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?

Add a Dimension...

• 4D tree Octree (8-tree) → 16 tree?

Time-Space Partition Tree

• Two Level Hierarchical Subdivision
• 1st level Spatial subdivision → Octree
Time-Space Partition Tree

- Temporal Subdivision

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

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

Important 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

  \[ H(X) = -\sum_{x \in X} p(x) \log p(x) \]

  – Mutual Information

  \[ I(X; Y) = \sum_{x \in X, y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \]

  – Conditional Entropy

  \[ H(X | Y) = H(X) - I(X; Y) \]

Examples

• Consider a time window for neighboring blocks

• Importance of a data block \(X_j\) at time step \(t\):

  \[ A_{ij} = \sum_{x \in X_j} H(X_j | Y, t) \]

• Importance of time step \(t\):

  \[ A_t = \sum_{j} A_{ij} \]
Cluster the Curves (k Means)

Results Highlights

- Earthquake

Time Budgeting

- Allocate rendering time base on importance
  \[ \omega_i = \Omega \cdot \frac{A_i^T}{\sum_{i=1}^{K} A_i^T} \]

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 \( t = \arg \max_i H(e_i | t) \)

Summary

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