

# 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**

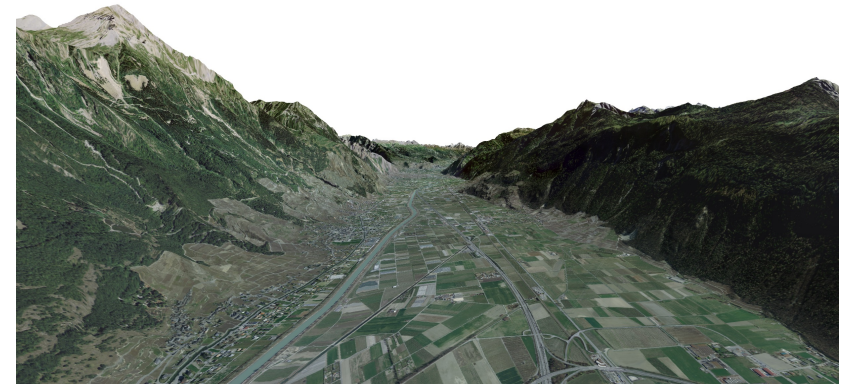
pajarola@ifi.uzh.ch

University of Zurich

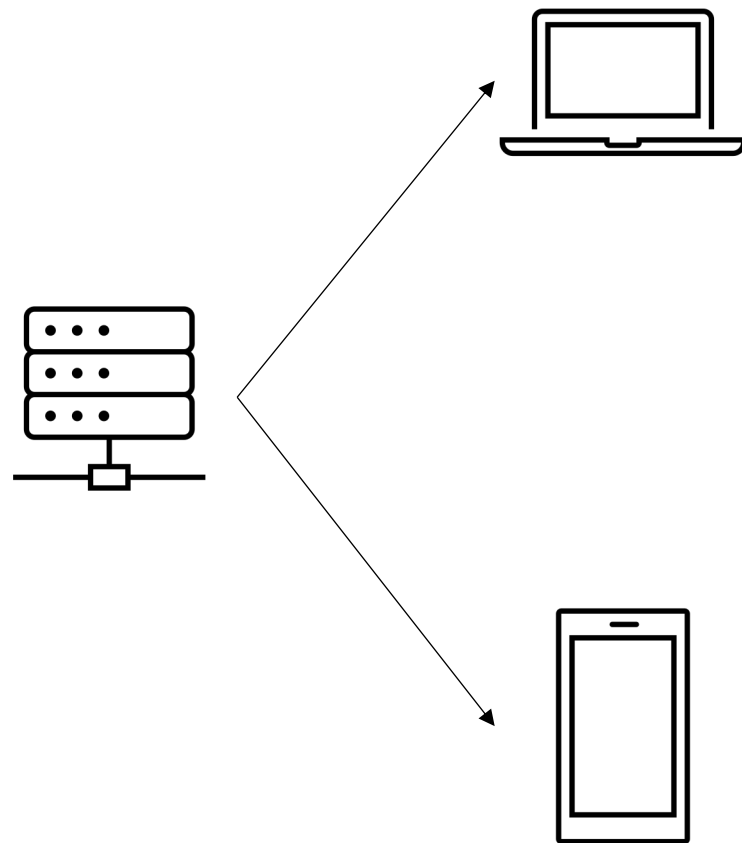
Zurich, Switzerland

The 27th International Conference on 3D Web Technology (Web3D '22)

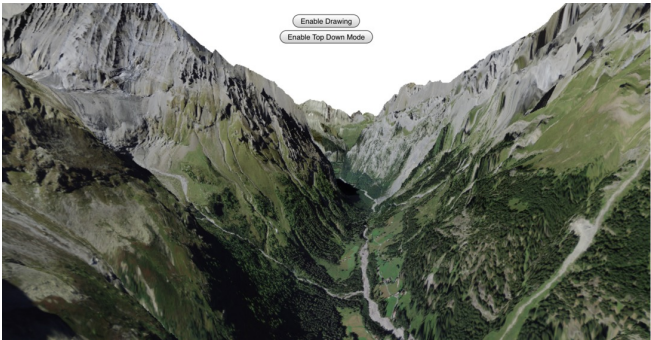
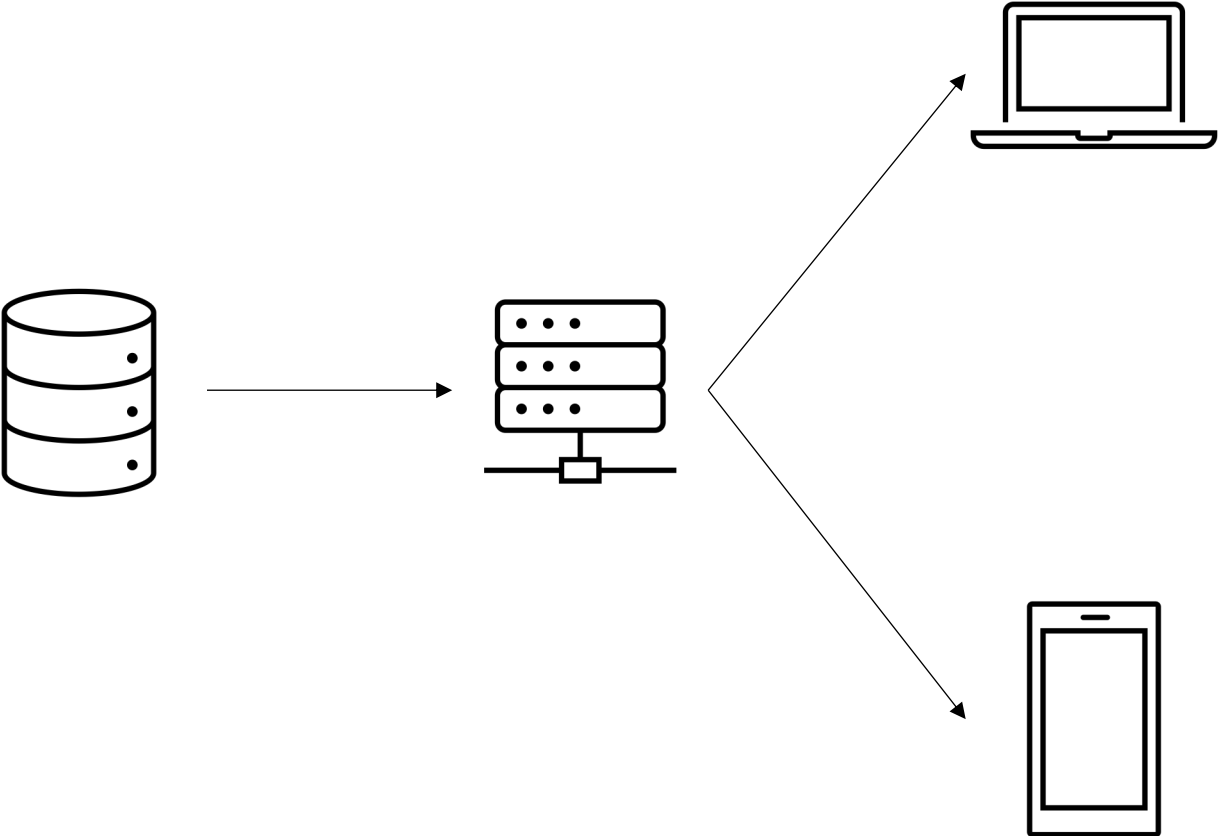
# Background



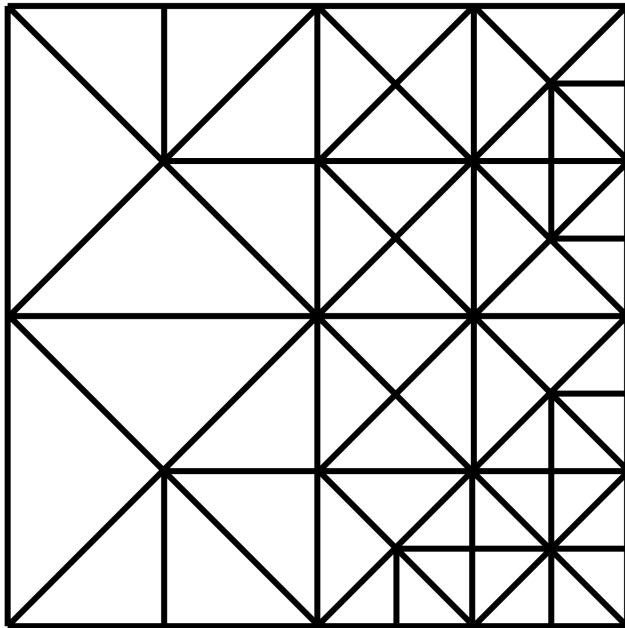
# Background



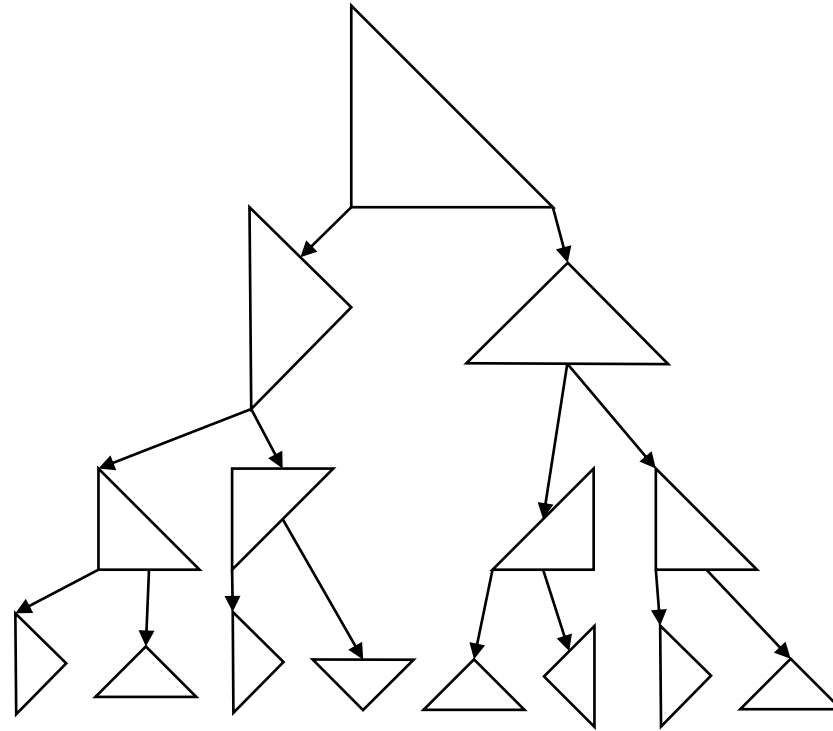
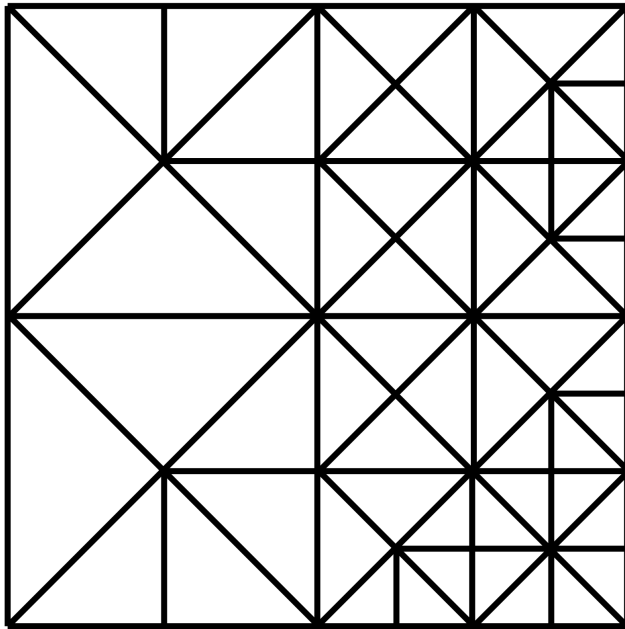
# Background



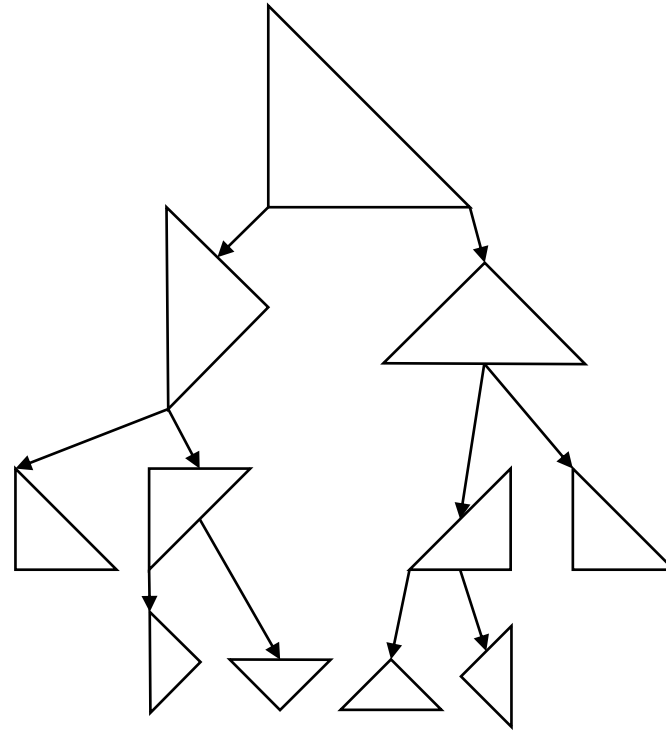
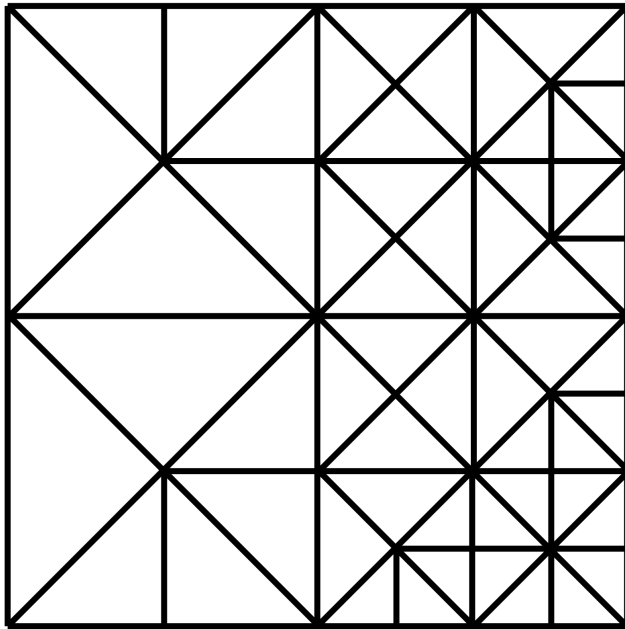
# RASTeR-Algorithm



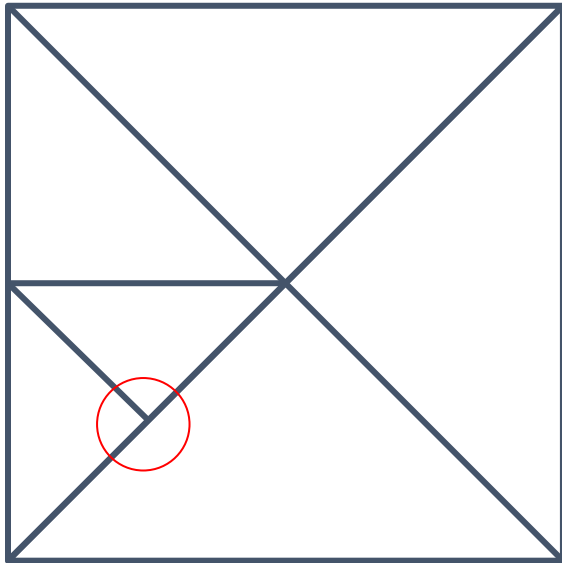
# RASTeR-Algorithm



# RASTeR-Algorithm

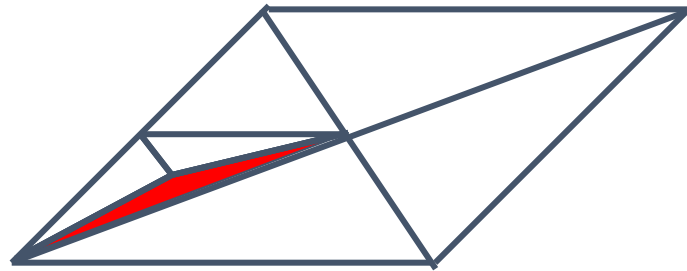
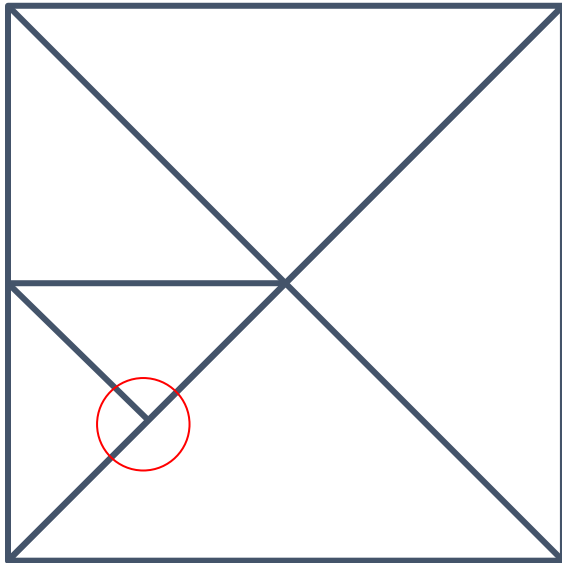


# RASTeR-Algorithm

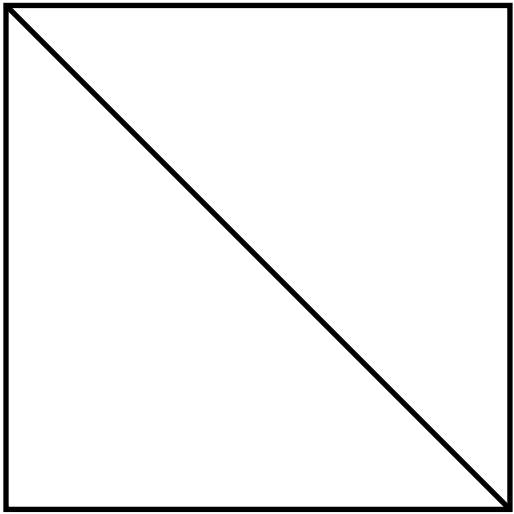




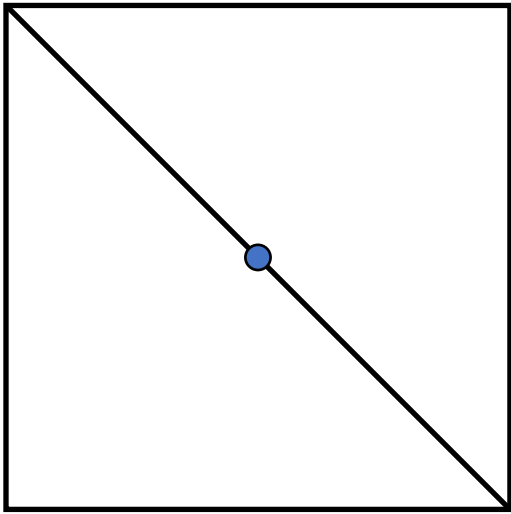
# RASTeR-Algorithm



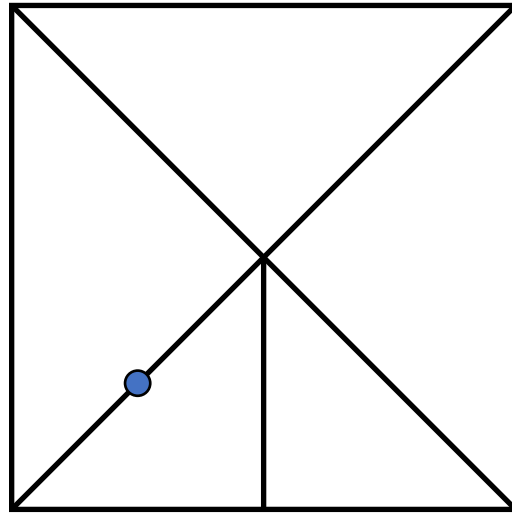
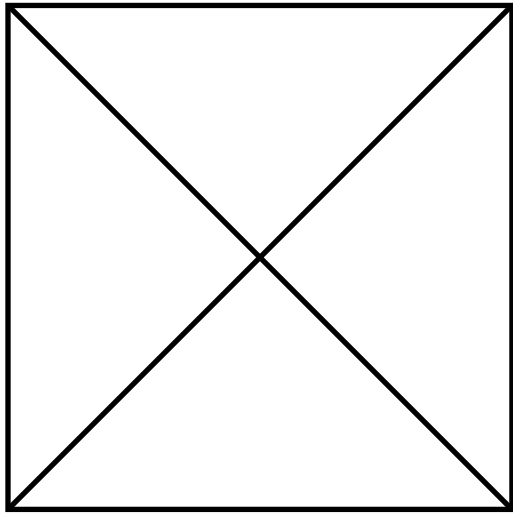
# RASTeR-Algorithm



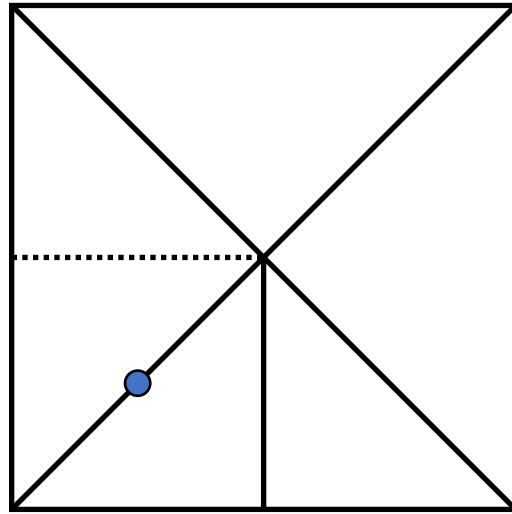
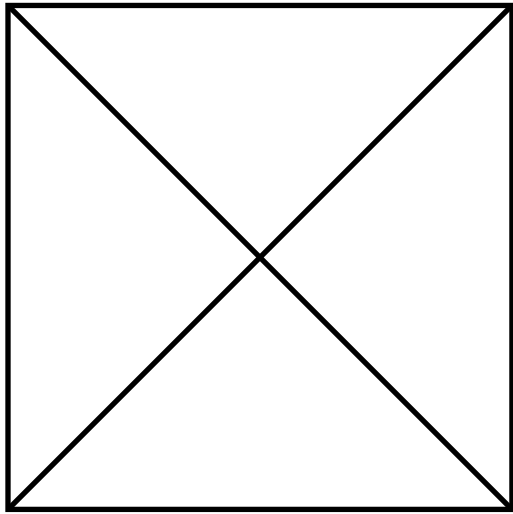
# RASTeR-Algorithm



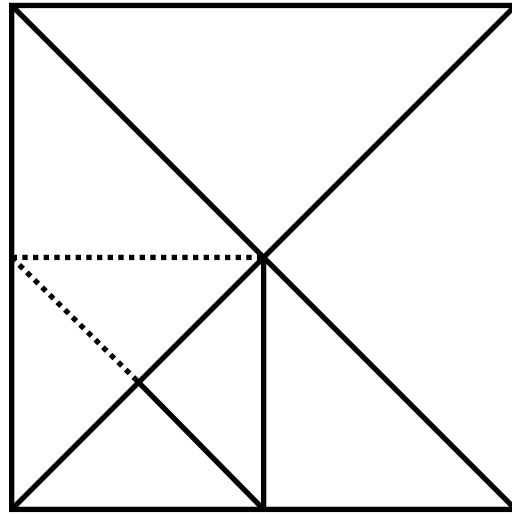
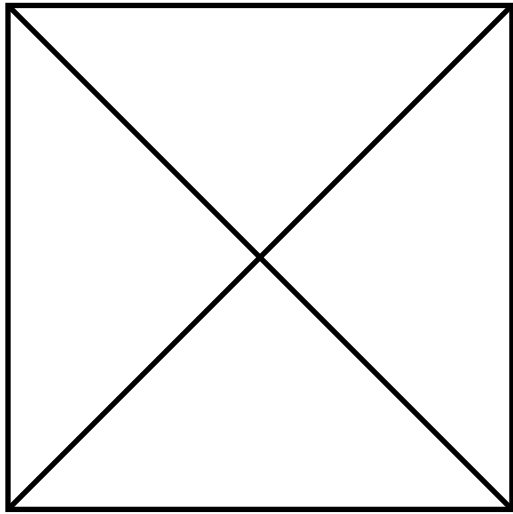
# RASTeR-Algorithm



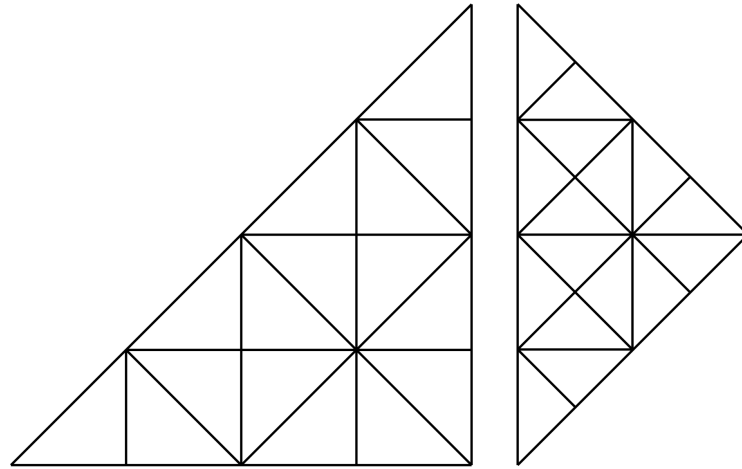
# RASTeR-Algorithm



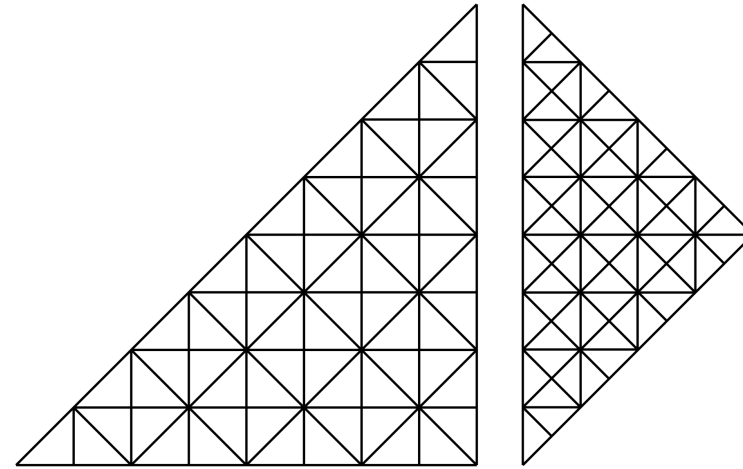
# RASTeR-Algorithm



# RASTeR-Algorithm

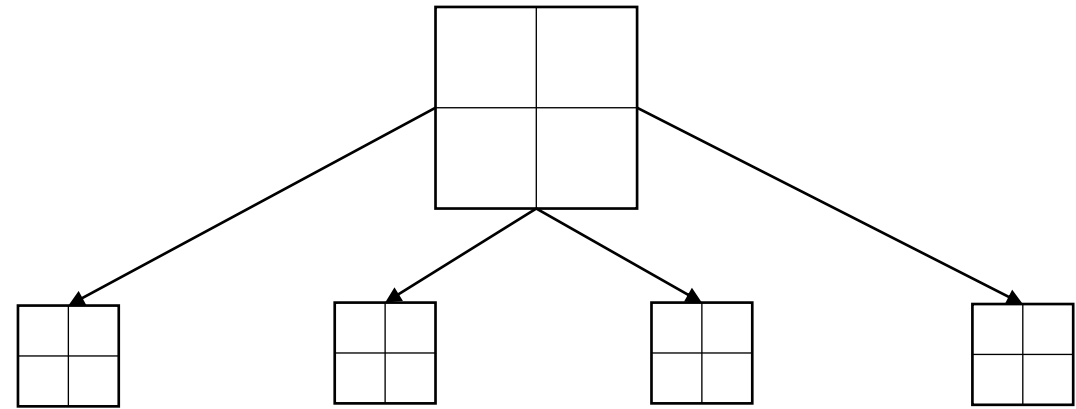
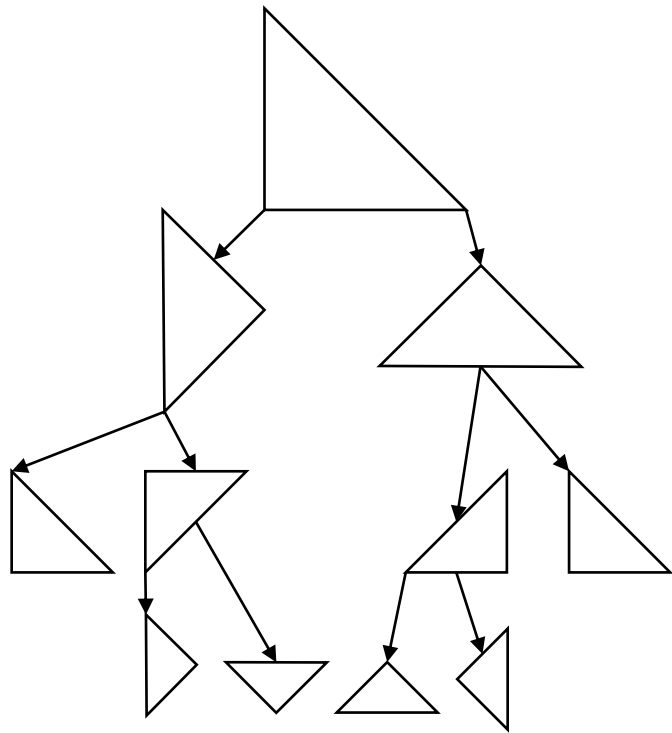


5-Patch



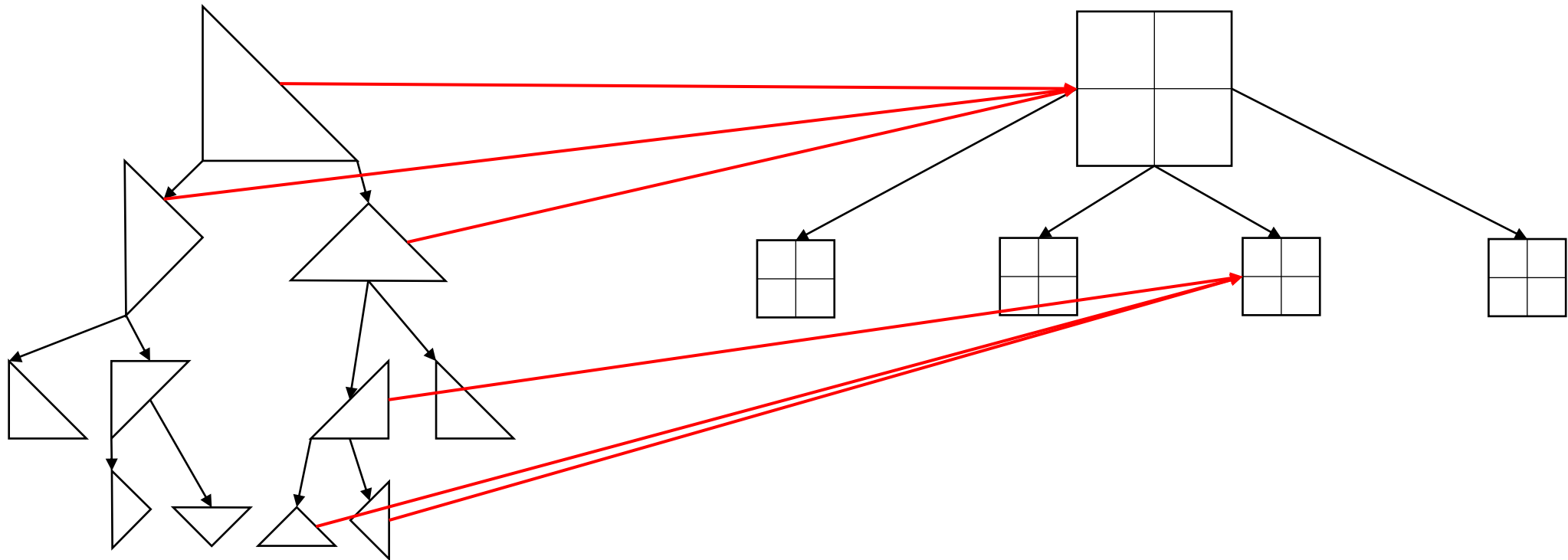
9-Patch

# RASTeR-Algorithm

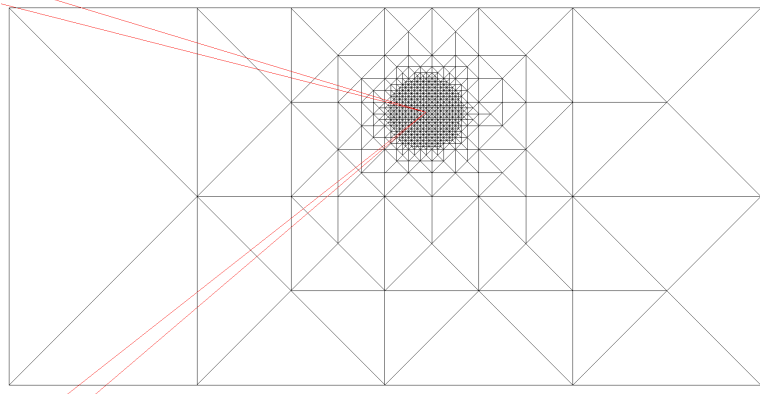




# RASTeR-Algorithm

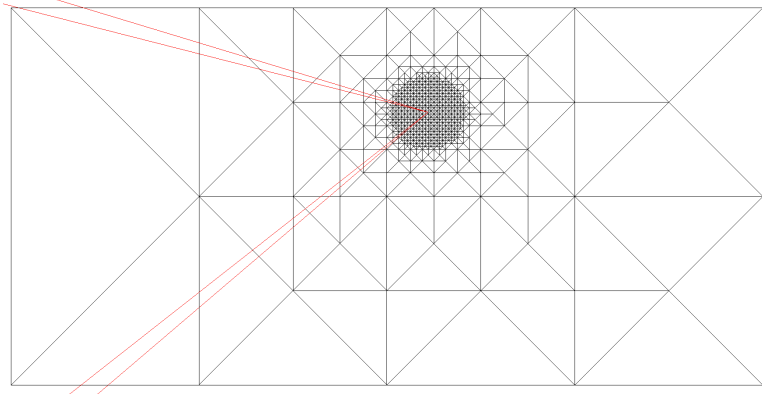


# Error Metrics

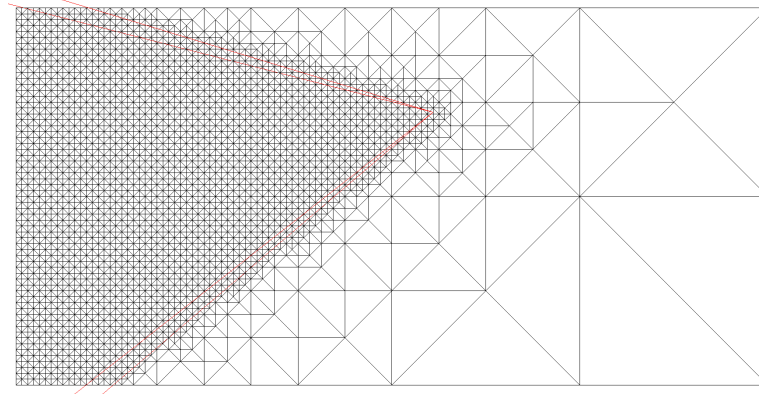


Distance Metric

# Error Metrics

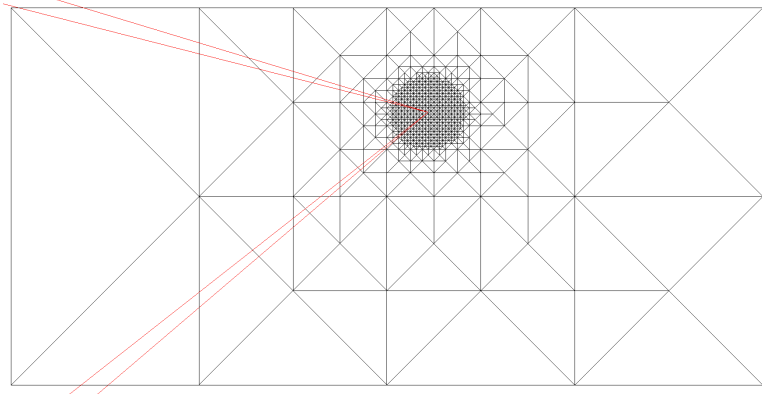


Distance Metric

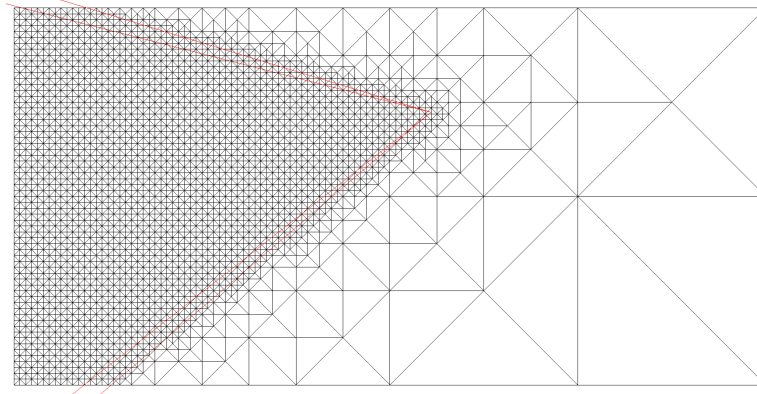


Culling Metric

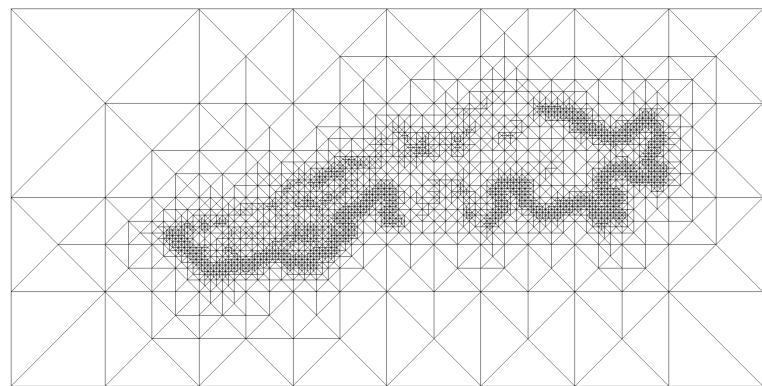
# Error Metrics



Distance Metric

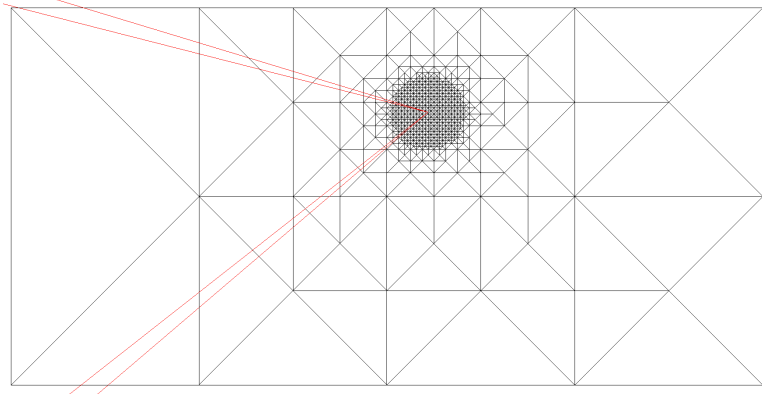


Culling Metric

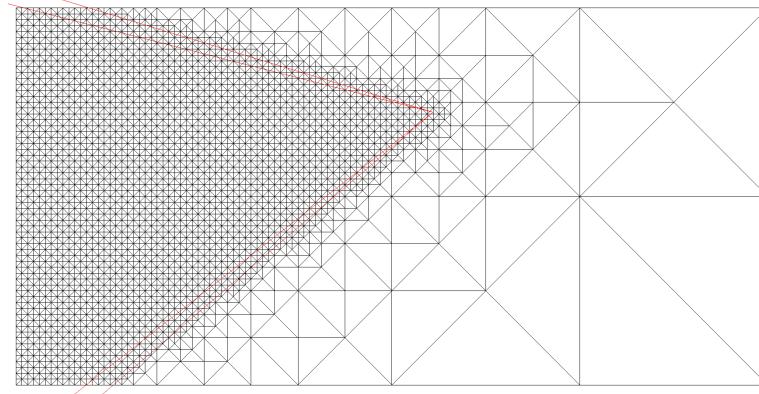


Geometry Metric

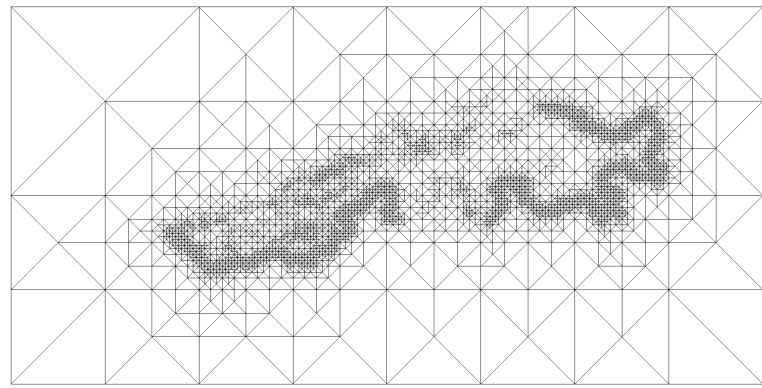
# Error Metrics



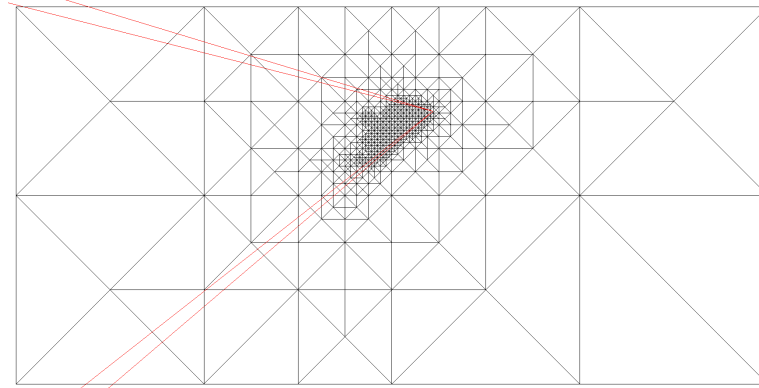
Distance Metric



Culling Metric



Geometry Metric



Metrics Combined

# Geometry Error

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

# Geometry Error

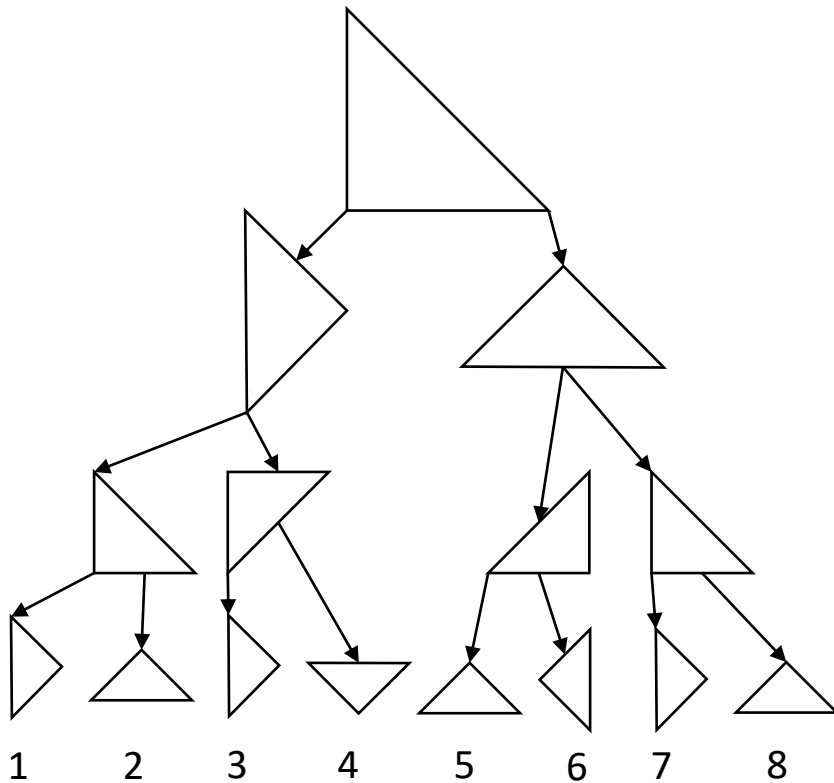
- Difference between min and max height bound of a bin tree node
- Created bottom up to guarantee saturation condition

# 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

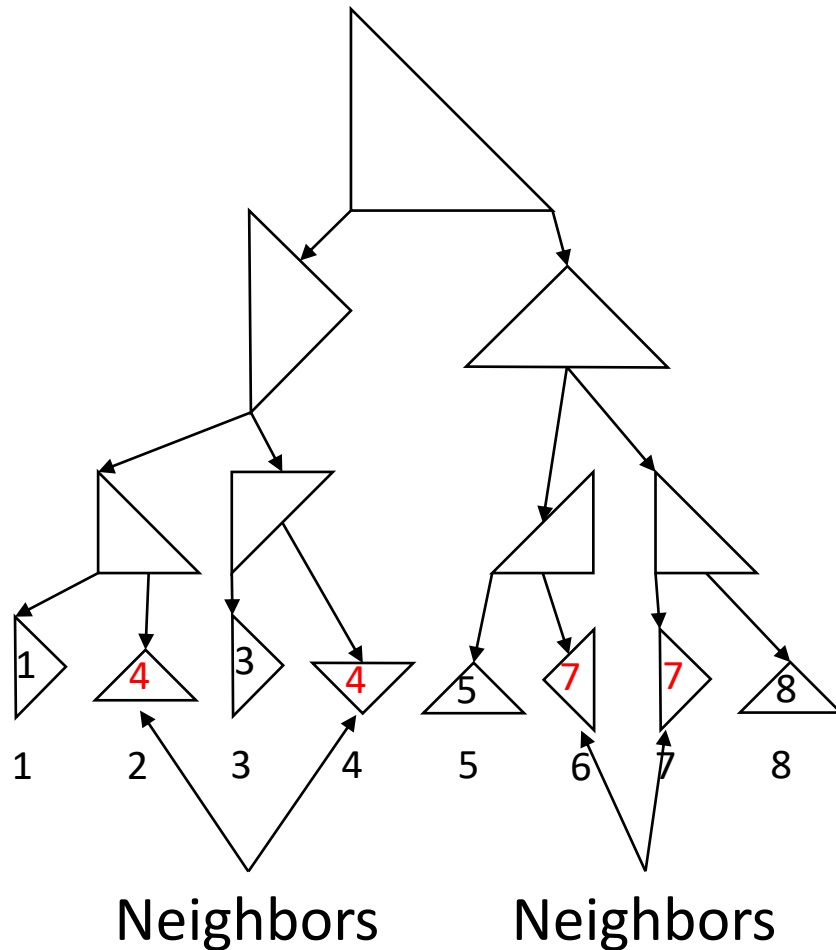


# Geometry Error



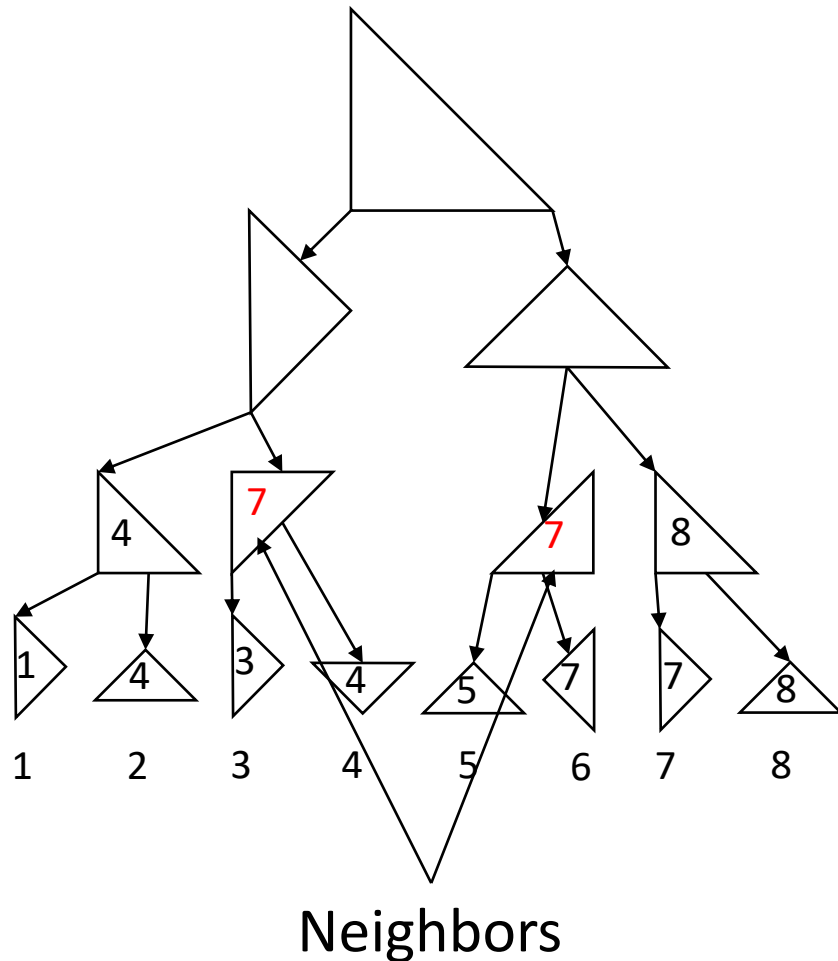
- Difference between min and max height bound of a bin tree node
- Created bottom up to guarantee saturation condition
- Min/max bound for a node given by the min/max of the bounds of its children and the children of its neighbor

# Geometry Error



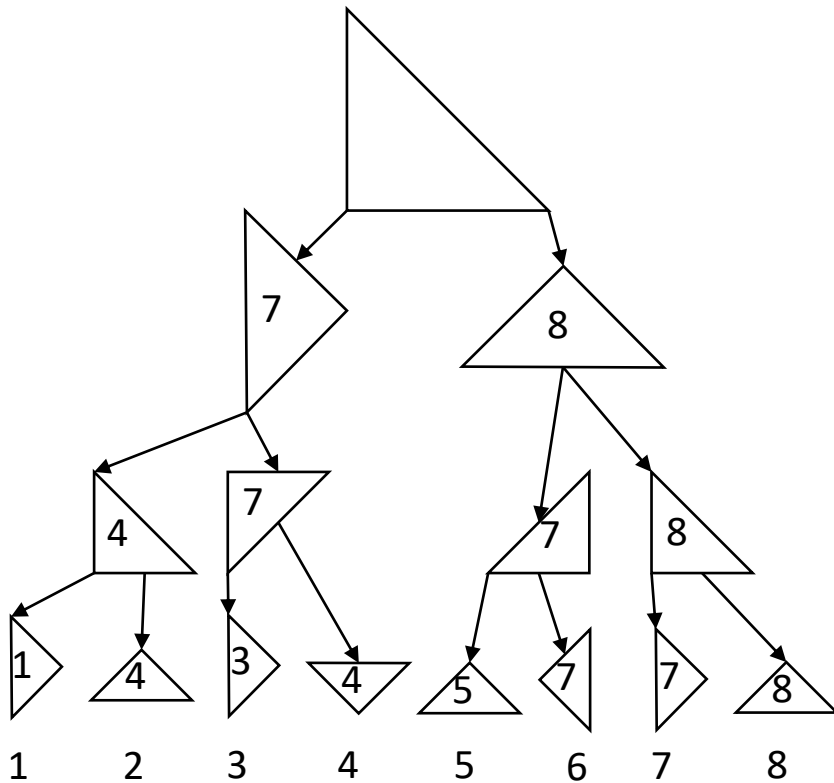
- Difference between min and max height bound of a bin tree node
- Created bottom up to guarantee saturation condition
- Min/max bound for a node given by the min/max of the bounds of its children and the children of its neighbor

# Geometry Error



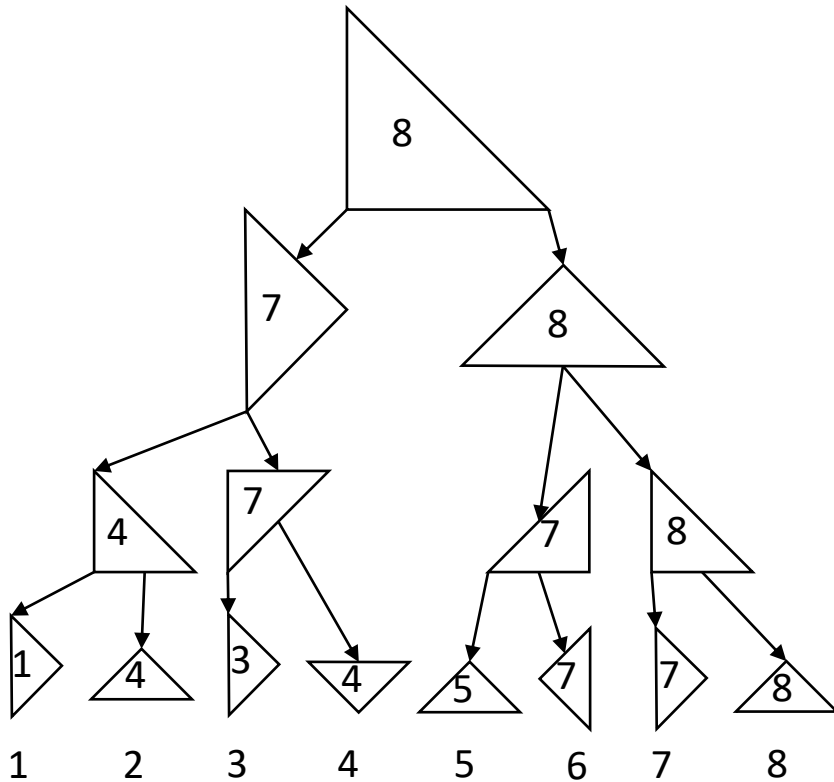
- Difference between min and max height bound of a bin tree node
- Created bottom up to guarantee saturation condition
- Min/max bound for a node given by the min/max of the bounds of its children and the children of its neighbor

# Geometry Error



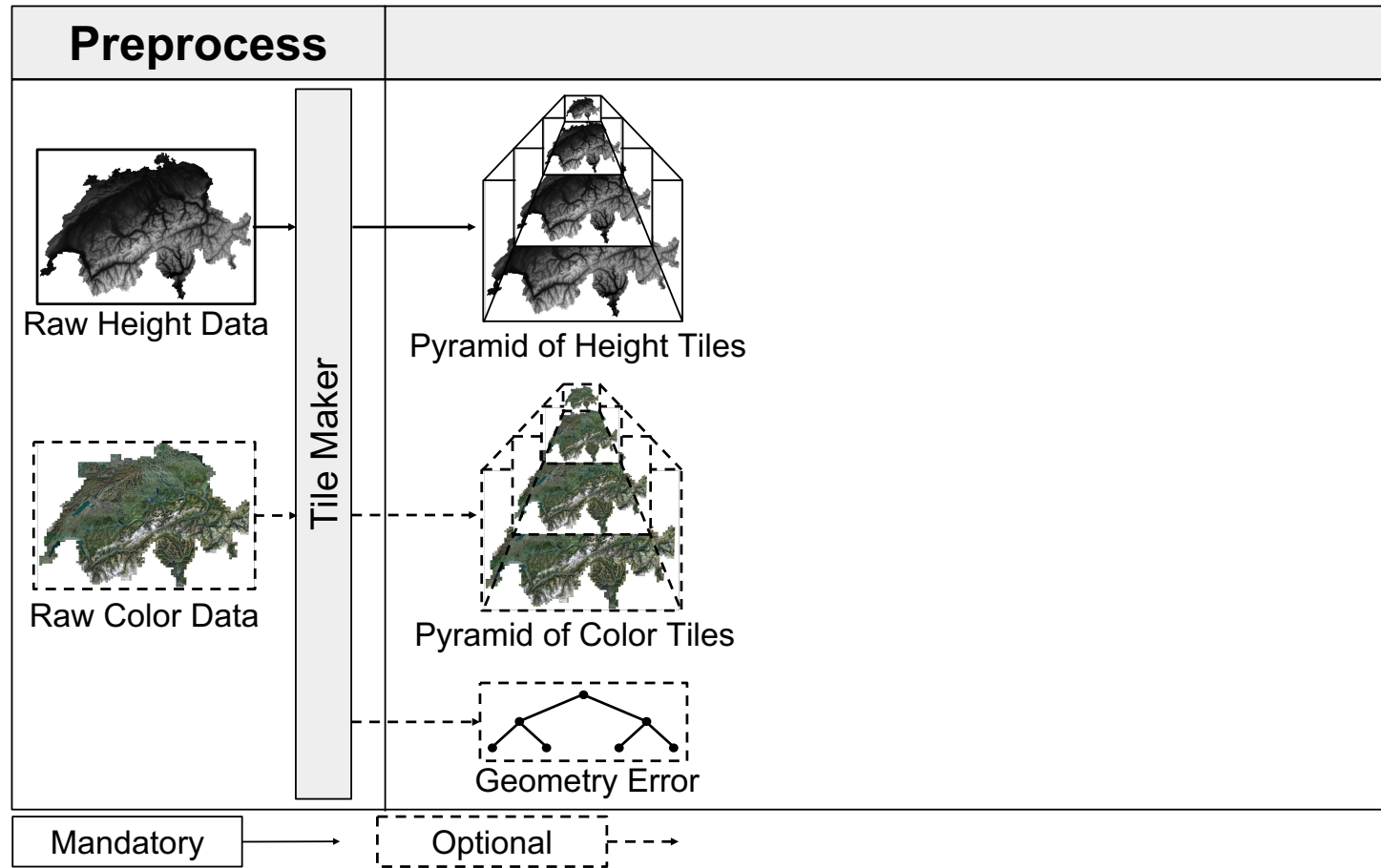
- Difference between min and max height bound of a bin tree node
- Created bottom up to guarantee saturation condition
- Min/max bound for a node given by the min/max of the bounds of its children and the children of its neighbor

# Geometry Error

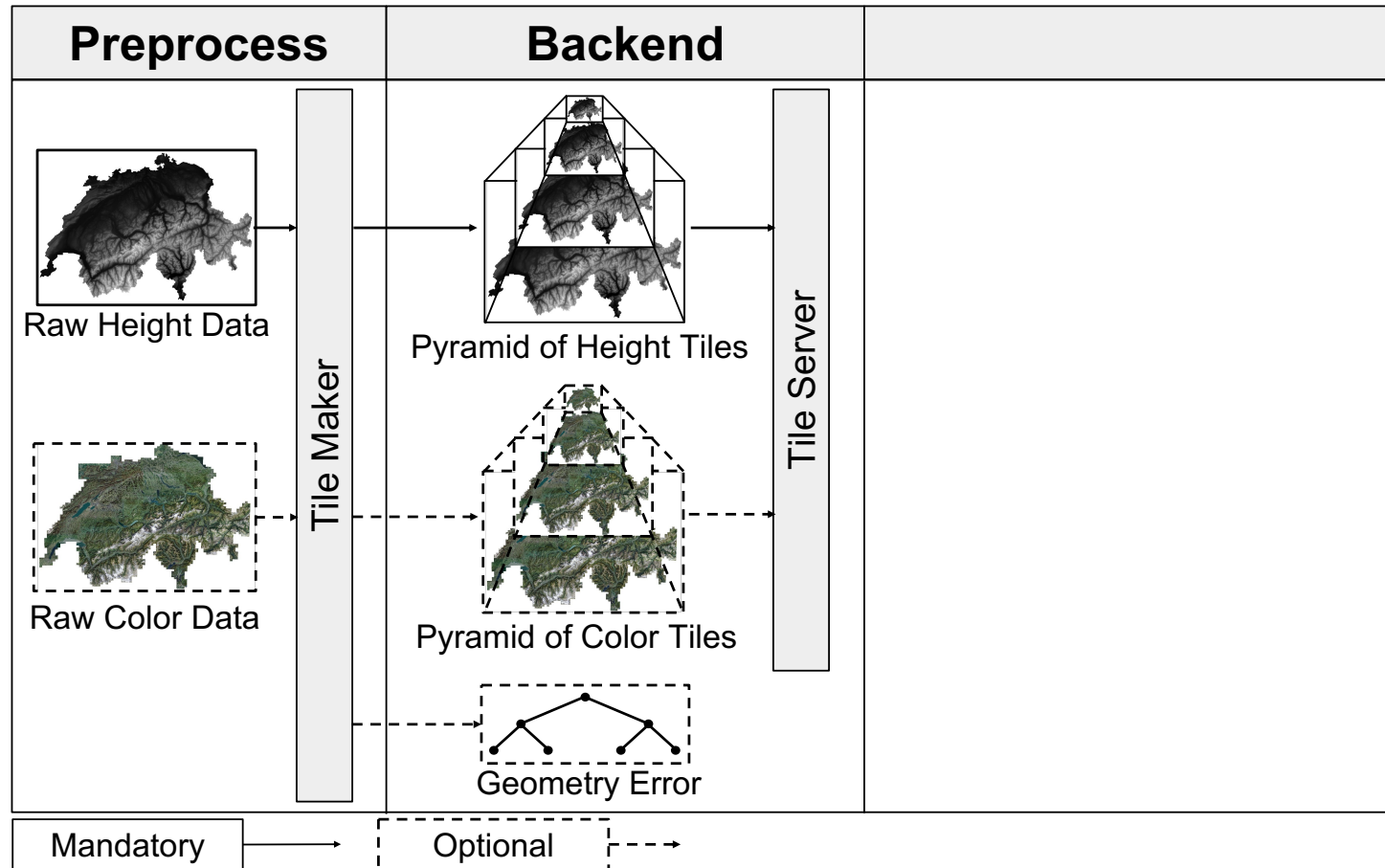


- Difference between min and max height bound of a bin tree node
- Created bottom up to guarantee saturation condition
- Min/max bound for a node given by the min/max of the bounds of its children and the children of its neighbor

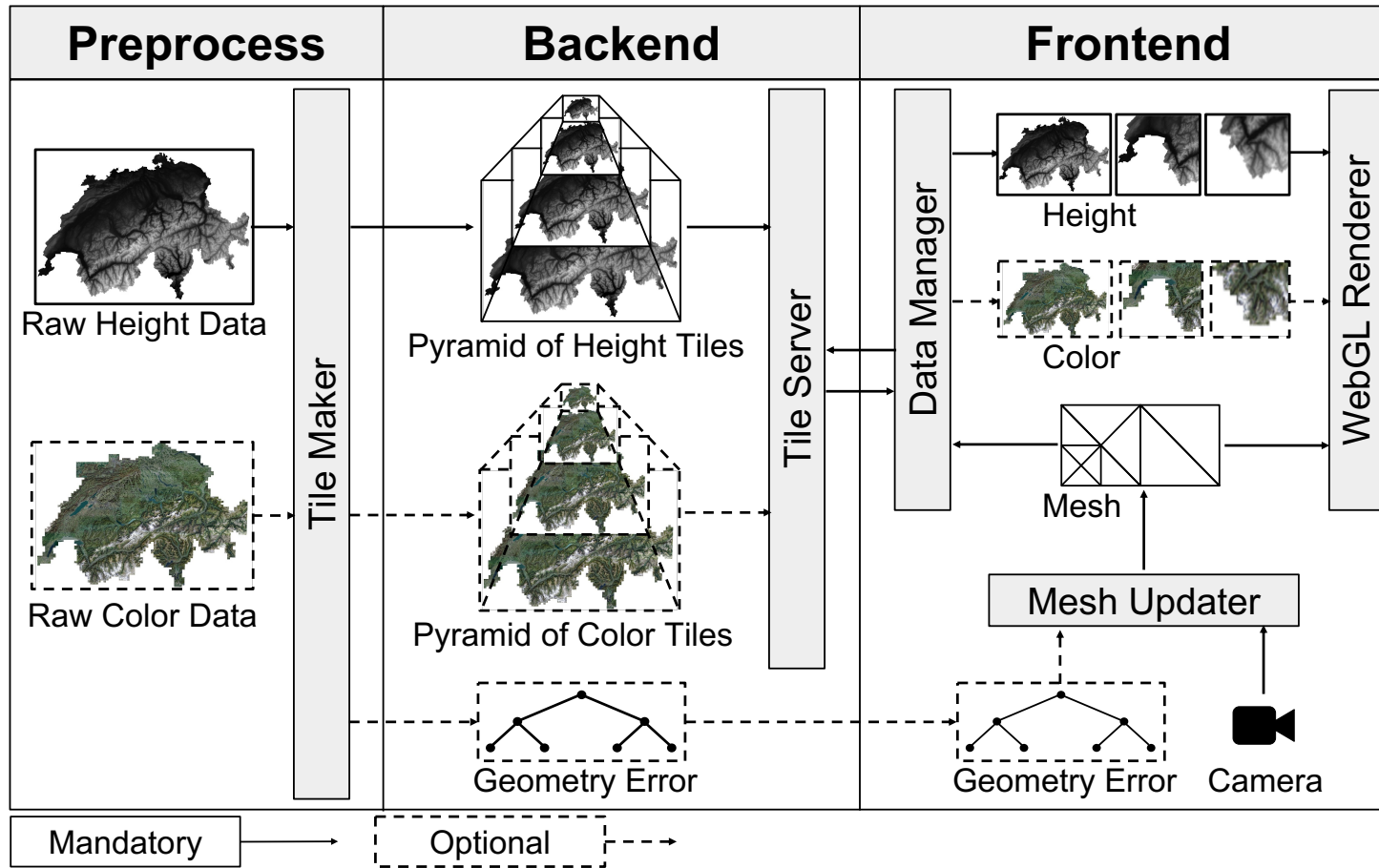
# Implementation: System Overview



# Implementation: System Overview

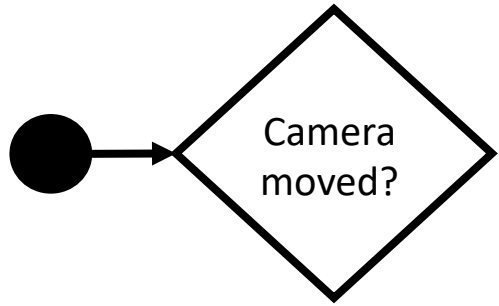


# Implementation: System Overview

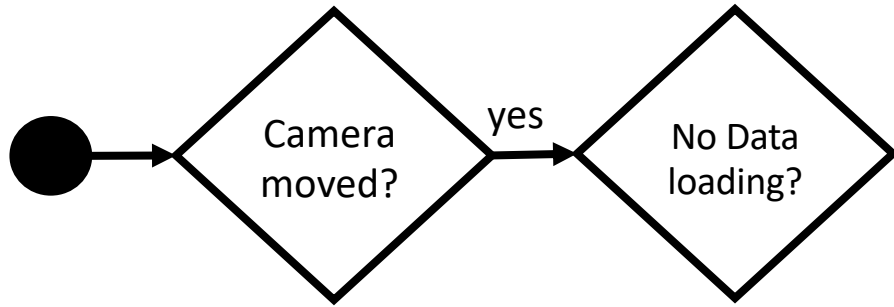




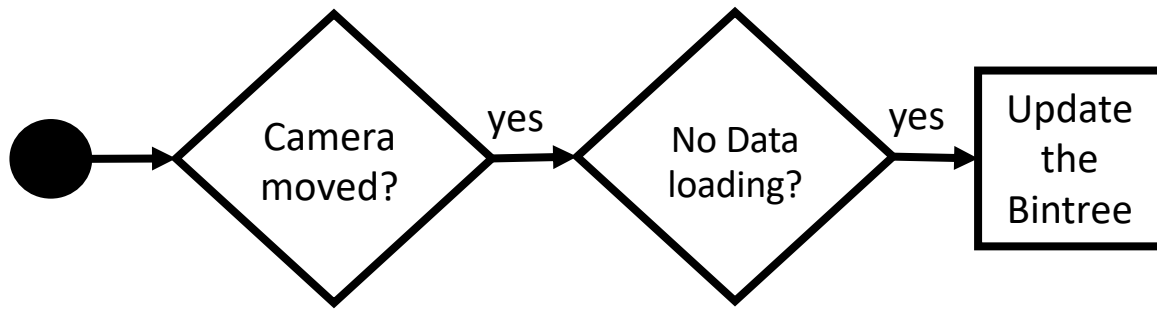
# Implementation: „Render Loop“



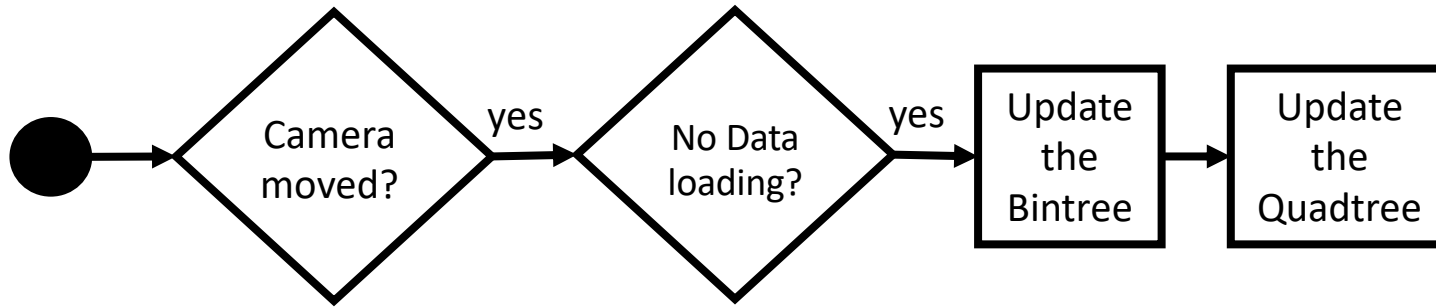
# Implementation: „Render Loop“



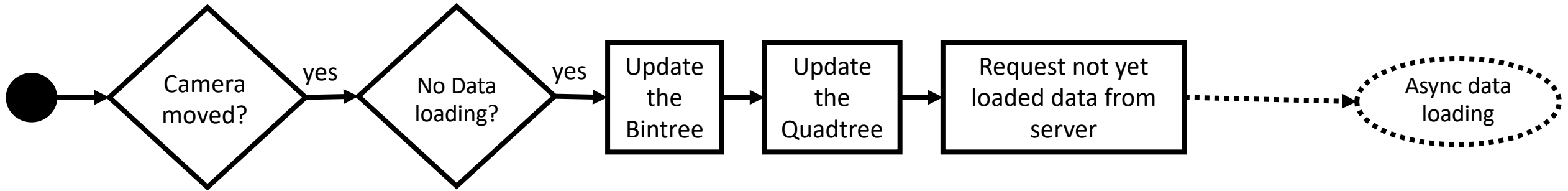
# Implementation: „Render Loop“



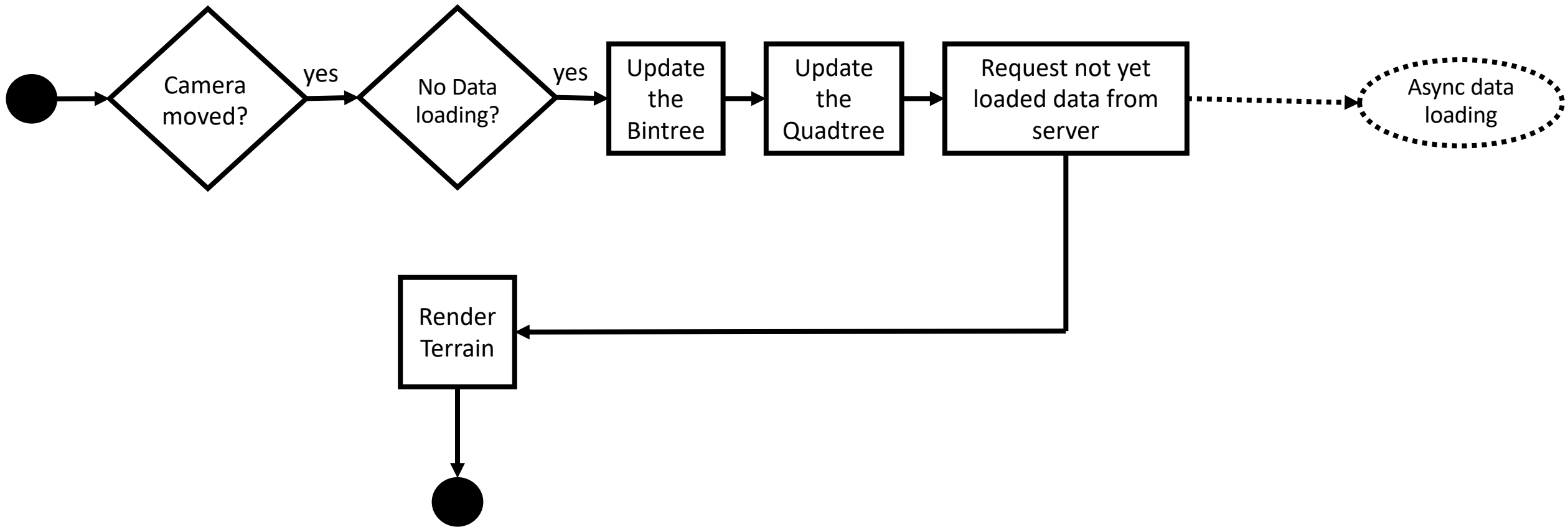
# Implementation: „Render Loop“



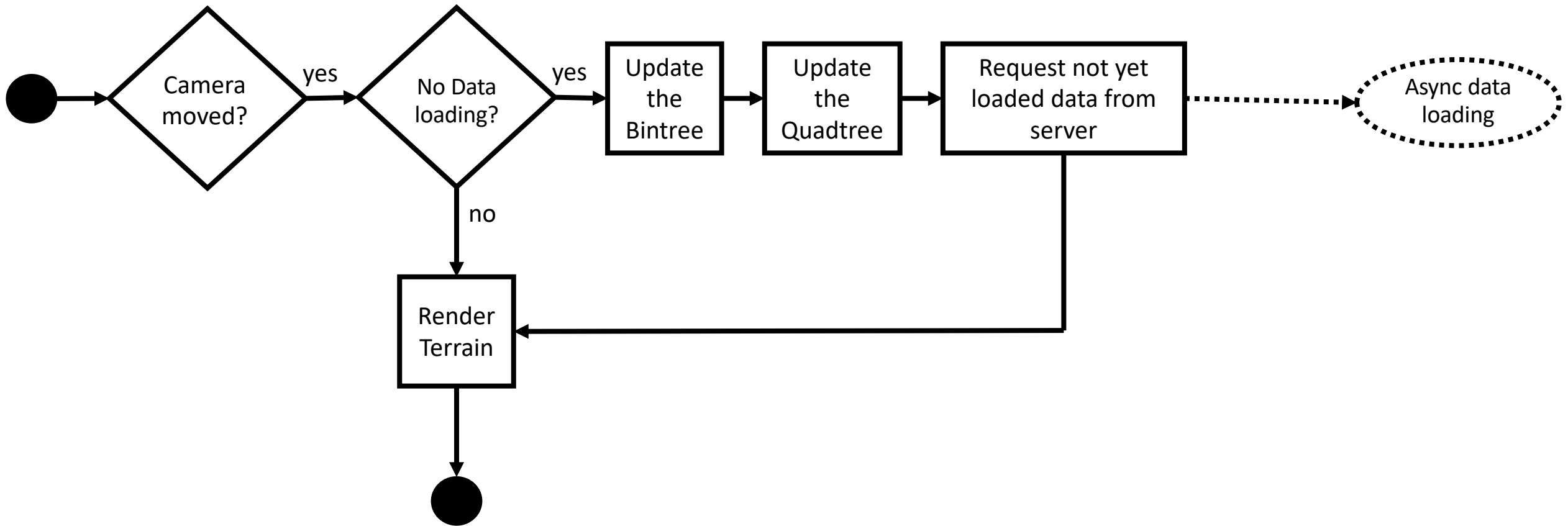
# Implementation: „Render Loop“



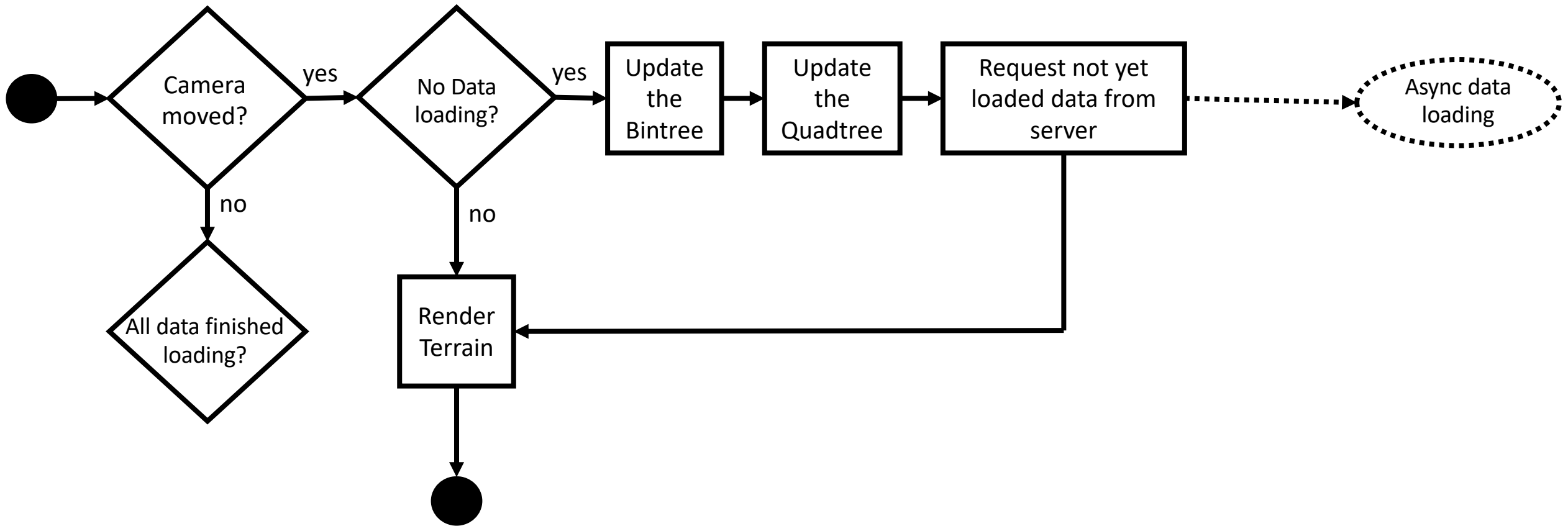
# Implementation: „Render Loop“



# Implementation: „Render Loop“

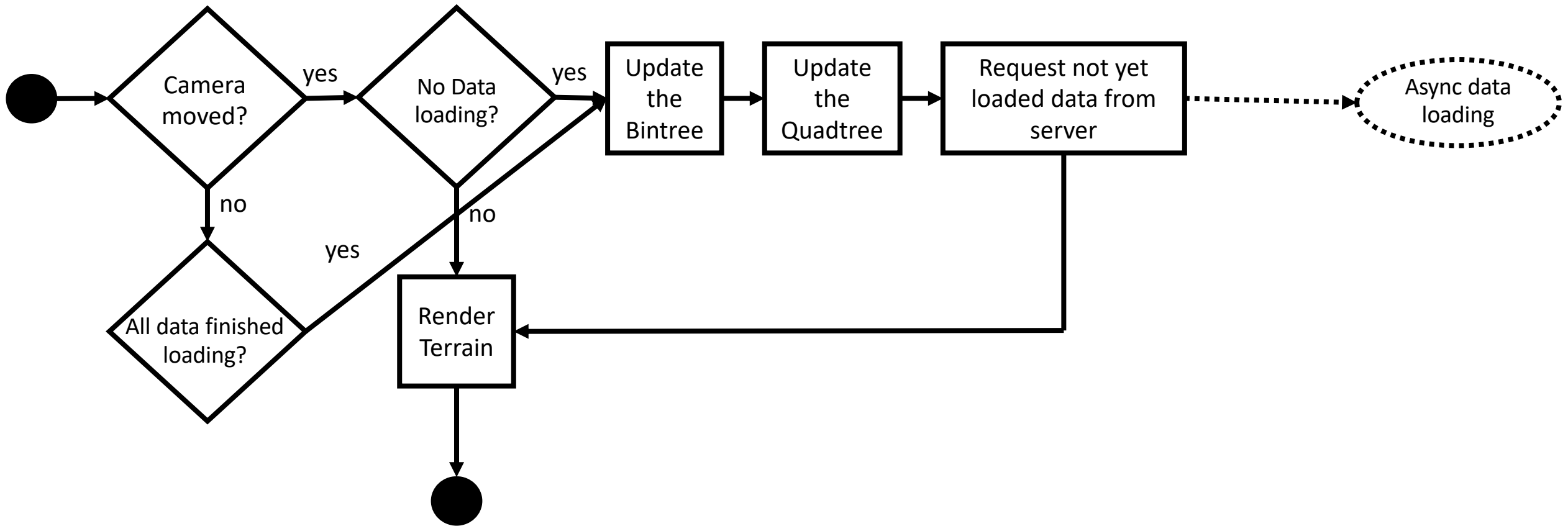


# Implementation: „Render Loop“

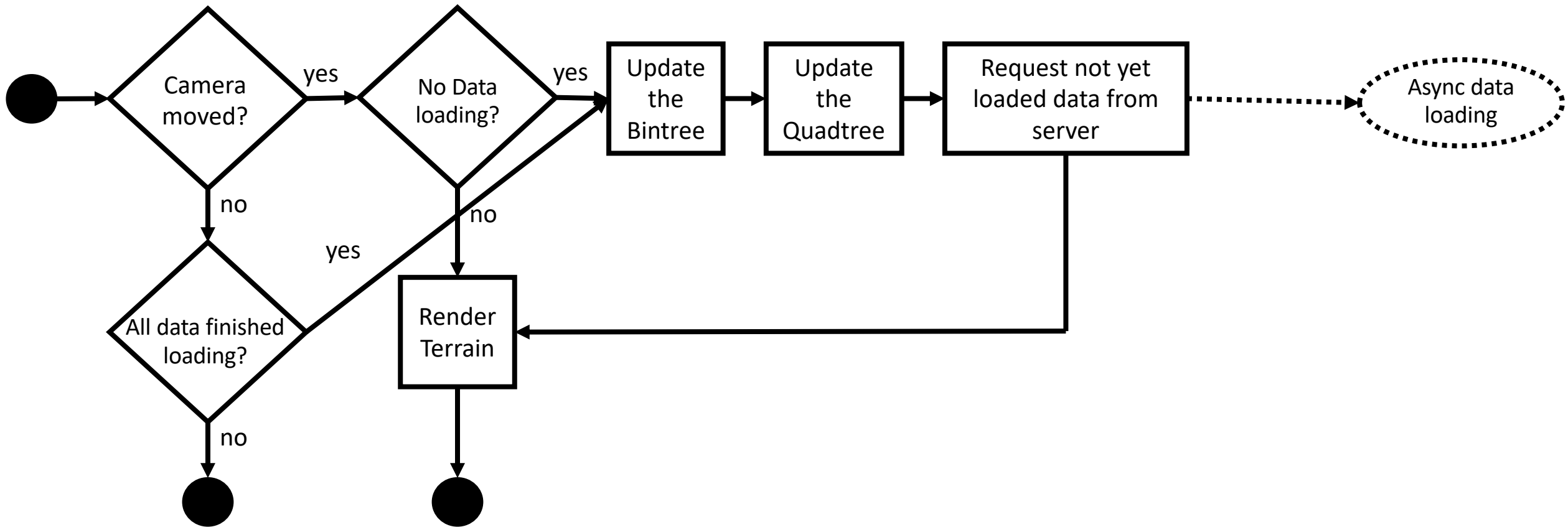




# Implementation: „Render Loop“

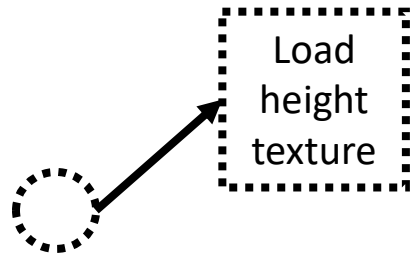


# Implementation: „Render Loop“



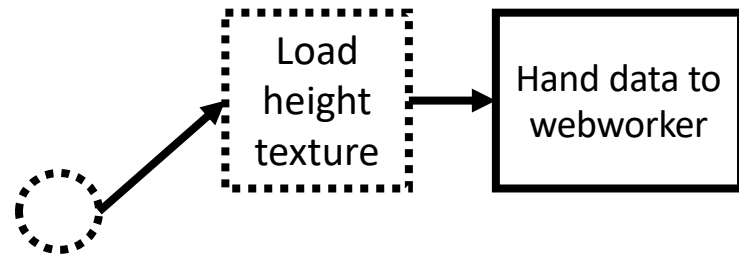
# Implementation: Async Data Loading

Individually executed for each M-block



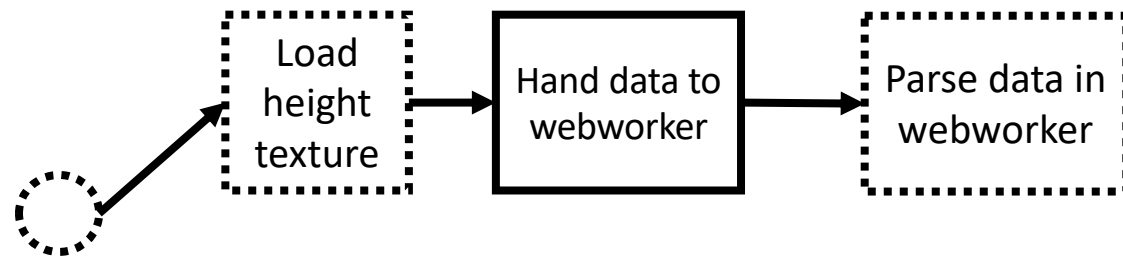
# Implementation: Async Data Loading

Individually executed for each M-block



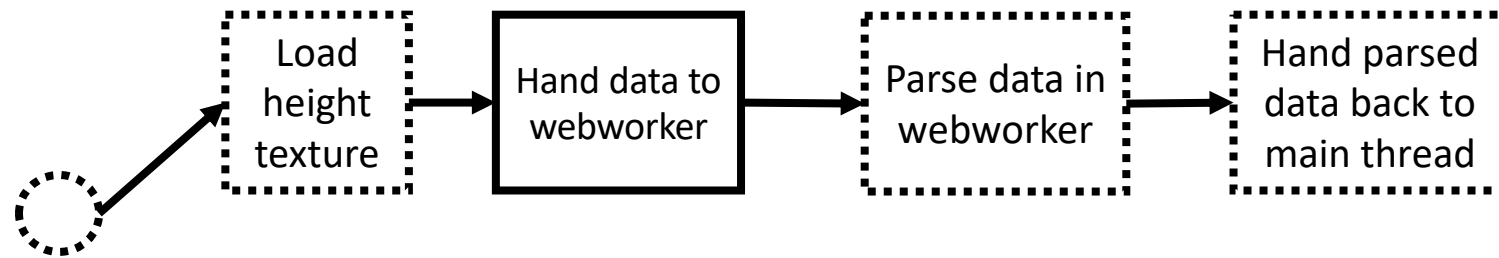
# Implementation: Async Data Loading

Individually executed for each M-block



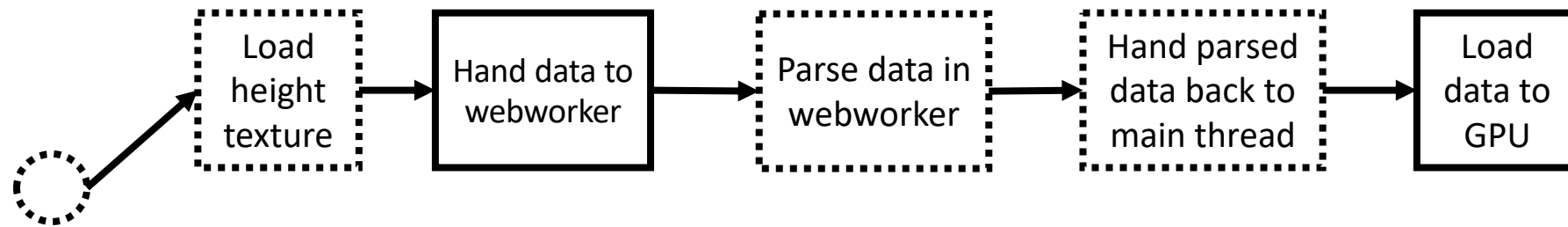
# Implementation: Async Data Loading

Individually executed for each M-block



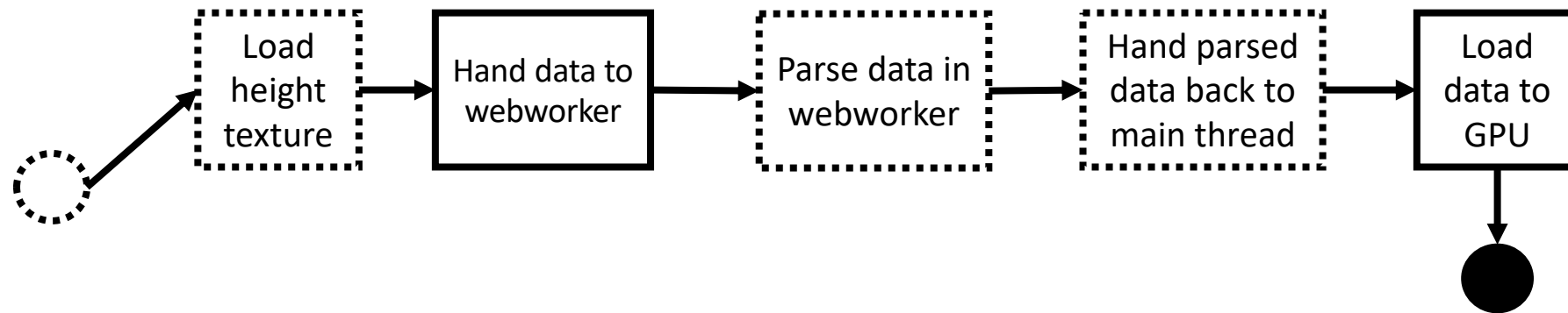
# Implementation: Async Data Loading

Individually executed for each M-block



# Implementation: Async Data Loading

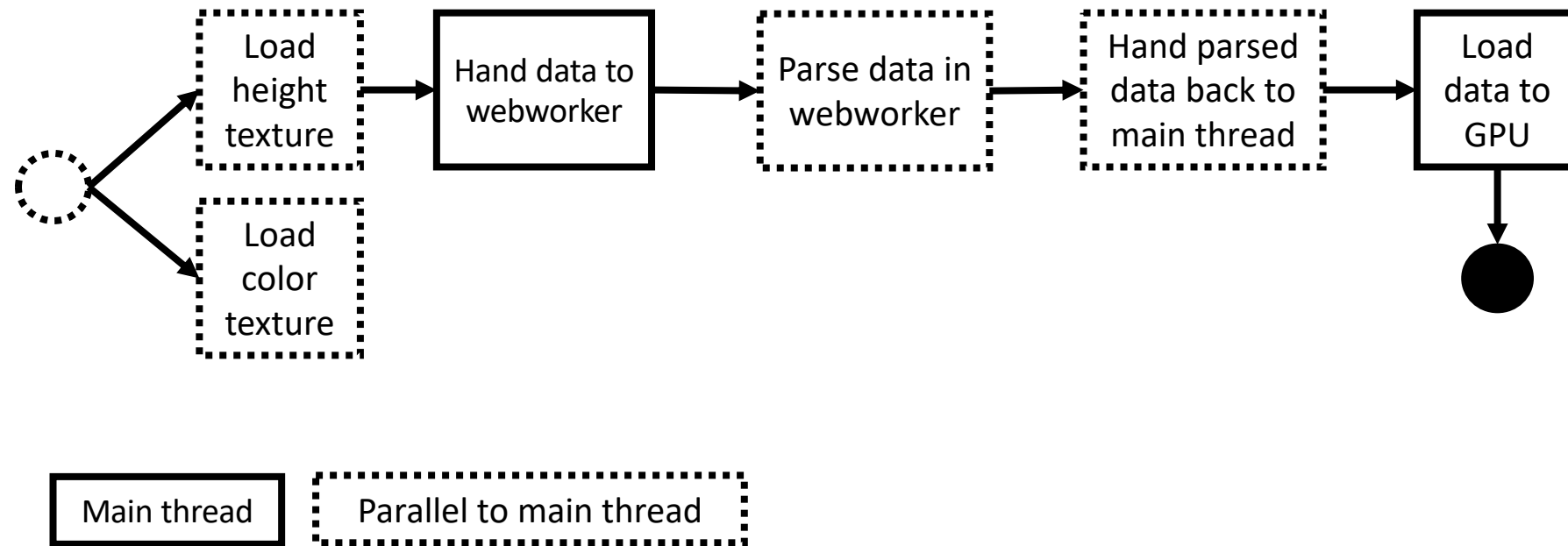
Individually executed for each M-block





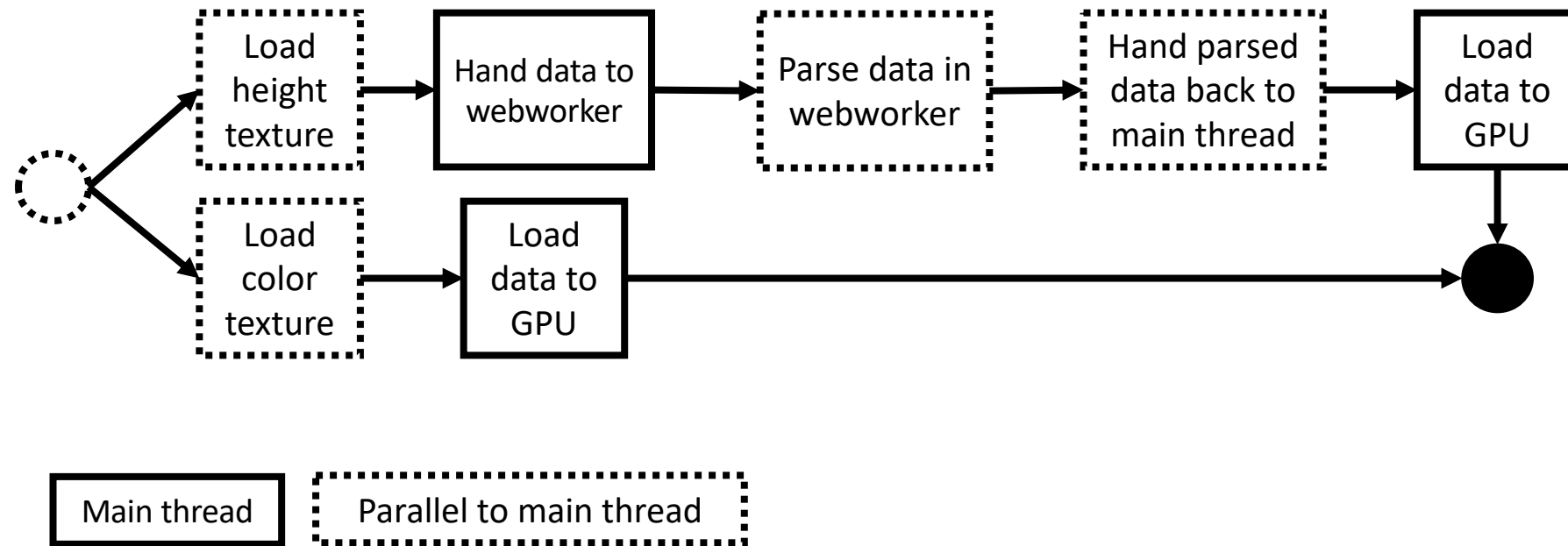
# Implementation: Async Data Loading

Individually executed for each M-block

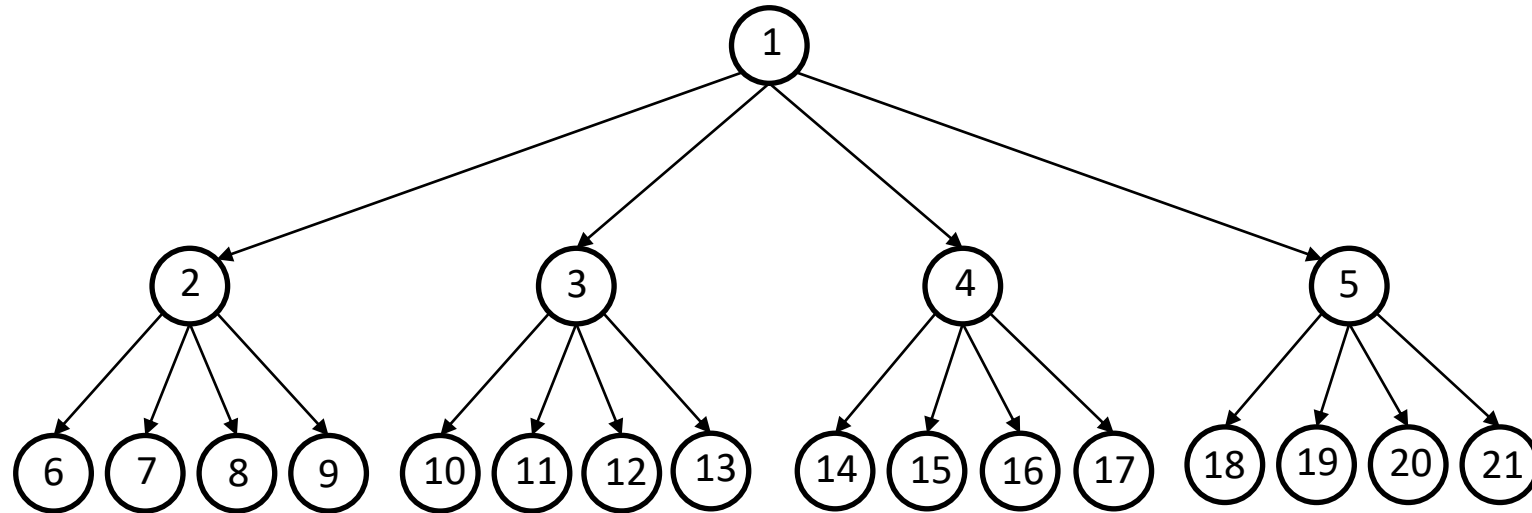


# Implementation: Async Data Loading

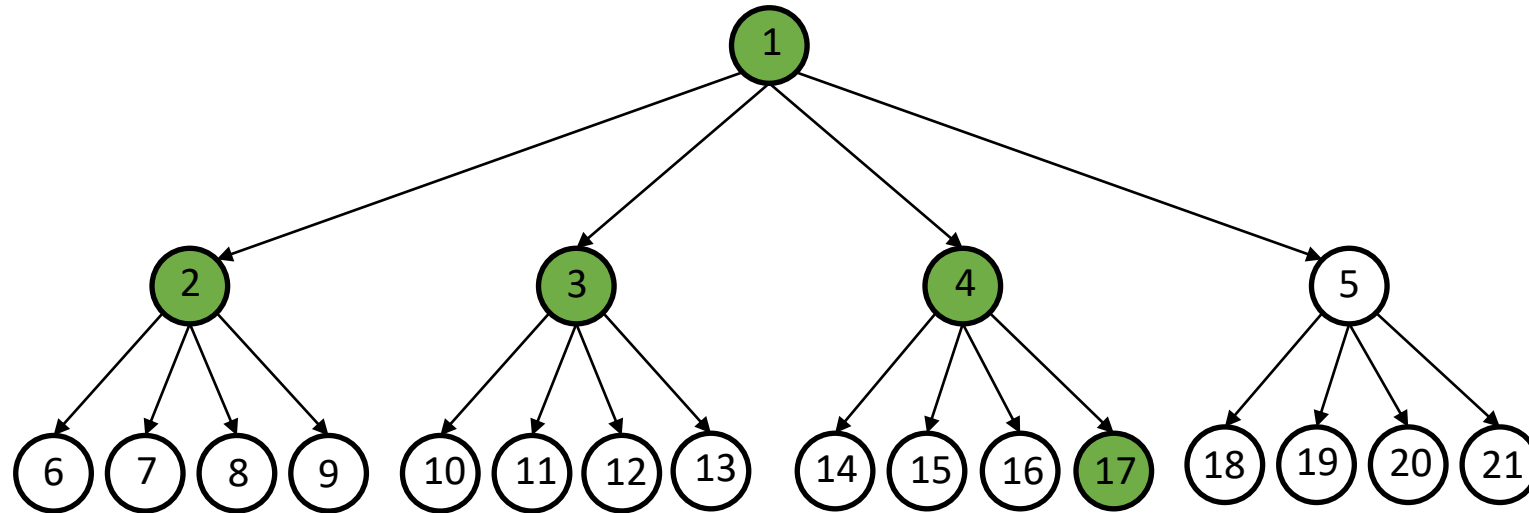
Individually executed for each M-block



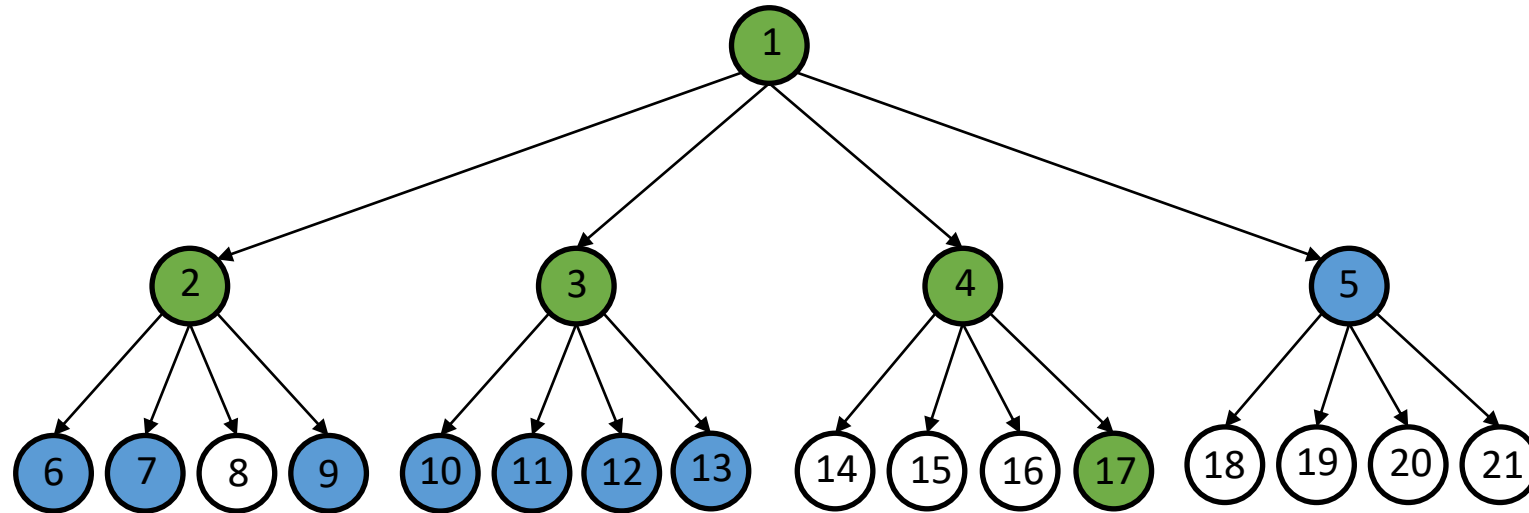
# Implementation: Client Data Management



# Implementation: Client Data Management



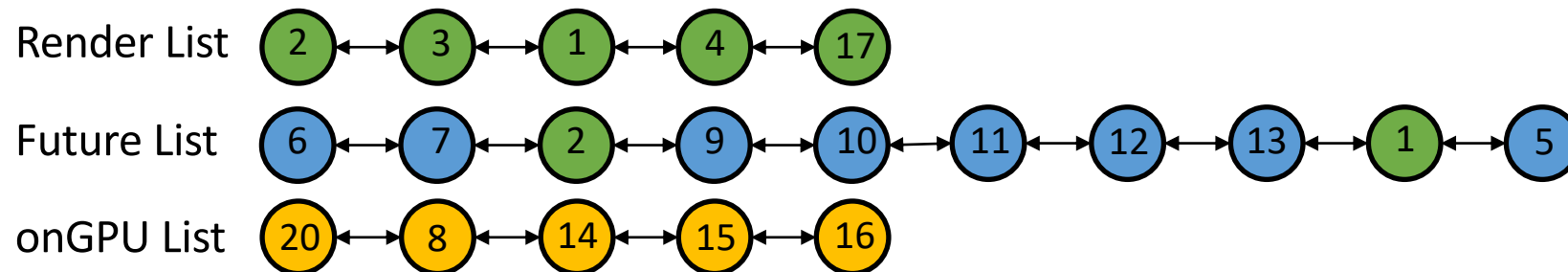
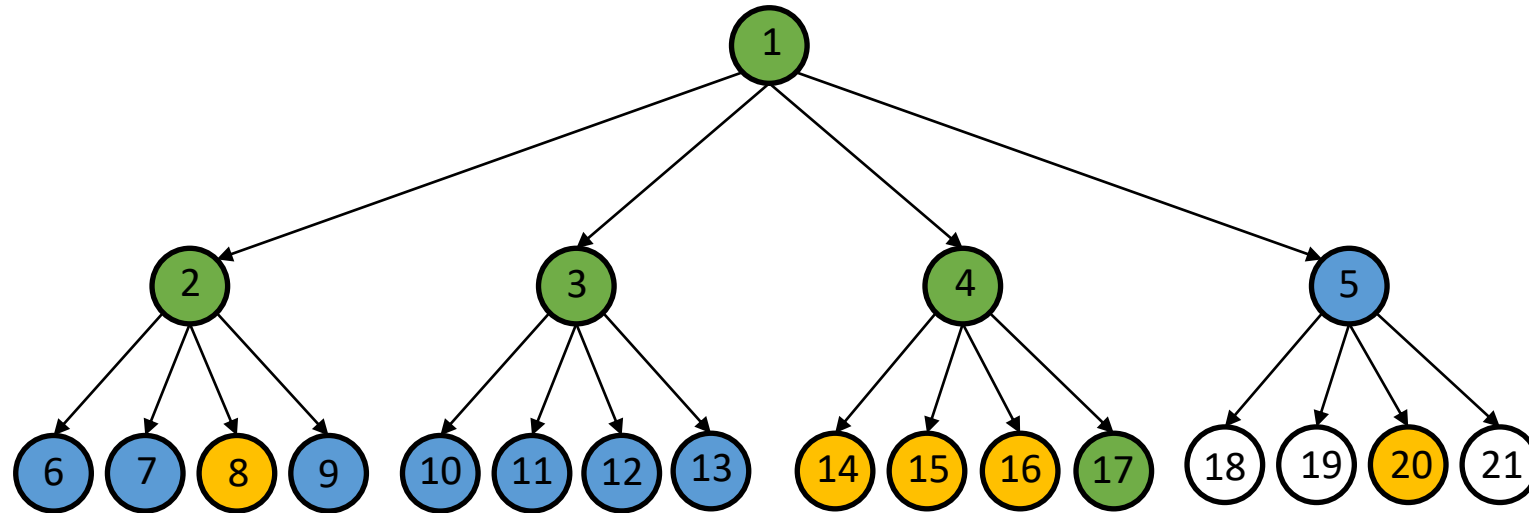
# Implementation: Client Data Management



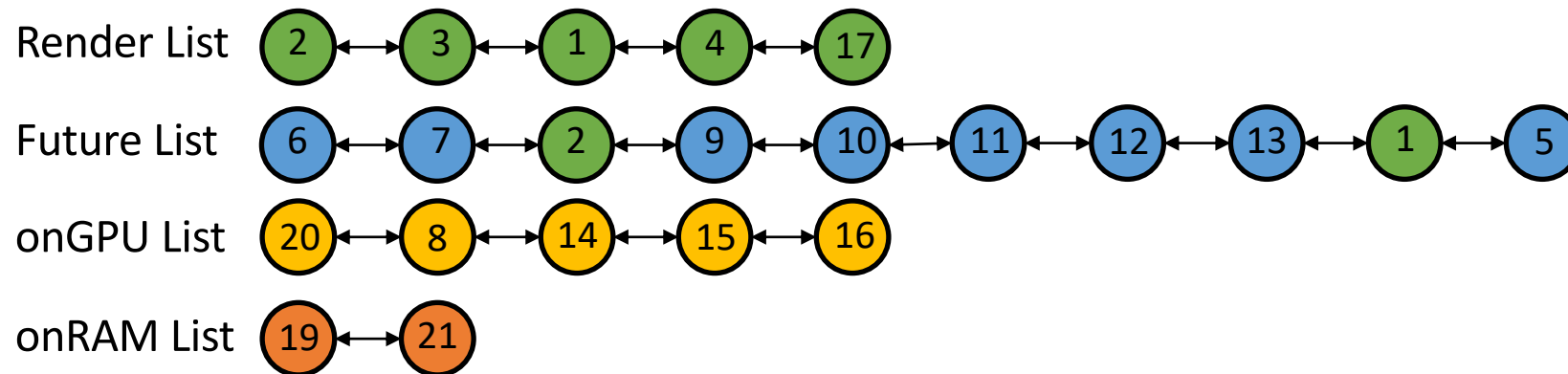
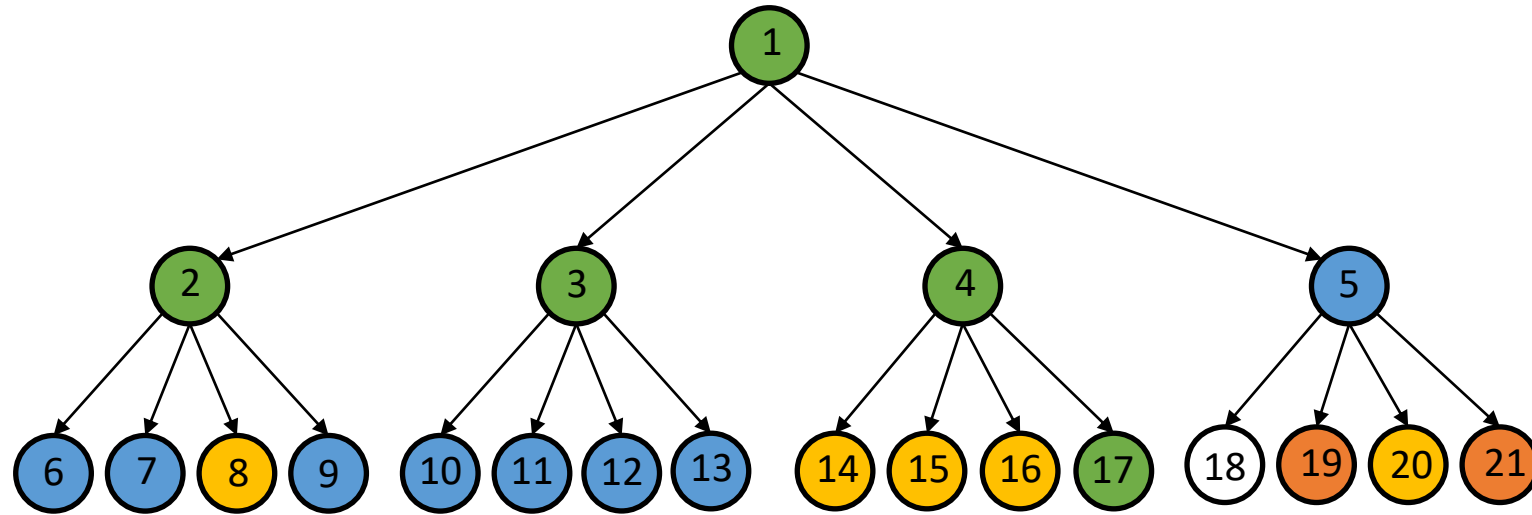
Render List 2 ↔ 3 ↔ 1 ↔ 4 ↔ 17

Future List 6 ↔ 7 ↔ 2 ↔ 9 ↔ 10 ↔ 11 ↔ 12 ↔ 13 ↔ 1 ↔ 5

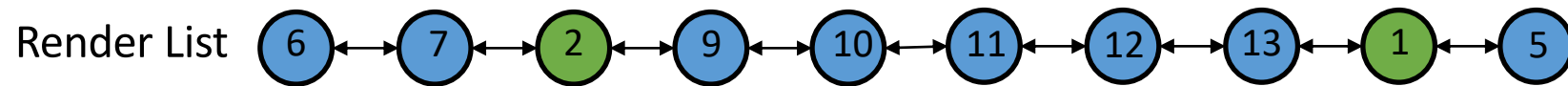
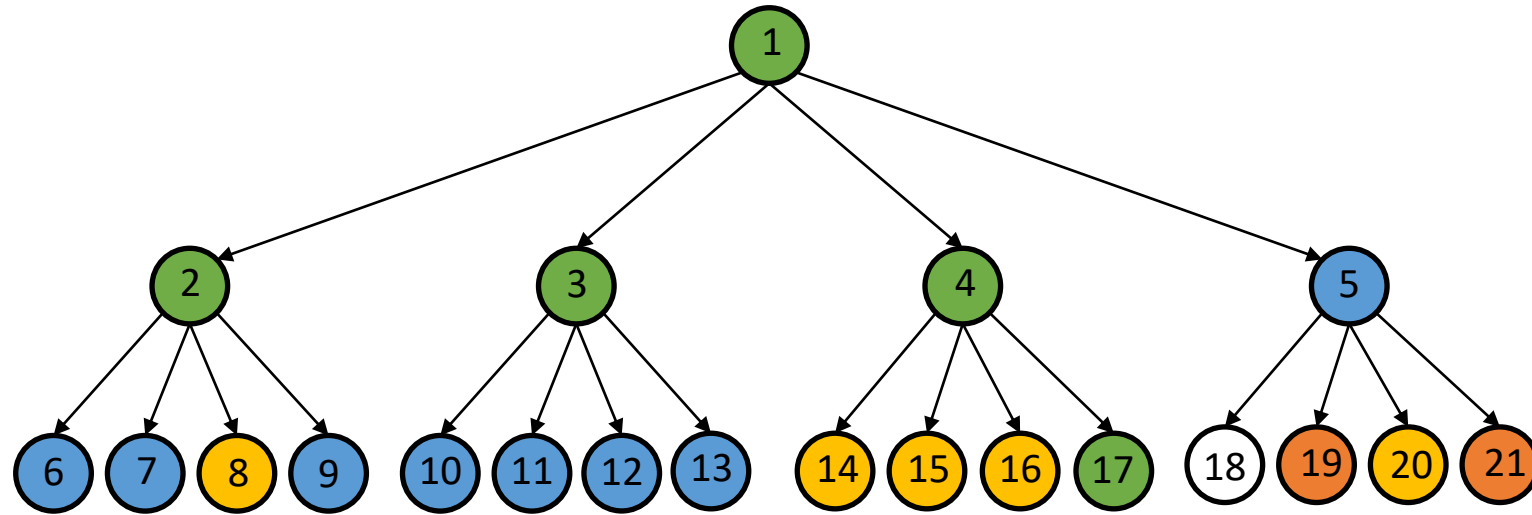
# Implementation: Client Data Management



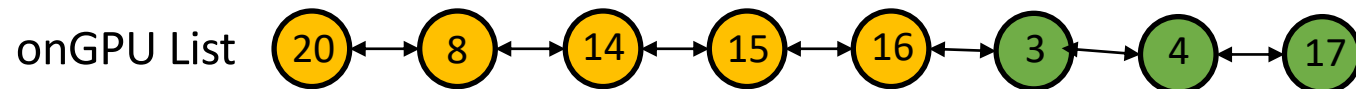
# Implementation: Client Data Management



# Implementation: Client Data Management

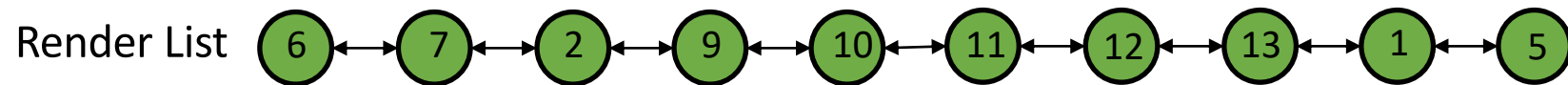
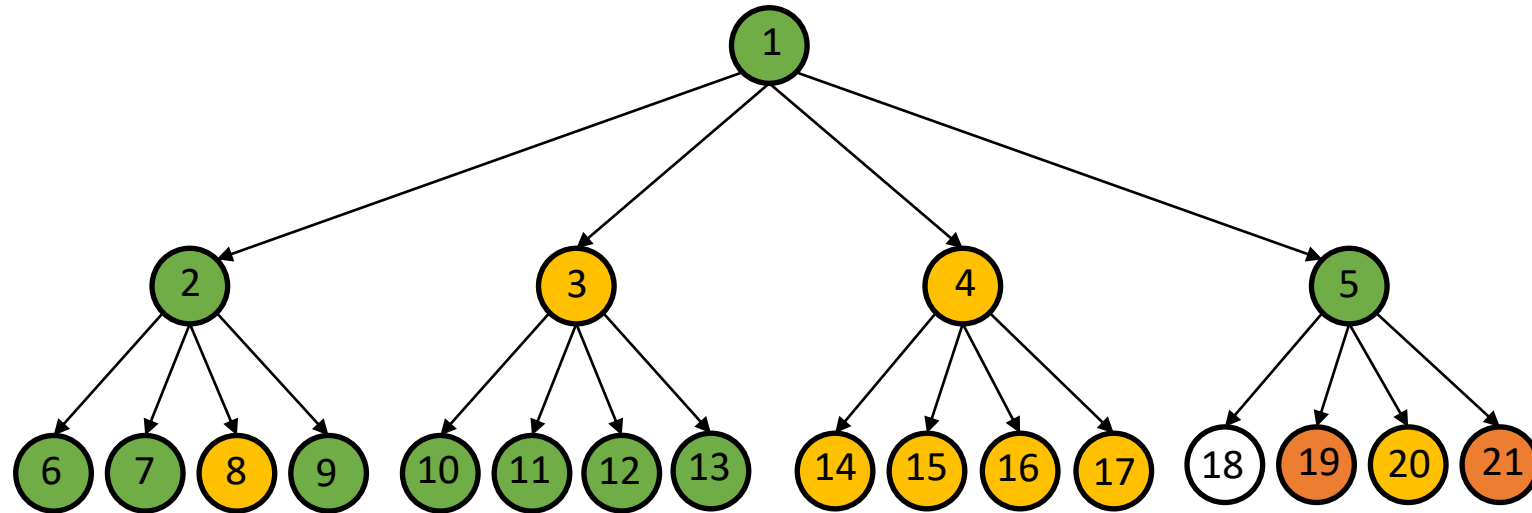


Future List

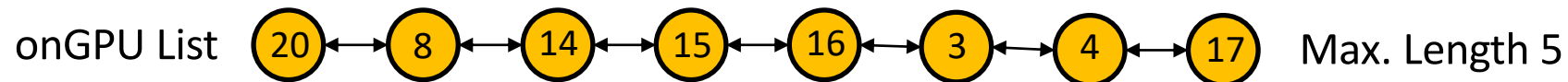




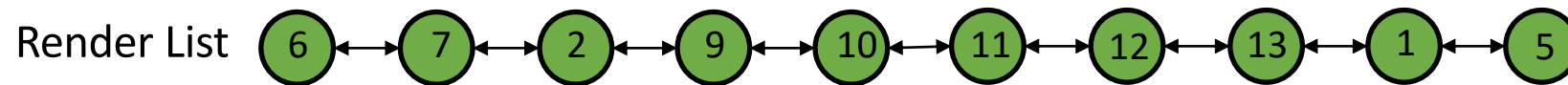
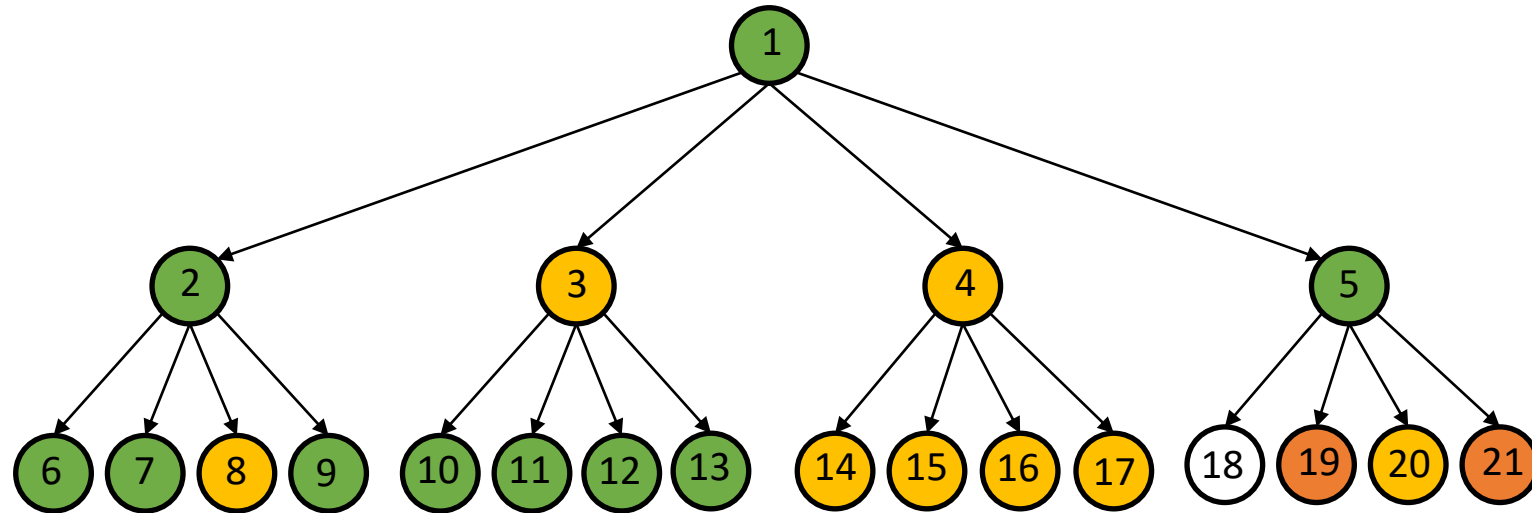
# Implementation: Client Data Management



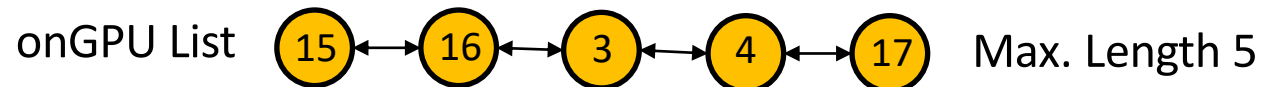
Future List



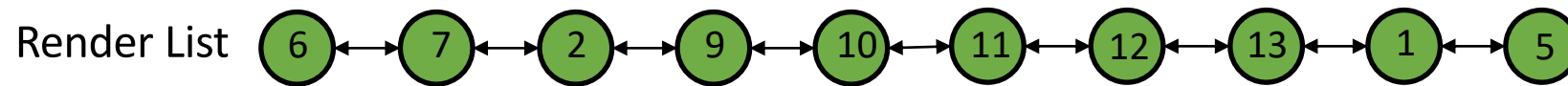
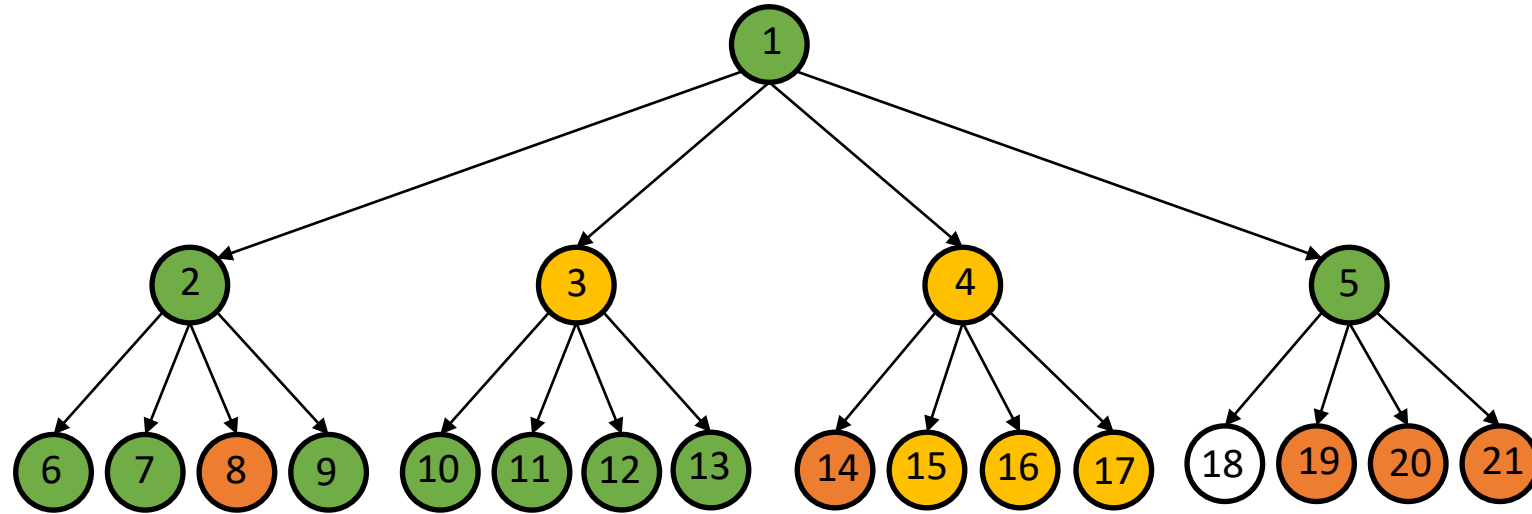
# Implementation: Client Data Management



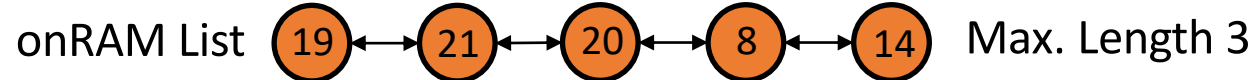
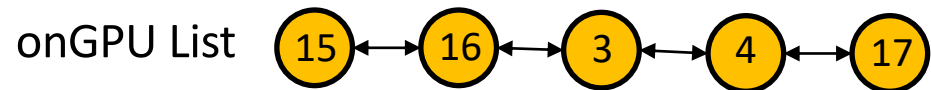
Future List



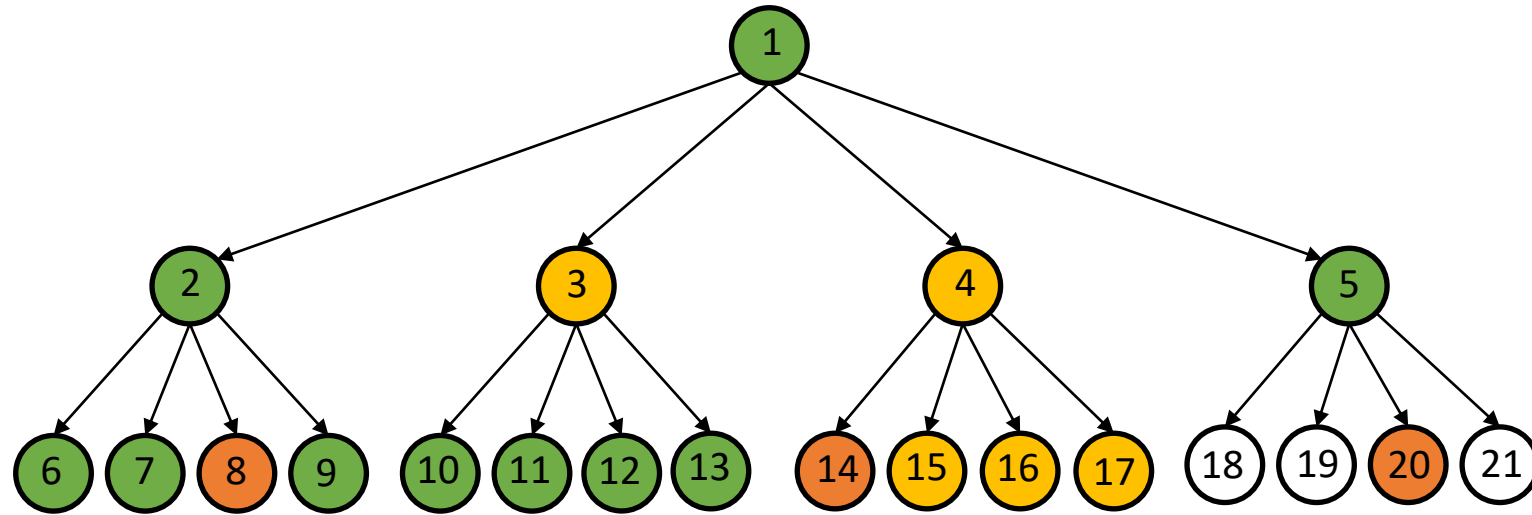
# Implementation: Client Data Management



Future List



# Implementation: Client Data Management



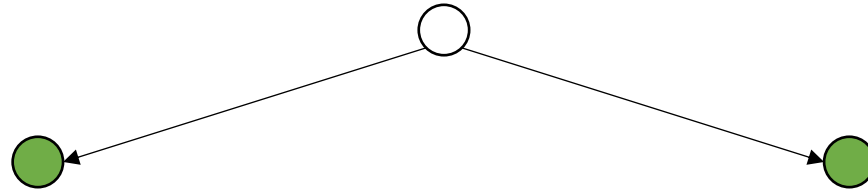
Render List 6 ↔ 7 ↔ 2 ↔ 9 ↔ 10 ↔ 11 ↔ 12 ↔ 13 ↔ 1 ↔ 5

Future List

onGPU List 15 ↔ 16 ↔ 3 ↔ 4 ↔ 17

onRAM List 20 ↔ 8 ↔ 14 Max. Length 3

# Implementation: Dynamic Updates



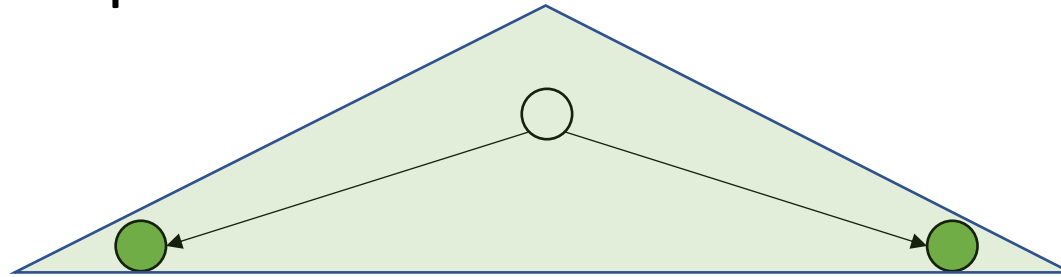
# Implementation: Dynamic Updates

Required M-Blocks: 1

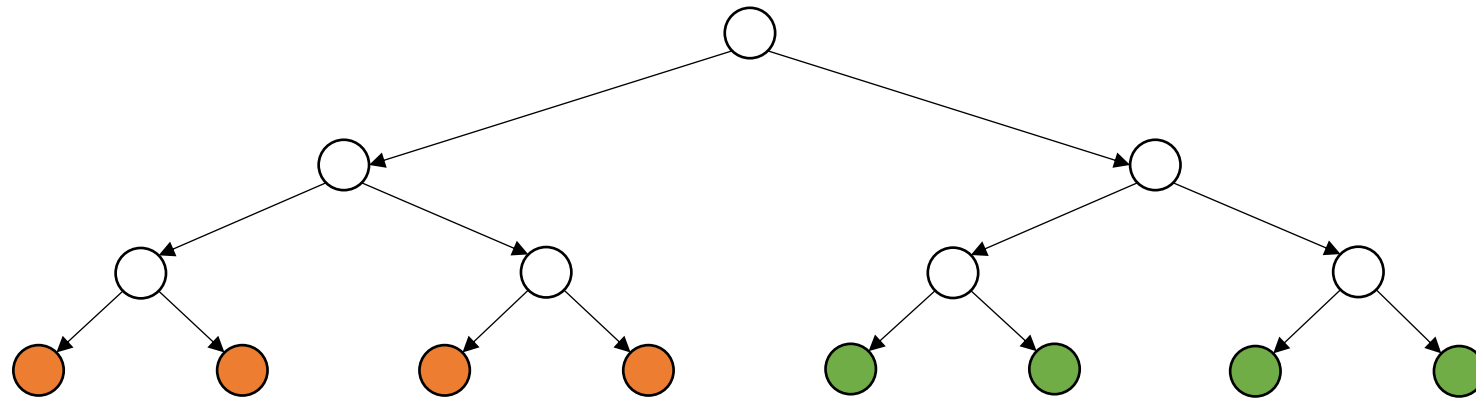
Not Ready M-Blocks: 0

Ratio: Not Ready / Required: 0

Threshold: 0.6



# Implementation: Dynamic Updates



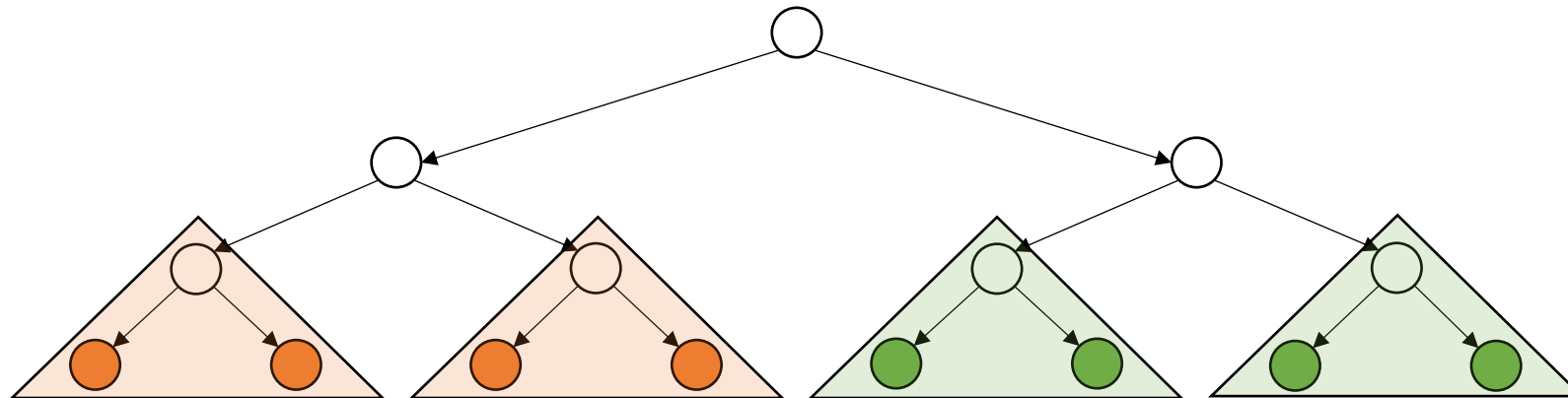
# Implementation: Dynamic Updates

Required M-Blocks: 4

Not Ready M-Blocks: 2

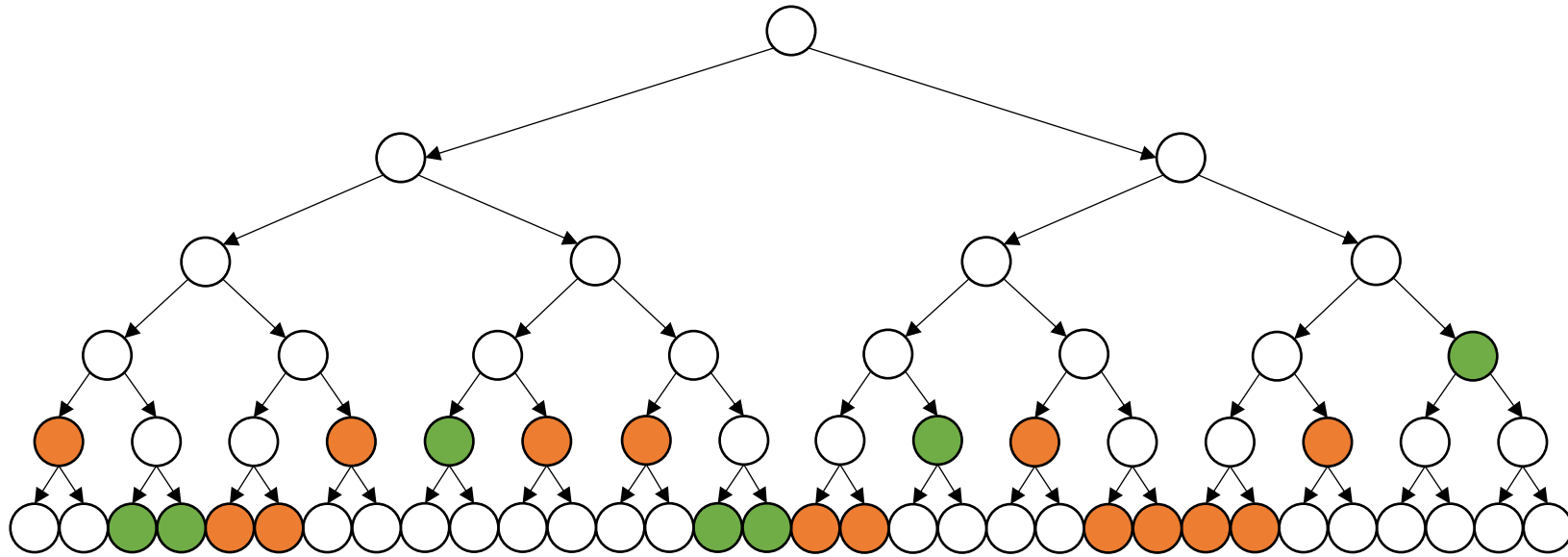
Ratio: Not Ready / Required: 0.5

Threshold: 0.6





# Implementation: Dynamic Updates



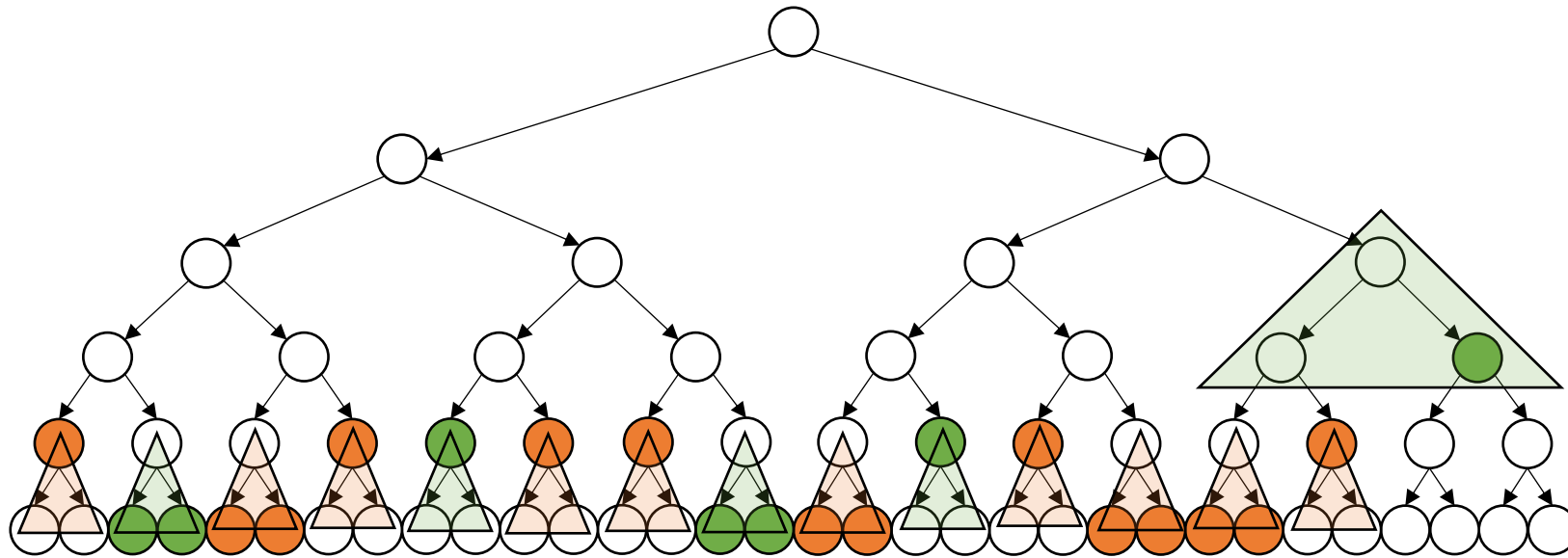
# Implementation: Dynamic Updates

Required M-Blocks: 15

Not Ready M-Blocks: 10

Ratio: Not Ready / Required: 0.67

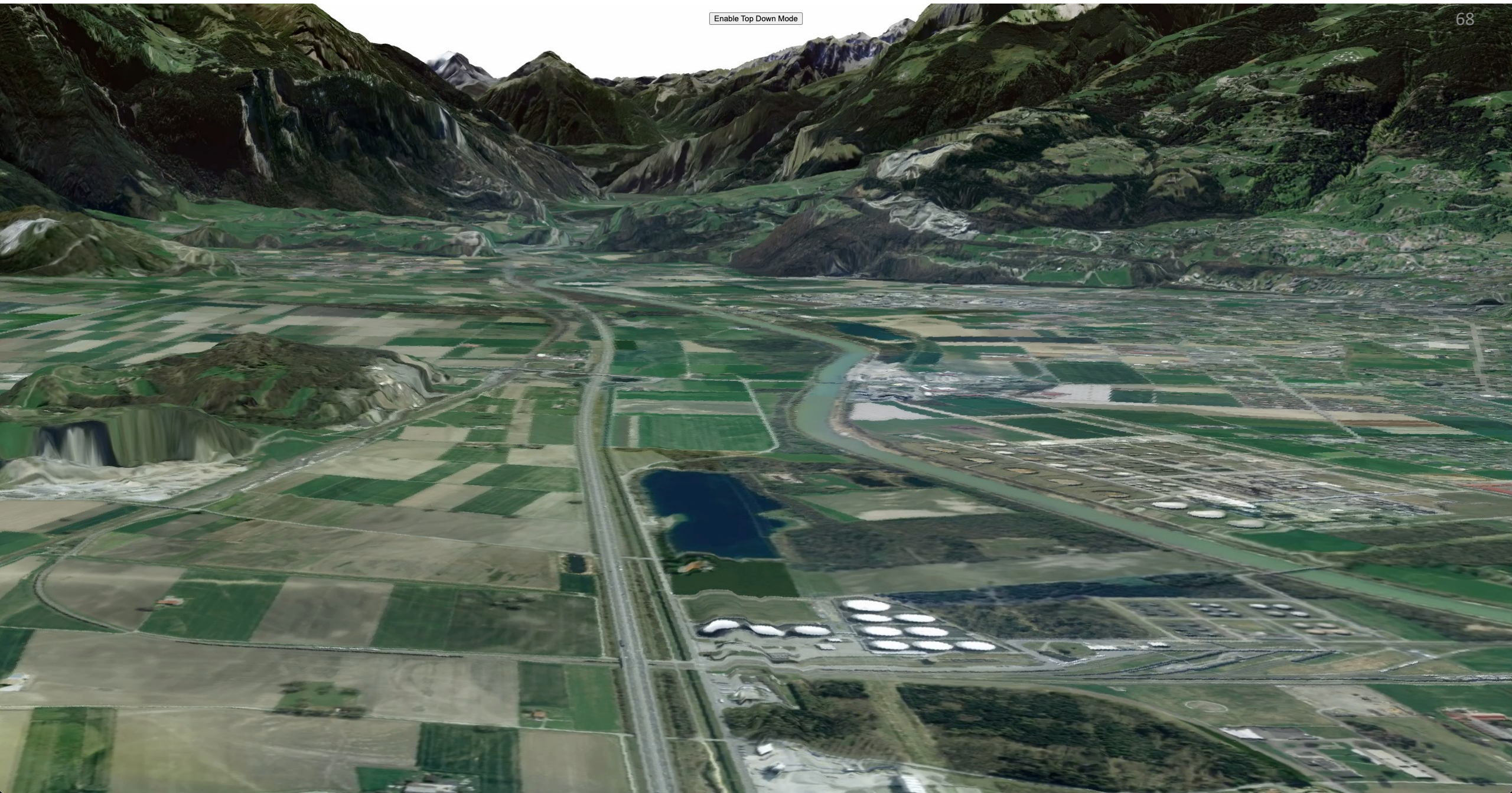
Threshold: 0.6



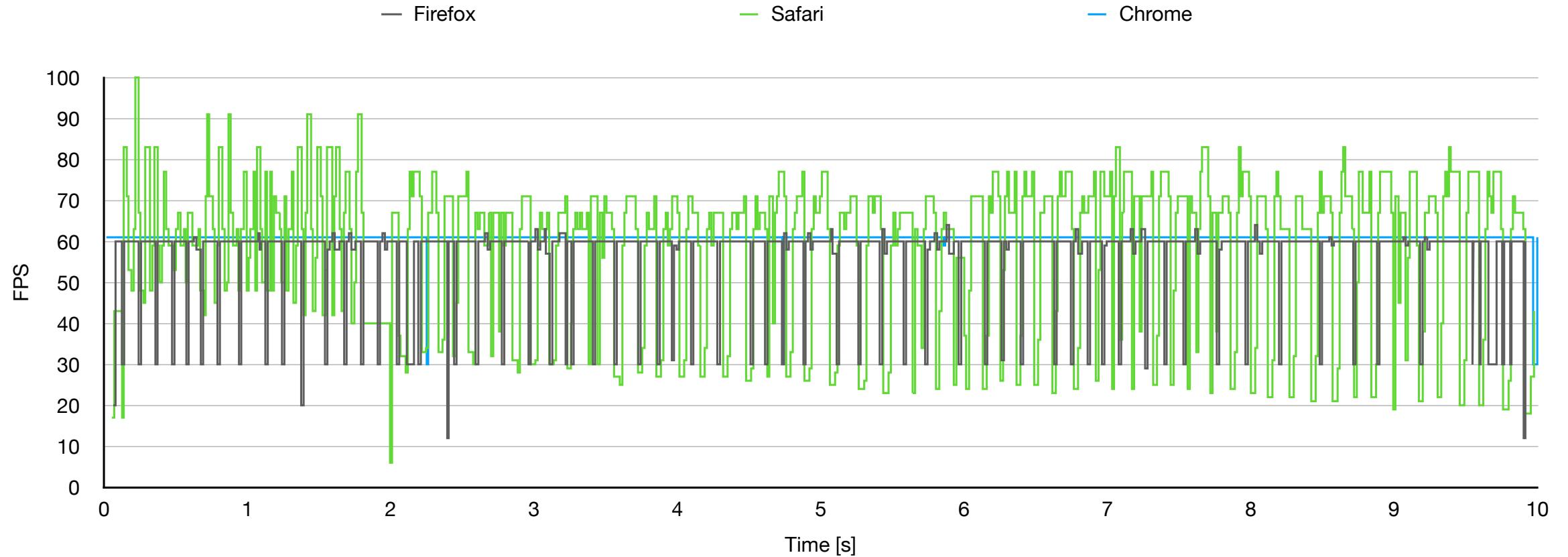
# Results: Bandwidth no limitations

Device: MacBook Pro M1 Max, 64GB RAM, Chrome Browser

Enable Top Down Mode



# Results: Bandwidth no limitations



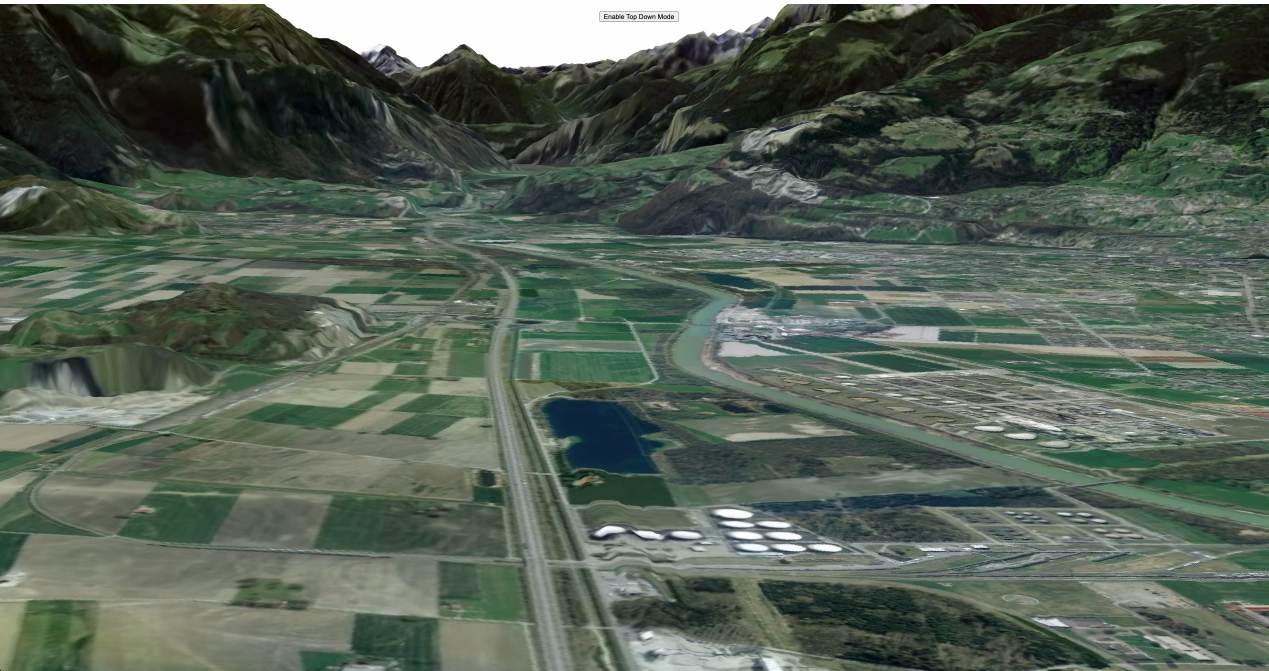
Device: MacBook Pro M1 Max, 64GB Ram

# Results: Dynamic Update (40Mb/s)

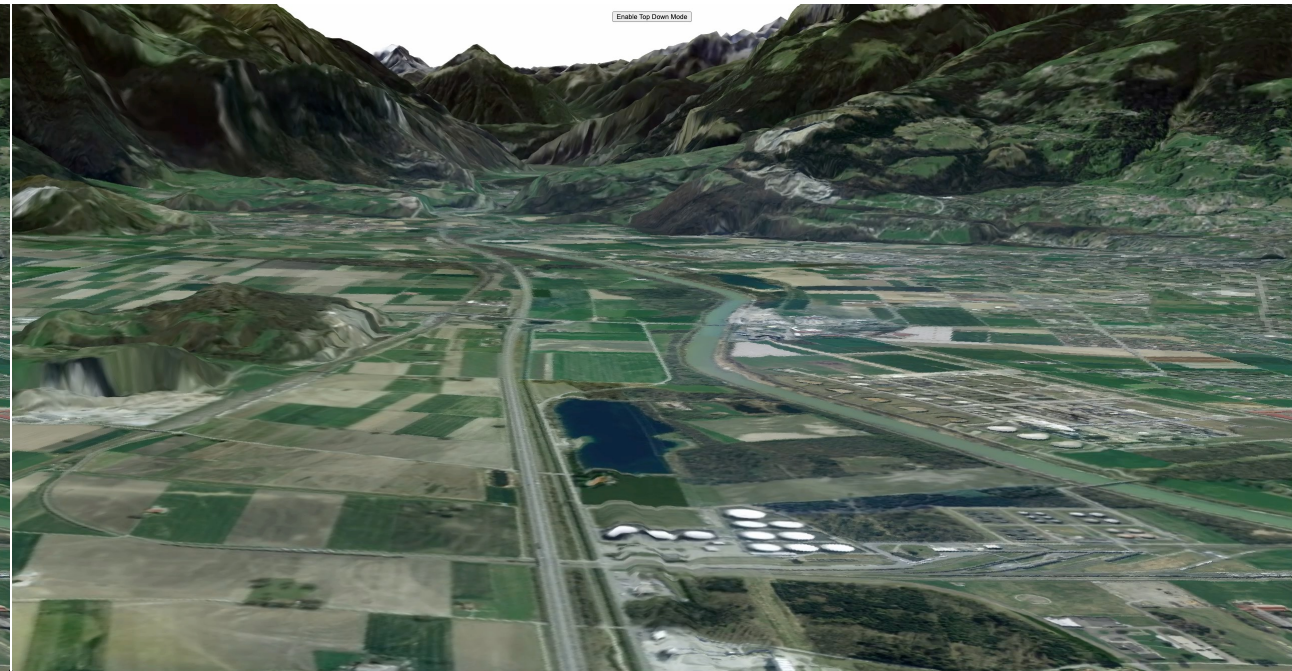
Video 40Mb/s without (left) and with (right) dynamic updates

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

# Results: Dynamic Update (40Mb/s)



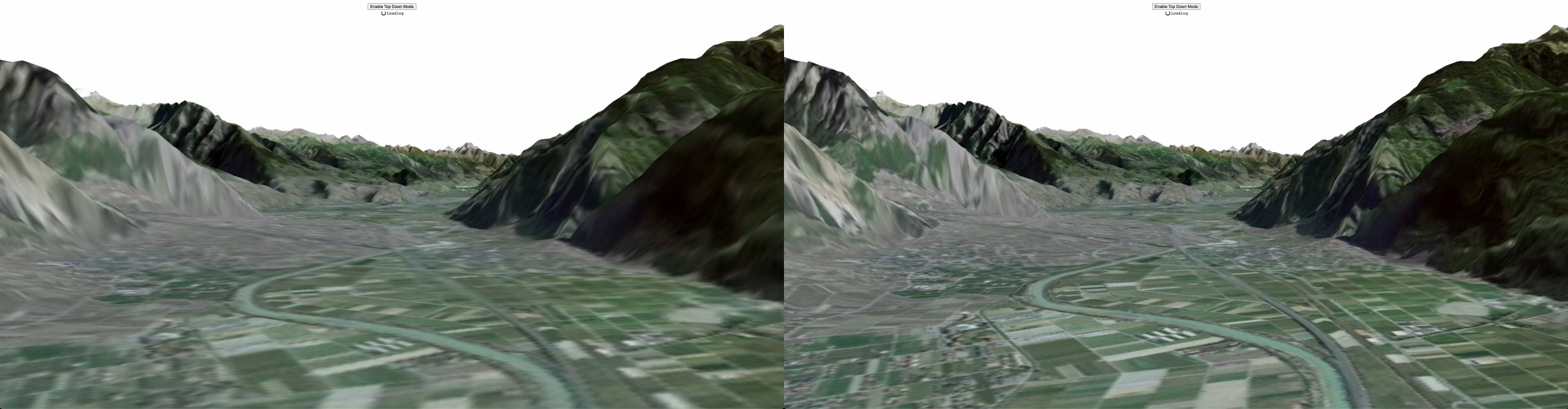
No Dynamic Updates



Dynamic Updates

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

# Results: Dynamic Update (40Mb/s)



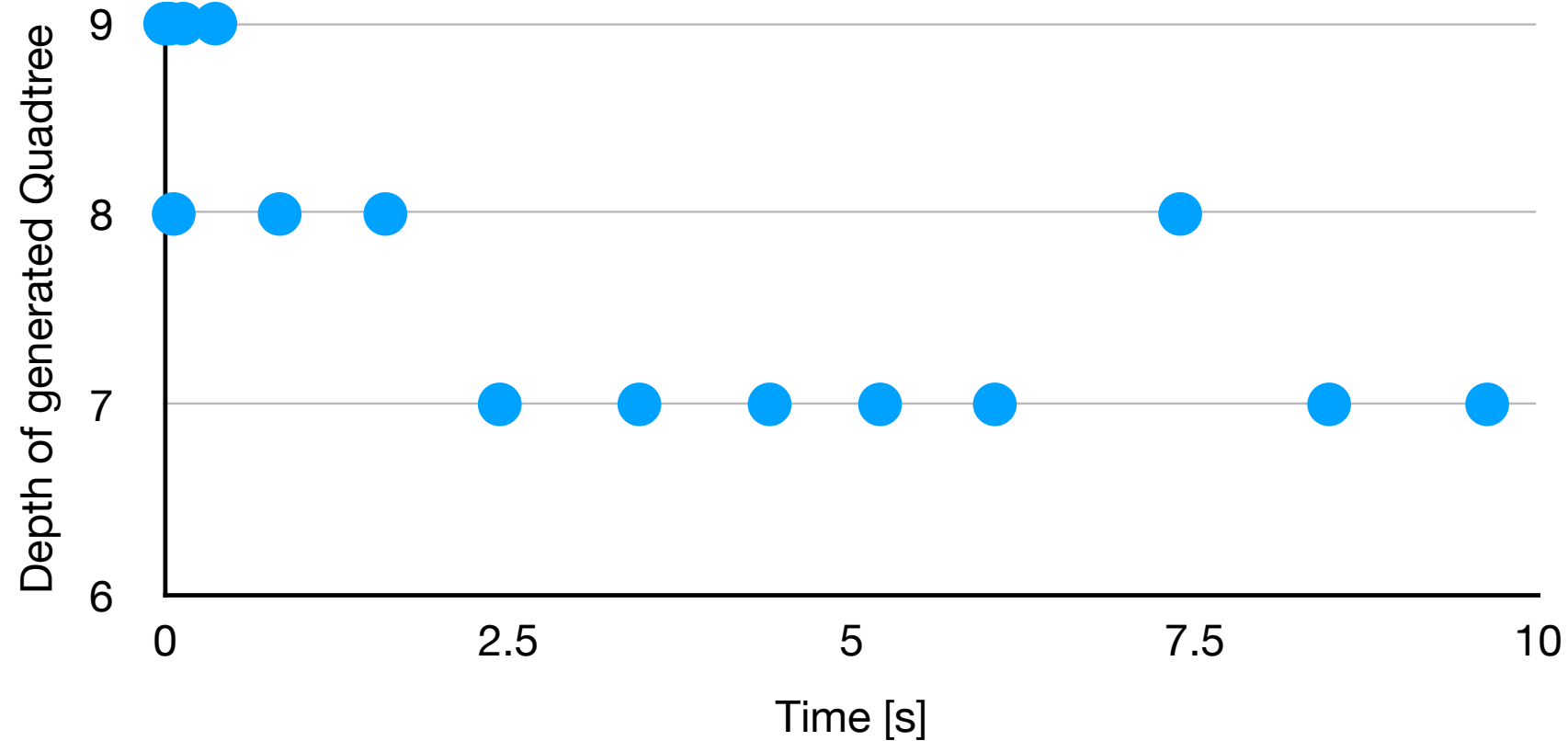
No Dynamic Updates

Dynamic Updates

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

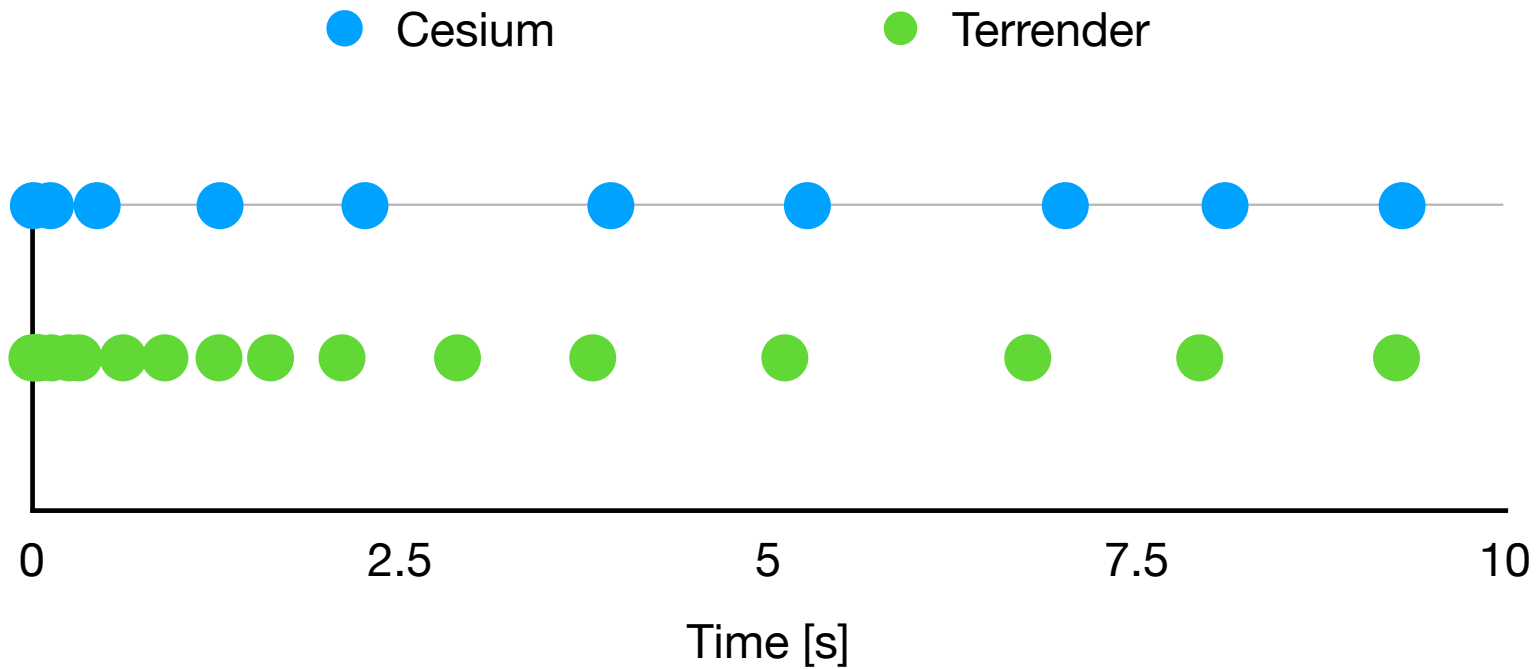


# Results: Dynamic Update (40Mb/s)

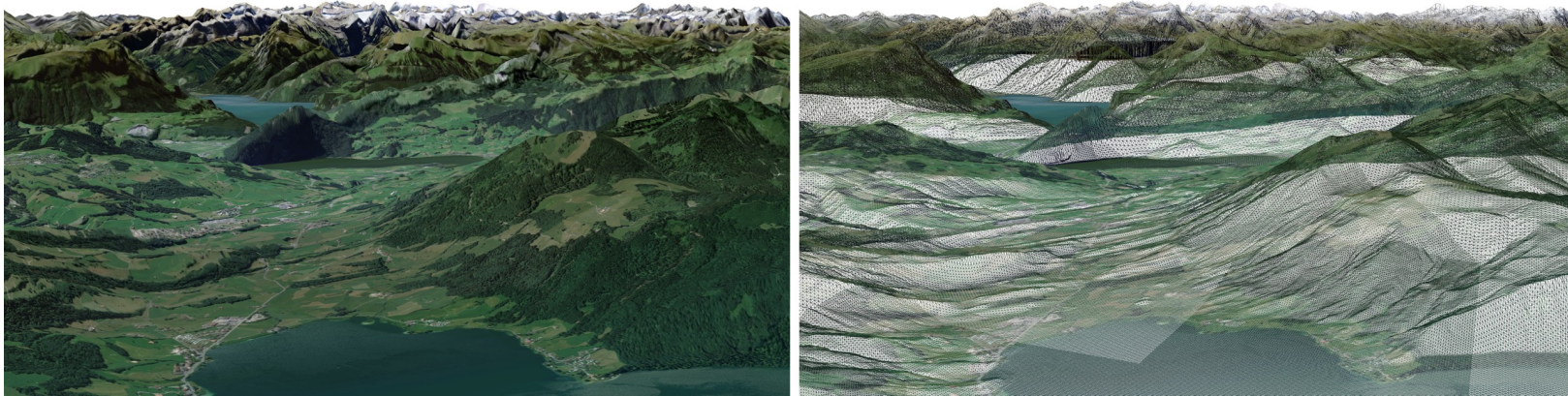


Device: MacBook Pro M1 Max, 64GB Ram, Chrome Browser

# Comparison with CesiumJS



Terrain updates during the same trajectory



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

Thank you for watching