High-Fidelity Rendering of Animations on the Grid: A Case Study

Vibhor Aggarwal, Alan Chalmers and Kurt Debattista

Visualisation Group
Outline

• Introduction
• Motivation
• Key Concepts
• Single-pass and Two-pass Approach
• Results
• Conclusion and Future Work
Introduction

• High-fidelity rendering
  – Physically-based quantities/materials
  – Global-illumination
  – Walk-through Animations
Motivation

• Speedup Rendering
  – Shared computational resources
• Current Parallel Solutions
  – Few dedicated resources
  – Rely heavily on communication
• Distributed Massively Parallel Systems
  – Grid Computing
Grid Computing

- Distributively-owned multi-programmed computing resource
- Set of interconnected clusters, databases and equipment
  - Spanning administrative and geographic boundaries
- Massively parallel
  - Reduced control
  - Restricted communication
Grid Computing – Job Execution

A user submits job to the Grid

Grid middleware finds suitable idle resources to execute the job

Binaries and any other data required is transferred to the remote node

Job is executed by the remote node

Results sent back to the User/Data Server

Significant time delay
Grid Computing contd…

- **Issues**
  - Minimise (no) control/data communication
    - Deadlock is easy
  - Dynamic change of resources
    - Multi-programmed/multi-user environments
  - Load Balancing
    - Heterogeneous Resources
  - Data Management
Single-pass Approach

- Queue up each frame of animation as a job [Chong et al 06]
  - Straightforward Approach
Example: Single-pass Approach
Single-pass Approach

• Issues
  – Slow
    • Does not take advantage of Temporal Coherence
  – Artefacts
    • Incoherent Irradiance Cache
Irradiance Cache

• Acceleration Data Structure [Ward et al 88]
  – Distributed ray tracing
  – Accelerates rendering by an order of magnitude

• Algorithm
  – Caches Indirect diffuse samples
  – Interpolates/extrapolated from previous samples within radius
Artefacts: Explanation
Two-pass Approach

• First pass
  – Select important frames
  – Many caches
  – Merge cache

• Second pass
  – Distribute merged cache
  – Render animation frames
First Pass & Merging

Irradiance Caches

Merged Cache
Second Pass

Merged Cache
First Pass

- Selection of frames
  - Giving equal weight to change of direction and position
    - Number of available resources at start
- Rendering pixels sampled using Sobol sequence
  - Irradiance cache hit miss ratio 10:1
Merging the Cache

• The Irradiance Cache
  – Merged
    • Fast process taking few seconds
  – Shared

• Scalability bottleneck
Second Pass

• Same as standard method
  – Queue up each frame as a job
    • Render using the merged cache

• Rendered frames
  – Stored on data server
  – Sent back to the user’s machine

• Job failure detected by Grid Technologies
Example: Two-pass Approach
Results

• Radiance Light Simulation package
• National Grid Service, United Kingdom
  – 4 Core Sites with about 1000 processors
  – Approximately 200 processors used
    • Restricted by the multi-user environment
• 2 to 3 indirect bounces
Visual Quality

• Brightness Flickering Metric (BFM)
  – $|| L_i - L_{i-1} ||$
    • $L_i$ – Average Brightness of the frame

• Temporal Noise
  – Restricted to parts of the frames predominantly lit by indirect lighting
Visual Quality

The University of Warwick

WDL
Digital Innovation
# Timing and Speed-up

<table>
<thead>
<tr>
<th>Animation Name</th>
<th>Frames</th>
<th>Resolution</th>
<th>Rendered on</th>
<th>Average Computation Time per frame (hr:min)</th>
<th>Makespan (hr:min)</th>
<th>Speed-up comparing Two-pass and Single-Pass Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalabsha Walk-through</td>
<td>91</td>
<td>1024×768</td>
<td>Intel Xeon 3.0 Ghz</td>
<td>00:06</td>
<td>02:03 09:37 4.69</td>
<td></td>
</tr>
<tr>
<td>Artgallery</td>
<td>192</td>
<td>2048×1536</td>
<td>NGS</td>
<td>00:37</td>
<td>04:35 05:49 1.26</td>
<td></td>
</tr>
<tr>
<td>Corridor</td>
<td>361</td>
<td>2048×2048</td>
<td>NGS</td>
<td>03:43</td>
<td>05:21 11:41 2.19</td>
<td></td>
</tr>
<tr>
<td>Kalabsha Rotation</td>
<td>1441</td>
<td>2048×1536</td>
<td>NGS</td>
<td>00:09</td>
<td>01:48 02:06 1.16</td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

• Use of Grid computing for rendering high-fidelity animations in reasonable time

• Two-pass approach
  – Enhanced visual quality
  – Speed-up achieved over single-pass approach
  – No additional data structure used
    • Can be extended to algorithms based on irradiance cache

• Communication costs on the Grid
  – Unsuitable for real-time rendering
  – Valuable resource
    • High-fidelity animations taking long time for rendering
    • Jobs where computation time is more than the communication costs
Future Work

• Future Work
  – Removing bottleneck at end of first pass
    • Distributing the merge process
    • Launching second pass for portion of animation for which irradiance cache has been calculated
  – Data management and Load balancing
Thank You

• Contact:
  – Vibhor.Aggarwal@warwick.ac.uk

• Acknowledgements
  – UK-EPSRC Grant: EP/D069874/2
  – University of Bristol for the use of the Corridor and Kalabsha models