Importance-driven data analysis and visualization
  – Quantify data importance using Entropy
  – Cluster the importance curves
  – Leverage the importance in visualization