Terrender:
A Web-Based Multi-Resolution Terrain Rendering Framework

Julian A. Croci
julian.croci@bluewin.ch
University of Zurich
Zurich, Switzerland

Alireza Amiraghdam
amiraghdam@ifi.uzh.ch
University of Zurich
Zurich, Switzerland

Renato Pajarola
pajarola@ifi.uzh.ch
University of Zurich
Zurich, Switzerland

The 27th International Conference on 3D Web Technology (Web3D ‘22)
Background
Background
Background
RASTeR-Algorithm
RASTeR-Algorithm
RASTeR-Algorithm
RASTeR-Algorithm
RASTeR-Algorithm
RASTeR-Algorithm
RASTeR-Algorithm
RASTeR-Algorithm
RASTeR-Algorithm
RASTeR-Algorithm
RASTeR-Algorithm

5-Patch

9-Patch
RASTeR-Algorithm
RASTeR-Algorithm
Error Metrics

Distance Metric
Error Metrics

Distance Metric

Culling Metric
Error Metrics

Distance Metric

Culling Metric

Geometry Metric
Error Metrics

- Distance Metric
- Culling Metric
- Geometry Metric
- Metrics Combined
Geometry Error

• Difference between min and max height bound of a bin tree node
Geometry Error

• Difference between min and max height bound of a bin tree node

• Created bottom up to guarantee saturation condition
Geometry Error

• Difference between min and max height bound of a bin tree node
• Created bottom up to guarantee saturation condition
• Min/max bound for a node given by the min/max of the bounds of its children and the children of its neighbor
Geometry Error

- Difference between min and max height bound of a bin tree node
- Created bottom up to guarantee saturation condition
- Min/max bound for a node given by the min/max of the bounds of its children and the children of its neighbor
Geometry Error

- Difference between min and max height bound of a bin tree node
- Created bottom up to guarantee saturation condition
- Min/max bound for a node given by the min/max of the bounds of its children and the children of its neighbor
Geometry Error

- Difference between min and max height bound of a bin tree node
- Created bottom up to guarantee saturation condition
- Min/max bound for a node given by the min/max of the bounds of its children and the children of its neighbor
Geometry Error

- Difference between min and max height bound of a bin tree node
- Created bottom up to guarantee saturation condition
- Min/max bound for a node given by the min/max of the bounds of its children and the children of its neighbor
Geometry Error

- Difference between min and max height bound of a bin tree node
- Created bottom up to guarantee saturation condition
- Min/max bound for a node given by the min/max of the bounds of its children and the children of its neighbor
# Implementation: System Overview

<table>
<thead>
<tr>
<th>Preprocess</th>
<th>Tile Maker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Height Data</td>
<td>Pyramid of Height Tiles</td>
</tr>
<tr>
<td>Raw Color Data</td>
<td>Pyramid of Color Tiles</td>
</tr>
<tr>
<td>Geometry Error</td>
<td></td>
</tr>
</tbody>
</table>

- **Mandatory**
  - Raw Height Data
  - Raw Color Data
- **Optional**
  - Pyramid of Height Tiles
  - Pyramid of Color Tiles
# Implementation: System Overview

<table>
<thead>
<tr>
<th>Preprocess</th>
<th>Backend</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Raw Height Data" /></td>
<td><img src="image" alt="Pyramid of Height Tiles" /></td>
</tr>
<tr>
<td><img src="image" alt="Raw Color Data" /></td>
<td><img src="image" alt="Pyramid of Color Tiles" /></td>
</tr>
<tr>
<td>Tile Maker</td>
<td>Tile Server</td>
</tr>
<tr>
<td>Geometry Error</td>
<td></td>
</tr>
</tbody>
</table>

- **Mandatory**: Raw Height Data, Raw Color Data
- **Optional**: Tile Maker, Pyramid of Height Tiles, Pyramid of Color Tiles, Geometry Error
Implementation: System Overview

<table>
<thead>
<tr>
<th>Preprocess</th>
<th>Backend</th>
<th>Frontend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Height Data</td>
<td>Pyramid of Height Tiles</td>
<td>Data Manager</td>
</tr>
<tr>
<td>Tile Maker</td>
<td>Pyramid of Color Tiles</td>
<td>Mesh updater</td>
</tr>
<tr>
<td>Raw Color Data</td>
<td>Geometry Error</td>
<td>Geometry Error</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Camera</td>
</tr>
</tbody>
</table>

Mandatory: Raw Height Data, Raw Color Data, Tile Maker, Pyramid of Height Tiles, Pyramid of Color Tiles, Data Manager, Mesh Updater, Geometry Error, Camera.

Optional: Backend, Frontend.
Implementation: „Render Loop“
Implementation: „Render Loop“
Implementation: „Render Loop“

- Camera moved? yes → No Data loading? yes → Update the Bintree
Implementation: „Render Loop“

- Camera moved?
  - Yes
- No Data loading?
  - Yes
- Update the Bintree
- Update the Quadtree
Implementation: „Render Loop“

Camera moved? yes

No Data loading? yes

Update the Bintree

Update the Quadtree

Request not yet loaded data from server

Async data loading
Implementation: „Render Loop“

Camera moved? yes
No Data loading? yes
Update the Bintree
Update the Quadtree
Request not yet loaded data from server
Async data loading

Render Terrain
Implementation: „Render Loop“

- Camera moved? yes → No Data loading? yes → Update the Bintree → Update the Quadtree → Request not yet loaded data from server → Async data loading
- Camera moved? yes → No Data loading? no → Render Terrain
- Camera moved? no → Render Terrain
Implementation: „Render Loop“

Camera moved?

No Data loading?

Update the Bintree

Update the Quadtrees

Request not yet loaded data from server

Async data loading

All data finished loading?

Render Terrain
Implementation: „Render Loop“

- Camera moved?
  - yes
  - no
    - yes: Update the Bintree
    - no: All data finished loading?
      - yes: Render Terrain
      - no: No Data loading?

- No Data loading?
  - yes
    - yes: Update the Quadtree
    - no: Request not yet loaded data from server
  - no: Async data loading
Implementation: „Render Loop“

Camera moved? yes -> No Data loading? yes -> Update the Bintree

Camera moved? no
   -> No Data loading? no
      -> All data finished loading? yes
         -> Render Terrain
      
All data finished loading? no
   -> No Data loading? yes
      -> Update the Quadtree

Update the Bintree

Update the Quadtree

Request not yet loaded data from server

Async data loading
Implementation: Async Data Loading

Individually executed for each M-block
Implementation: Async Data Loading

Individually executed for each M-block

Load height texture → Hand data to webworker

Main thread | Parallel to main thread
Implementation: Async Data Loading

Individually executed for each M-block

Load height texture → Hand data to webworker → Parse data in webworker

Main thread Parallel to main thread
Implementation: Async Data Loading

Individually executed for each M-block

Load height texture → Hand data to webworker → Parse data in webworker → Hand parsed data back to main thread

Main thread  Parallel to main thread
Implementation: Async Data Loading

Individually executed for each M-block

Load height texture → Hand data to webworker → Parse data in webworker → Hand parsed data back to main thread → Load data to GPU

Main thread

Parallel to main thread
Implementation: Async Data Loading

Individually executed for each M-block

Load height texture → Hand data to webworker → Parse data in webworker → Hand parsed data back to main thread → Load data to GPU

Main thread

Parallel to main thread
Implementation: Async Data Loading

Individually executed for each M-block

Load height texture

Load color texture

Hand data to webworker

Parse data in webworker

Hand parsed data back to main thread

Load data to GPU

Main thread

Parallel to main thread
Implementation: Async Data Loading

Individually executed for each M-block

Load height texture → Hand data to webworker → Parse data in webworker → Hand parsed data back to main thread → Load data to GPU

Load color texture → Load data to GPU

Main thread

Parallel to main thread
Implementation: Client Data Management
Implementation: Client Data Management
Implementation: Client Data Management

![Diagram of Client Data Management]

- **Render List**: 2 → 3 → 1 → 4 → 17
- **Future List**: 6 → 7 → 2 → 9 → 10 → 11 → 12 → 13 → 1 → 5
Implementation: Client Data Management

![Diagram showing data management flow with nodes labeled 1 to 21, and lists named Render List, Future List, and onGPU List.

- Render List: 2 → 3 → 1 → 4 → 17
- Future List: 6 → 7 → 2 → 9 → 10 → 11 → 12 → 13 → 1 → 5
- onGPU List: 20 → 8 → 14 → 15 → 16]
Implementation: Client Data Management

```
.Render List  2 <-> 3 <-> 1 <-> 4 <-> 17
.Future List  6 <-> 7 <-> 2 <-> 9 <-> 10 <-> 11 <-> 12 <-> 13 <-> 1 <-> 5
.onGPU List   20 <-> 8 <-> 14 <-> 15 <-> 16
.onRAM List   19 <-> 21
```
Implementation: Client Data Management
Implementation: Client Data Management

![Diagram showing data management processes and lists]

- **Render List**: 6 → 7 → 2 → 9 → 10 → 11 → 12 → 13 → 1 → 5
- **Future List**: 18 → 19 → 20 → 21
- **onGPU List**: 20 → 8 → 14 → 15 → 16 → 3 → 4 → 17
- **onRAM List**: 19 → 21

Max. Length 5
Implementation: Client Data Management

Render List

Future List

onGPU List

onRAM List

Max. Length 5
Implementation: Client Data Management

```
Render List: 6 7 2 9 10 11 12 13
Future List
onGPU List: 15 16 3 4 17
onRAM List: 19 21 20 8 14
Max. Length 3
```
Implementation: Client Data Management

- Render List: 6 → 7 → 2 → 9 → 10 → 11 → 12 → 13 → 14 → 15 → 16 → 17 → 18 → 19 → 20 → 21
- Future List
- onGPU List: 15 → 16 → 3 → 4 → 17
- onRAM List: 20 → 8 → 14

Max. Length 3
Implementation: Dynamic Updates
Implementation: Dynamic Updates

Required M-Blocks: 1
Not Ready M-Blocks: 0
Ratio: Not Ready / Required: 0

Threshold: 0.6
Implementation: Dynamic Updates
Implementation: Dynamic Updates

Required M-Blocks: 4
Not Ready M-Blocks: 2
Ratio: Not Ready / Required: 0.5

Threshold: 0.6
Implementation: Dynamic Updates
Implementation: Dynamic Updates

Required M-Blocks: 15
Not Ready M-Blocks: 10
Ratio: Not Ready / Required: 0.67

Threshold: 0.6
Results: Bandwidth no limitations

Device: MacBook Pro M1 Max, 64GB RAM, Chrome Browser
Results: Bandwidth no limitations

Device: MacBook Pro M1 Max, 64GB Ram
Results: Dynamic Update (40Mb/s)

Video 40Mb/s without (left) and with (right) dynamic updates

Device: MacBook Pro M1 Max, 64GB Ram, Chrome Browser
Results: Dynamic Update (40Mb/s)

Device: MacBook Pro M1 Max, 64GB Ram, Chrome Browser
Results: Dynamic Update (40Mb/s)

No Dynamic Updates

Dynamic Updates

Device: MacBook Pro M1 Max, 64GB Ram, Chrome Browser
Results: Dynamic Update (40Mb/s)

Device: MacBook Pro M1 Max, 64GB Ram, Chrome Browser
Comparison with CesiumJS

Terrain updates during the same trajectory
Terrender: A Web-Based Multi-Resolution Terrain Rendering Framework

Thank you for watching