LOD for 3D Terrain Models

Fuan Tsai
ftsai@csrsr.ncu.edu.tw
Agenda

- Introduction
- LOD for Terrain Meshes
  - Tile-based view-depend culling
  - Quad-tree based LOD
  - Examples
- Concluding Remarks
Introduction

- Challenges in cyber city visualization
  - Vast amount of data
  - Unnecessarily high resolution (detail) of data in different rendering conditions

- Need better algorithms to present information efficiently.
  - Reduce data (details) for processing and transmission
  - Maintain critical features
OGC CityGML LOD

LoD0  LoD1  LoD2

LoD3  LoD4
LOD for Visualization
LOD for Visualization

- Only visible regions need to be processed.
- Process the lowest level of detail first and increase the level of detail as needed.
- Only changed (diff.) data need to be transmitted.
- Users should get the roughest scene as soon as possible.
- The scene refines and changes smoothly.
LOD for Terrain Rendering
Tile-Based View-Dependent Culling
Quad-Tree LOD Processing

DTM → Divided DTM as regular blocks → Construct LOD data for every block → Record the different vertices between adjacent LOD layers → diff vectors
Quad-Tree Thresholding Scheme

\[ \theta = \frac{\text{FOV}}{\text{pixels per scanline}} \]

\[ GSD = \frac{D \times \tan(\theta)}{\cos(r)} \]
Quad-Tree Subdivision

\[
\frac{\text{Number (Distance > Threshold)}}{\text{Number (Vertices in triangle)}} < 5 \quad \rightarrow \text{Stop Refinement}
\]

Side View

Th

Th
Image Pyramid (LOD) for Terrain Texture

- Base Image
- R: Resolution
- Unit: m

- R=640 → 640 < GSD
- R=320 → 320 < GSD < 640
- R=160 → 160 < GSD < 320
- R=80 → 80 < GSD < 160
- R=40 → 40 < GSD < 80
- R=20 → 20 < GSD < 40
- R=10 → 10 < GSD < 20
- R=5

- Image Pyramid (LOD) for Terrain Texture
Two-Layers LOD Sets

Core-LOD-Set
( Finer LOD Set )

Outer-LOD-Set
( Coarser LOD Set )

Finer Level

Coarser Level

Resample

Finer Level

Coarser Level
T-Junction Removal
T-Junction Removal

Main Tile

Adjacent Tile

Main

YM1 = YA1
YM2 = YA2
YM3 < YA3
YM4 = YA3

Delete Original Triangle
Add 2 New Triangles

Adjacent

YM2 = YA2
Test Example

- DEM of Taiwan
  - 40 m horizontal resolution
  - 1 m height resolution
  - Highest altitude: 3941 m
  - Lowest altitude: 0 m
  - $5022 \times 9555$ cells
  - $10 \times 19$ regular tiles
  - Tile size: $513 \times 513$ pixels.
- 2.5 m SPOT-5 Mosaic (880000x160000)
- Tile the image into 10×19 regular images
- 8 levels of detail for each image

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Resolution(m)</th>
<th>Storage(MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>640</td>
<td>0.273</td>
</tr>
<tr>
<td>64</td>
<td>320</td>
<td>0.657</td>
</tr>
<tr>
<td>128</td>
<td>160</td>
<td>2.01</td>
</tr>
<tr>
<td>256</td>
<td>80</td>
<td>7.18</td>
</tr>
<tr>
<td>512</td>
<td>40</td>
<td>27</td>
</tr>
<tr>
<td>1024</td>
<td>20</td>
<td>105</td>
</tr>
<tr>
<td>2048</td>
<td>10</td>
<td>428</td>
</tr>
<tr>
<td>4096</td>
<td>5</td>
<td>1721</td>
</tr>
</tbody>
</table>
Test Example 2—Puget Sound

- DEM of Puget Sound
  - Located in Washington, USA
  - 9 m horizontal resolution
  - 1 feet altitude resolution
  - Highest altitude: 3665 m
  - Lowest altitude: -290 m
  - 11777 × 14849 pixels
  - 23 × 29 regular tiles
  - Tile size: 513 × 513 pixels.
Test Example2—Puget Sound

- Image of Puget Sound
  - Synthetic Image
  - Dimension: 2000x2500
Comparison with Delaunay

- Delaunay triangulation is generated from MT (Multi-Tesselation) Package
- Test area:
  - Taiwan (Tile 6-m)
- Size: 20 km × 20 km
- Maximum Height: 1633 m
- Minimum Height: 0 m
## Comparison with Delaunay

<table>
<thead>
<tr>
<th>Level</th>
<th>LOD 0</th>
<th>LOD 1</th>
<th>LOD 2</th>
<th>LOD 3</th>
<th>LOD 4</th>
<th>LOD 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Vertices</td>
<td>2257</td>
<td>3087</td>
<td>4182</td>
<td>5808</td>
<td>9236</td>
<td>14900</td>
</tr>
<tr>
<td>Number of Quadtree Triangles</td>
<td>4412</td>
<td>6053</td>
<td>8213</td>
<td>11407</td>
<td>18180</td>
<td>29388</td>
</tr>
<tr>
<td>Number of Delaunay Triangles</td>
<td>4459</td>
<td>6103</td>
<td>8285</td>
<td>11524</td>
<td>18350</td>
<td>29638</td>
</tr>
</tbody>
</table>
Comparison with Delaunay

- Level 1: 3087 vertices
Comparison with Delaunay

- Level 3: 5808 vertices
Comparison with Delaunay

- Level 5: 14900 vertices
Performance Evaluation

- CPU : AMD Athlon(tm) Processor 64 3500+
- GPU : NVIDIA GeForce 7300GT (128 MB Memory)
- RAM : 2 GB
- Programming Platform: wxDev C++
- Rendering API : OpenGL
- Rendering Window Size : 1024 × 768
Case 1

- Routes simulation: Taiwan
  - Mountain: 513 km, 960 frames
  - Plain: 623 km, 1521 frames
Case 2

- Routes simulation: Puget Sound
  - Mountain: 144 km, 1202 frames
  - Plain: 160 km, 1480 frames
## Real-Time Rendering Performance

<table>
<thead>
<tr>
<th>Threshold Scheme</th>
<th>Taiwan (Mountain)</th>
<th>Taiwan (Plain)</th>
<th>Puget Sound (Mountain)</th>
<th>Puget Sound (Plain)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FPS</td>
<td>M-tri/sec</td>
<td>FPS</td>
<td>M-tri/sec</td>
</tr>
<tr>
<td>2 pixels error</td>
<td>24.51</td>
<td>5.00</td>
<td>42.71</td>
<td>5.07</td>
</tr>
<tr>
<td>3 pixels error</td>
<td>48.73</td>
<td>4.54</td>
<td>73.96</td>
<td>3.75</td>
</tr>
<tr>
<td>4 pixels error</td>
<td>61.26</td>
<td>3.22</td>
<td>91.01</td>
<td>2.46</td>
</tr>
</tbody>
</table>
Conclusions

- Level of Detail techniques can reduce the amount of data to process for real-time rendering of terrain meshes and building models.
- For terrain rendering, the combination of tile-based view-dependent culling and quad-tree based LOD is effective.
- T-junction removal improves visual quality and only with negligible performance penalty.
Things to do from here:

- Real-time LOD generation
- Impact of LOD to attributes