

# Multi-Resolution Meshes for Feature-Aware Hardware Tessellation

Thibaud Lambert, Pierre Bénard, Gaël Guennebaud



# Motivation

## Rendering of

- highly detailed meshes
- in highly complex scene



## Level of details

1,6M



100k



25k



1500



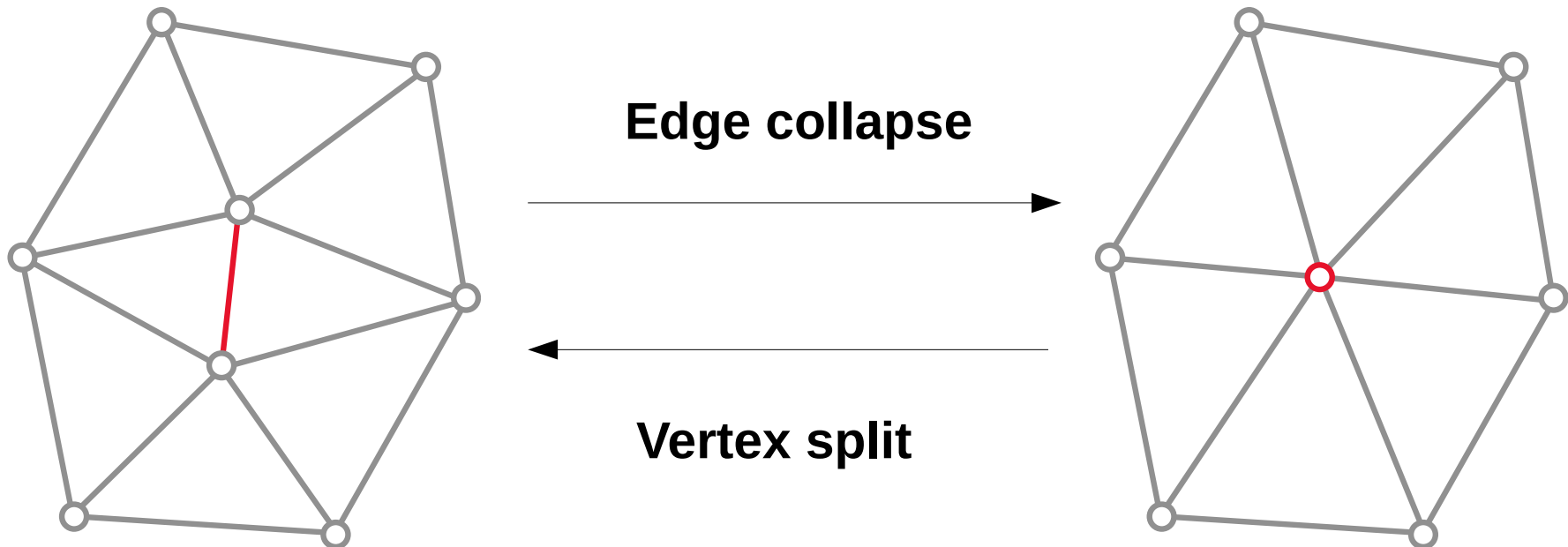
# Level of details

## Challenges

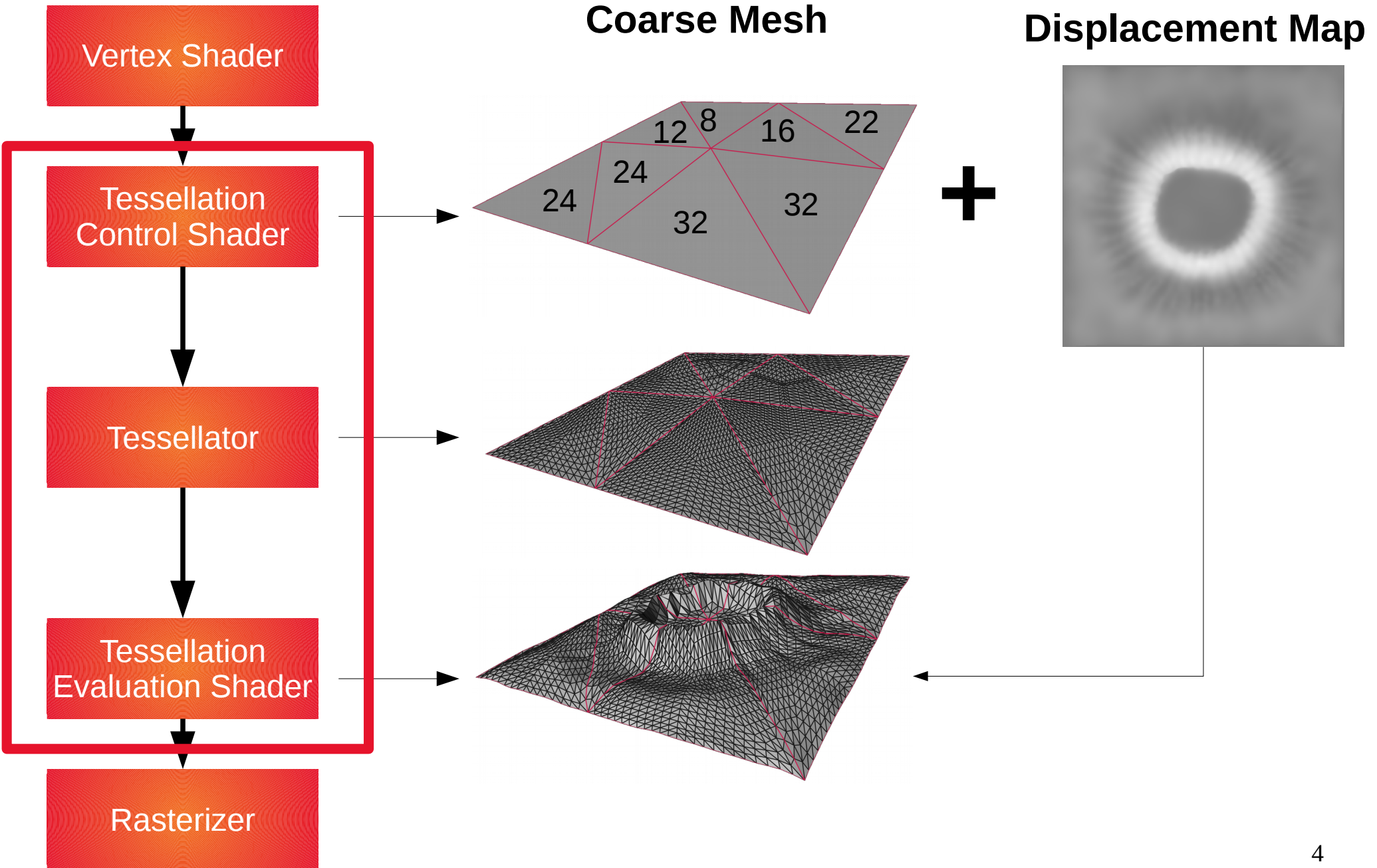
- Smooth transition
- Efficient rendering

## Progressive Mesh [Hop97] [LSH10]

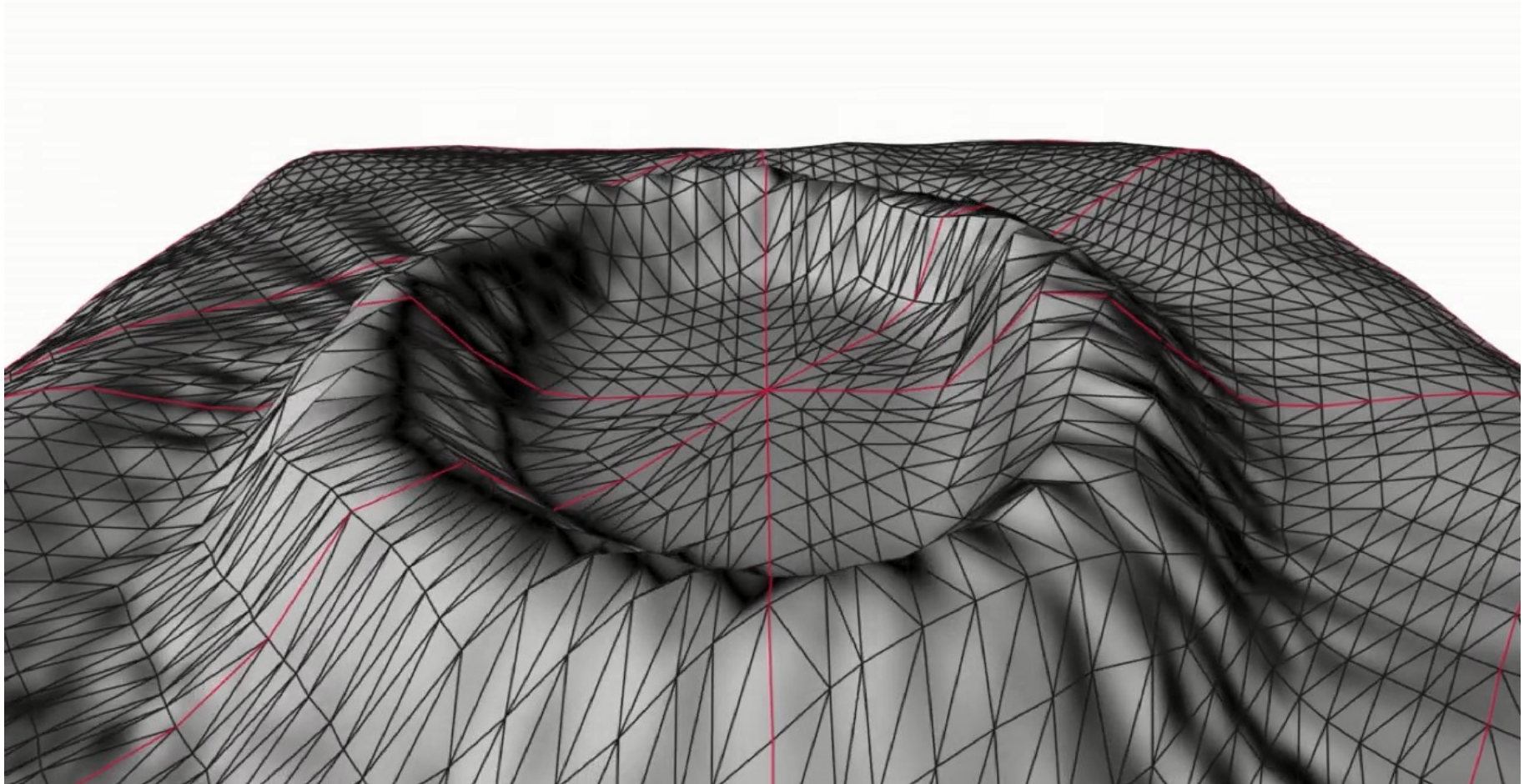
- + Smooth transition
- + Highly flexible
- Not GPU friendly



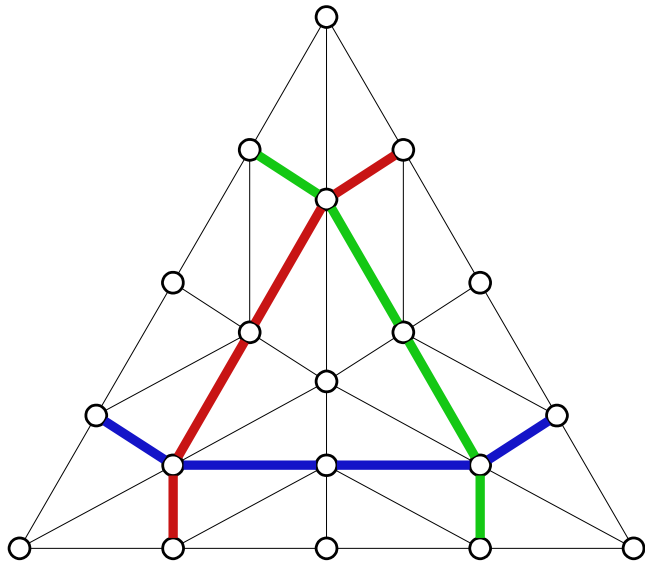
# Displacement on Hardware Tessellation



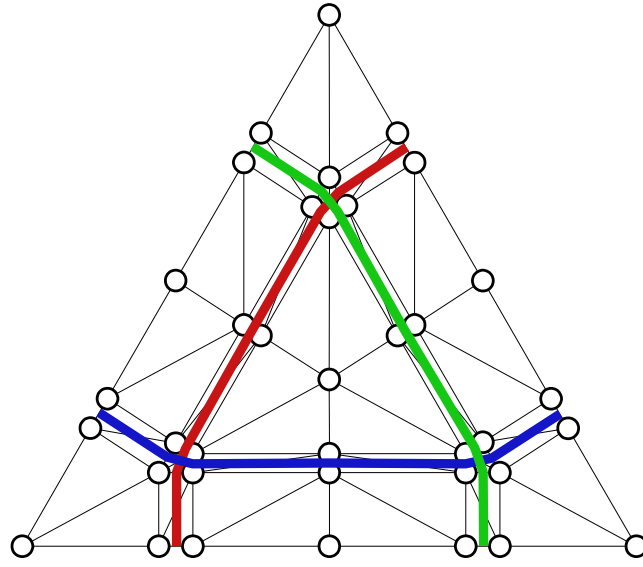
# Fractional Tessellation



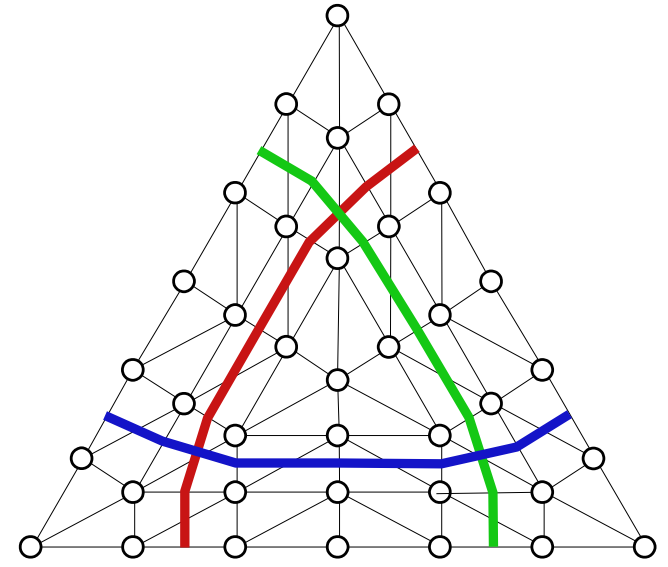
# Swimming artifacts



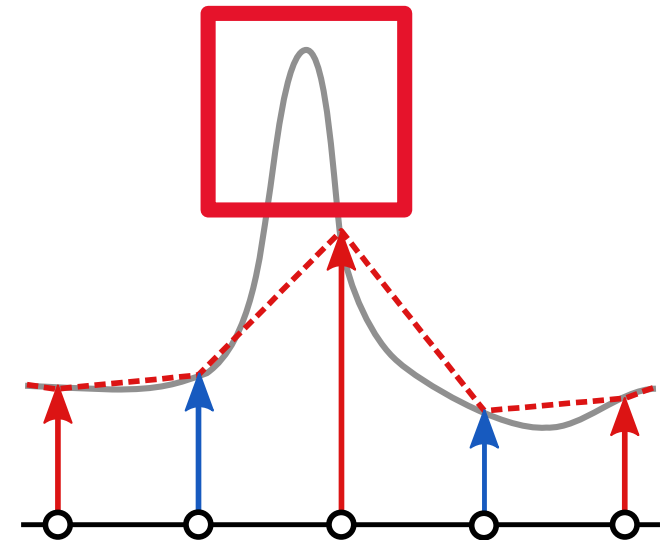
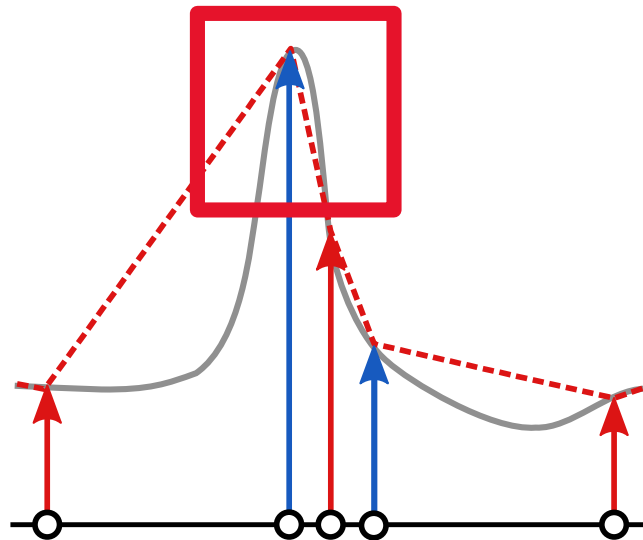
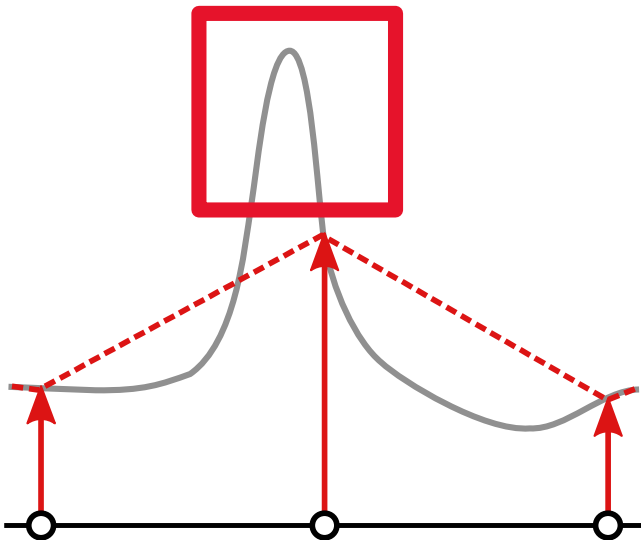
**Factors 4**



**Fractional Factors 4.9**

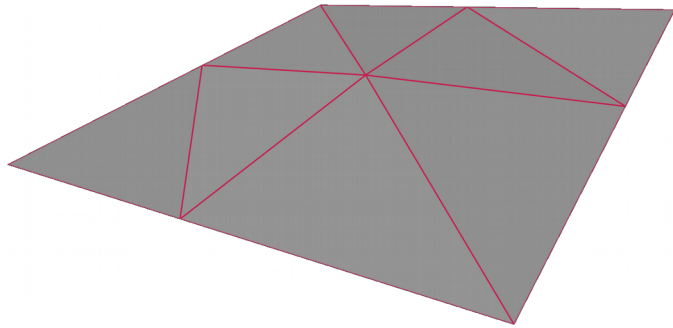


**Factors 6**



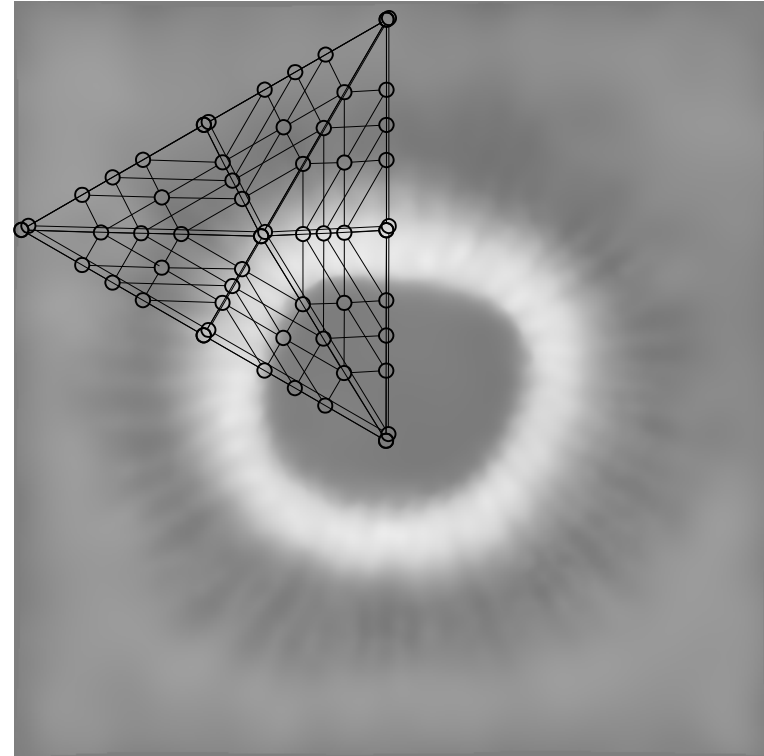
# Uniform Tessellation

**Coarse Mesh**



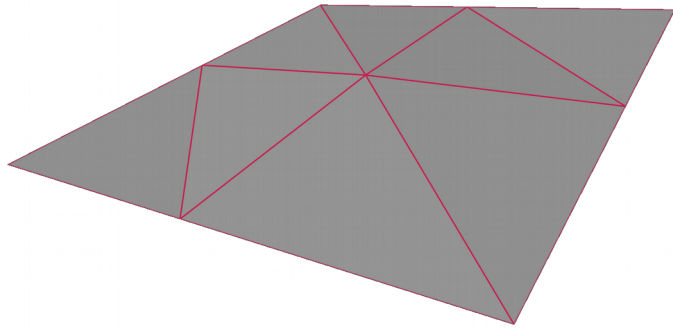
+

**Displacement Map**



# Uniform Tessellation

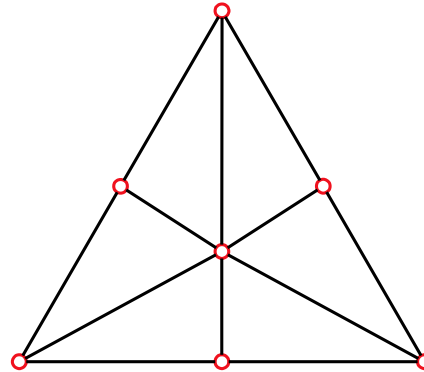
Coarse Mesh



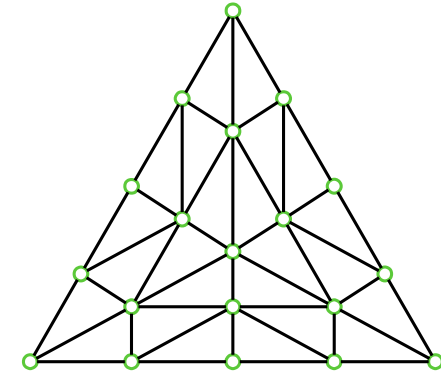
+

Multi-resolution Attributes [SPM13]

Level 2



Level 4



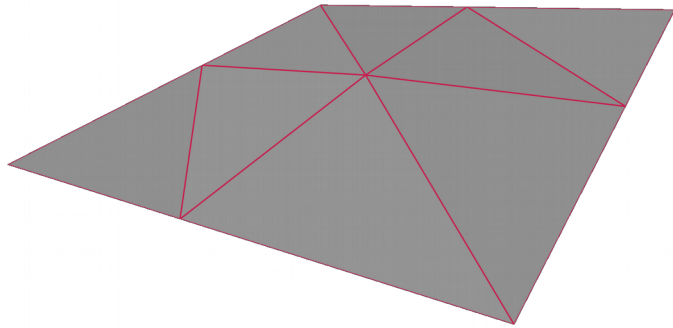
Per vertex storage

+ One to one mapping



# Uniform Tessellation

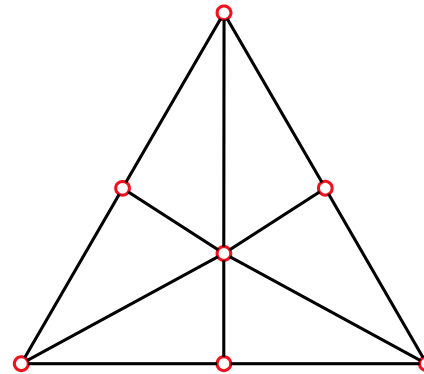
Coarse Mesh



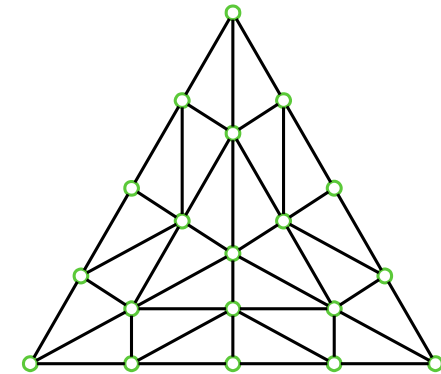
+

Multi-resolution Attributes [SPM13]

Level 2

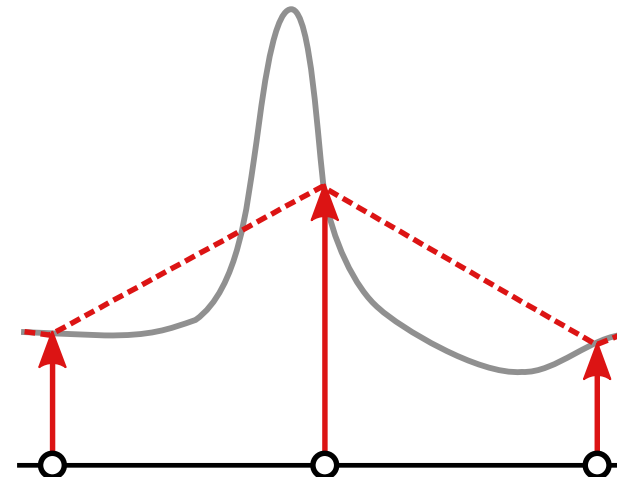


Level 4



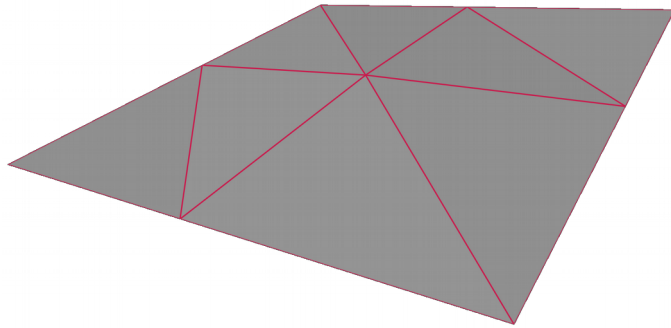
**Per vertex storage**

- + One to one mapping
- + Swimming free



# Uniform Tessellation

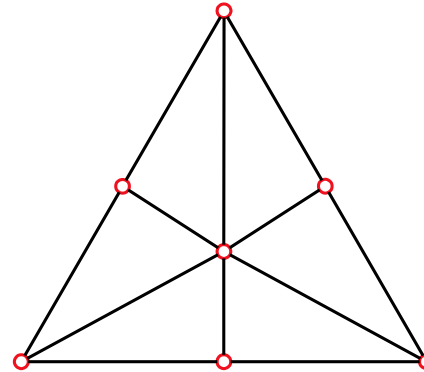
Coarse Mesh



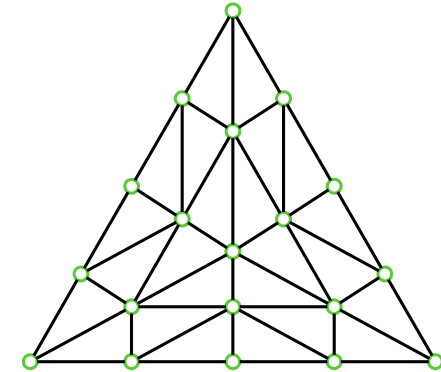
+

Multi-resolution Attributes [SPM13]

Level 2

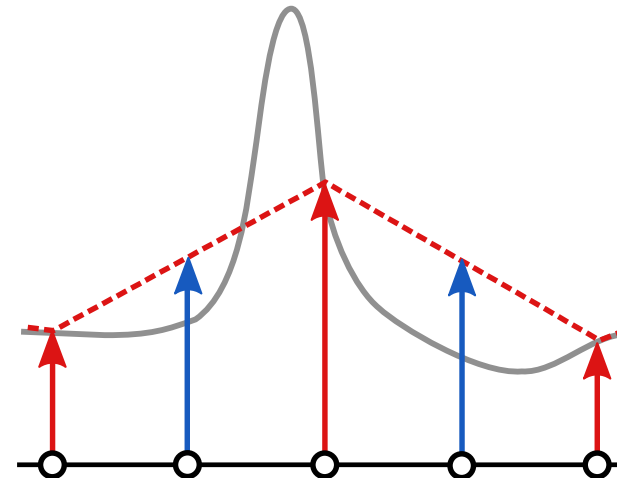


Level 4



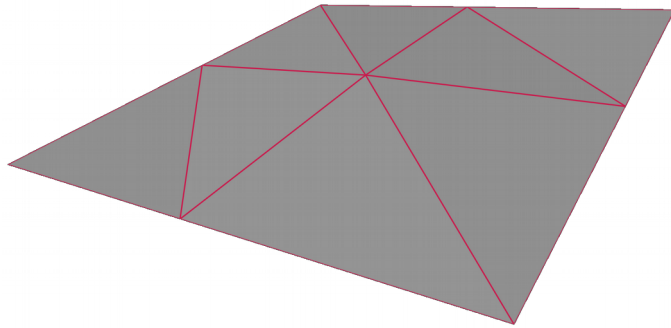
**Per vertex storage**

- + One to one mapping
- + Swimming free



# Uniform Tessellation

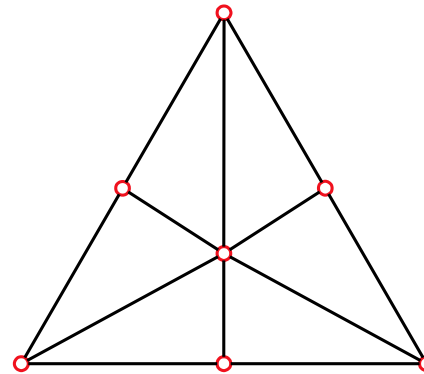
Coarse Mesh



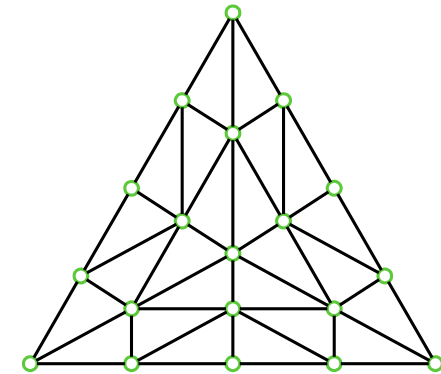
+

Multi-resolution Attributes [SPM13]

Level 2

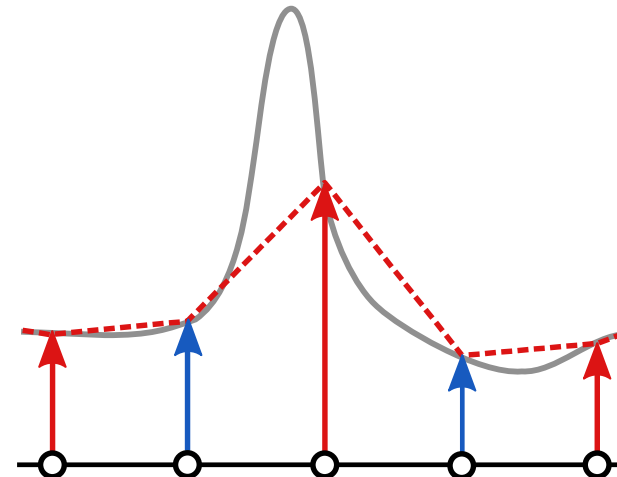


Level 4



## Per vertex storage

- + One to one mapping
- + Swimming free
- Uniform transition



# Objectives

## Progressive Mesh

- + Flexible transition
- Not GPU friendly

## Multi-resolution attributes

- Uniform transition
- + Hardware tessellation

## Contributions

- Relax tessellation constraint
- LOD Generation

# Objectives

## Progressive Mesh

- + Flexible transition
- Not GPU friendly

## Multi-resolution attributes

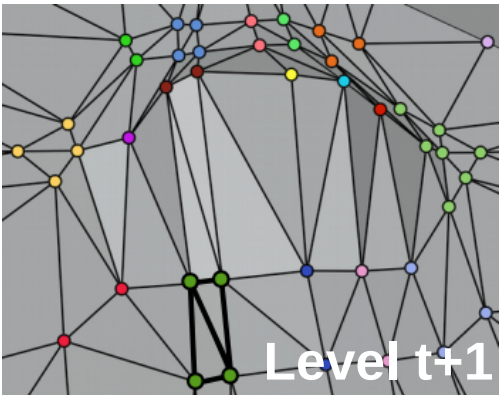
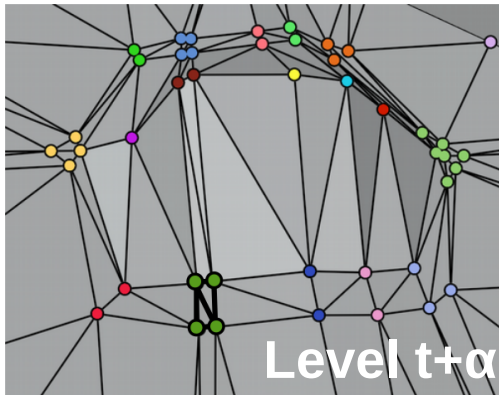
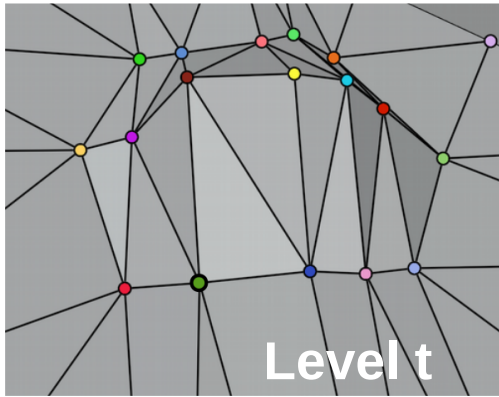
- Uniform transition

- + Hardware tessellation

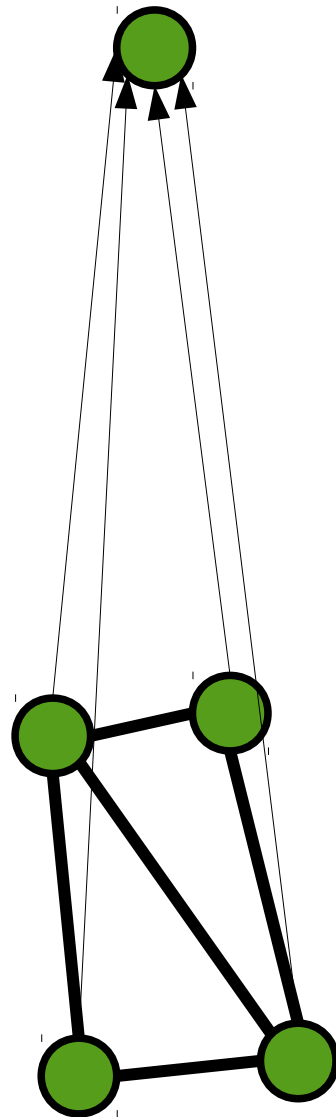
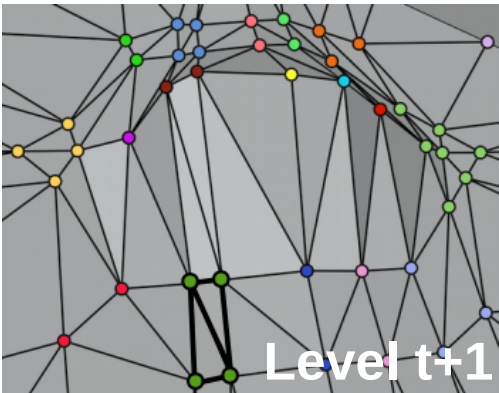
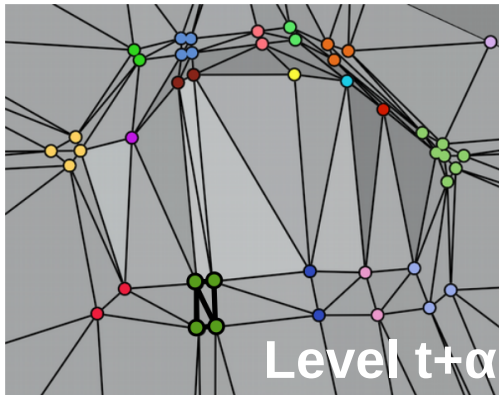
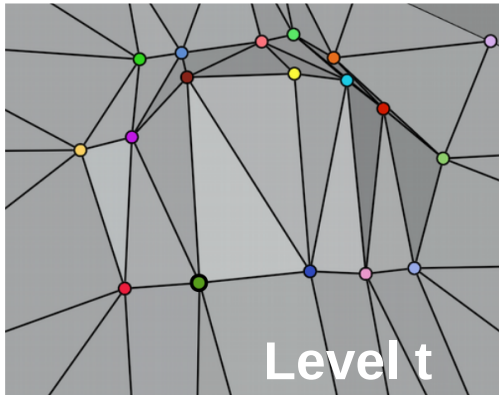
## Contributions

- Relax tessellation constraint
- LOD Generation

# Continuous LOD



# Continuous LOD



## Hierarchical representation

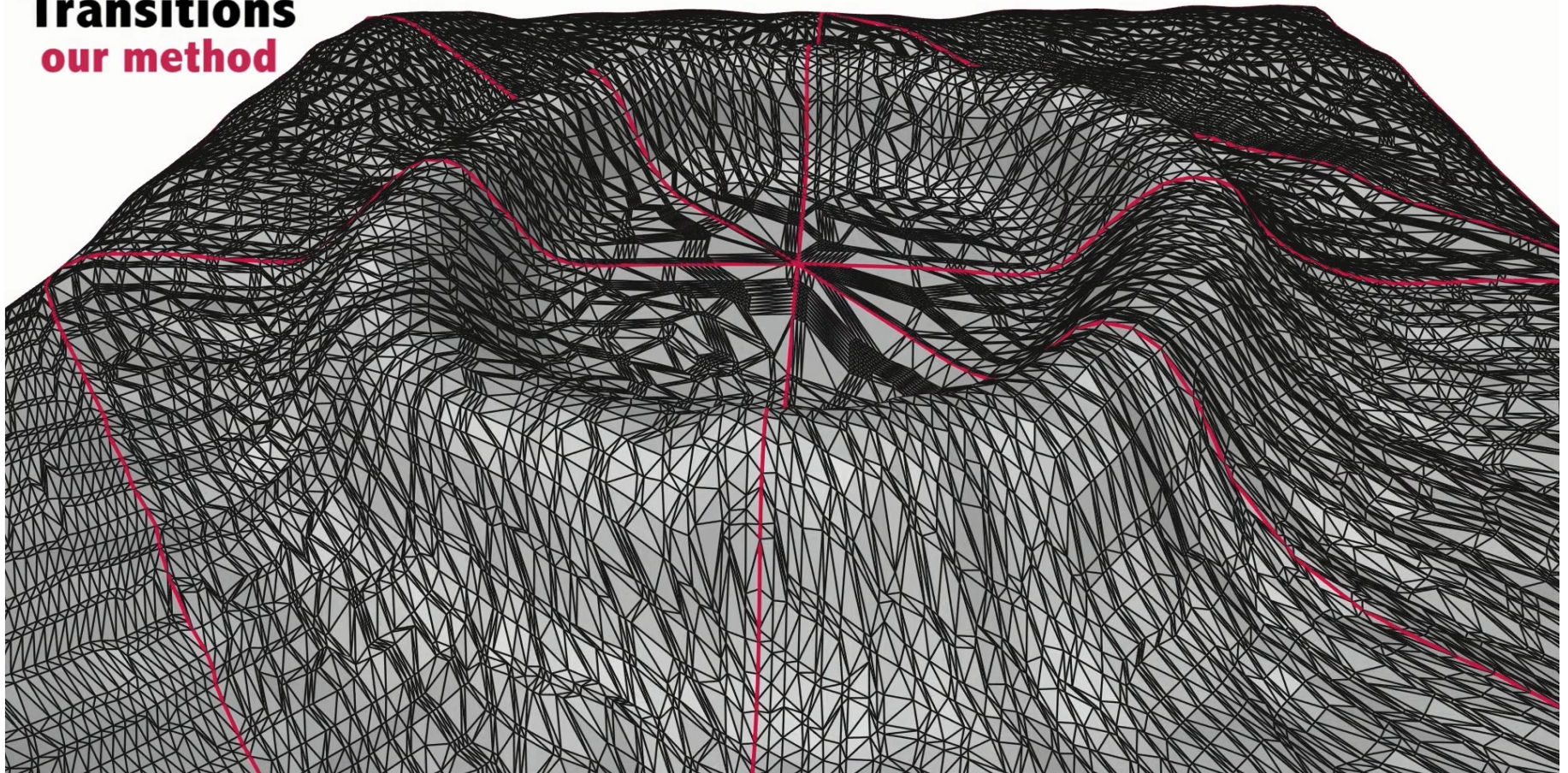
- Each Vertex keeps a pointer to its parent

## Transition

- Linear interpolation between level  $l$  and  $l + 1$

# Continuous LOD

**Transitions**  
our method

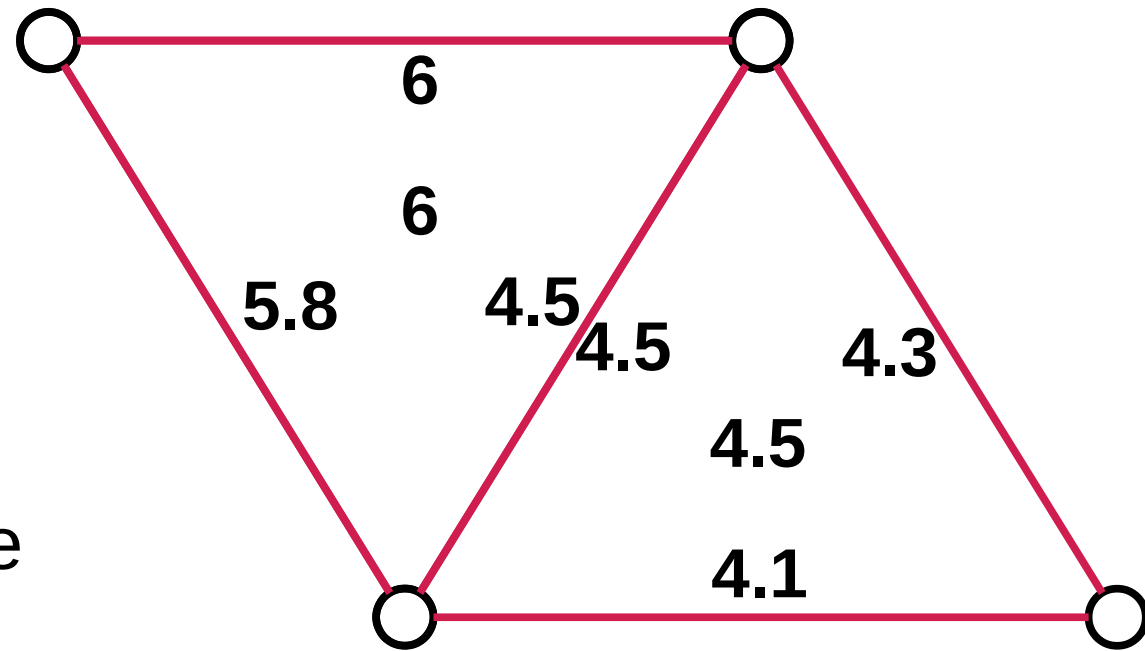




# Spatial Continuity

## For each triangle patch :

- One level for the face
- One level per edge

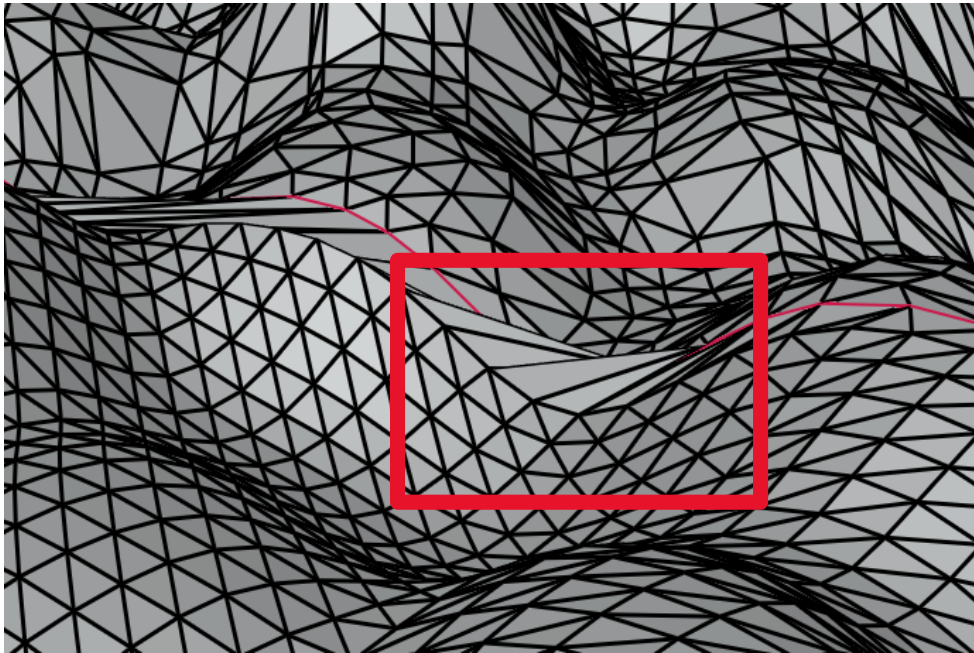


## Avoid cracks :

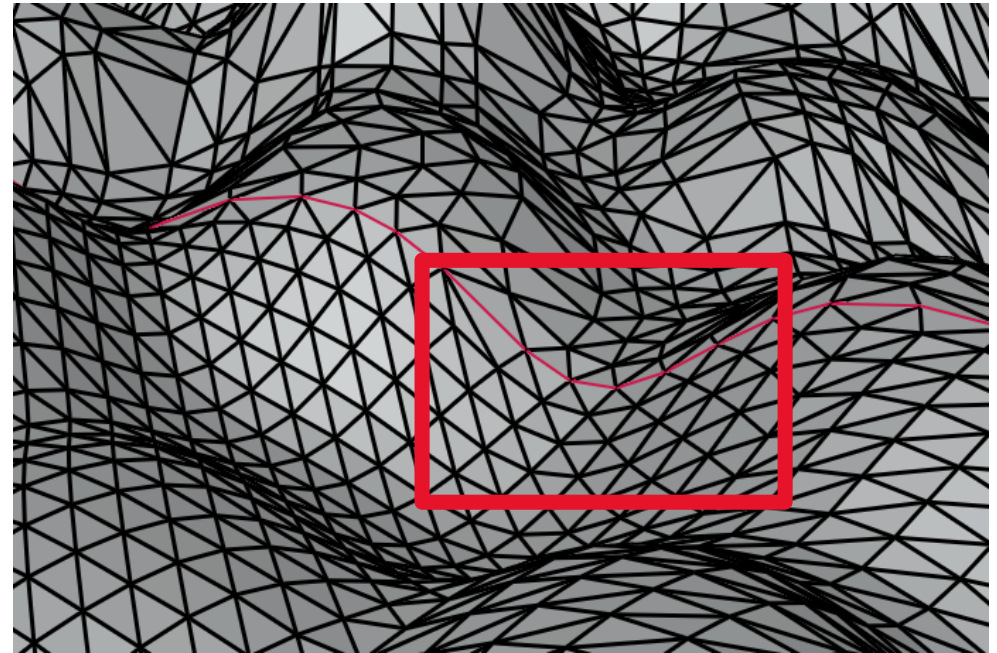
- Same level for the edge of adjacent patches

# Adaptive LOD

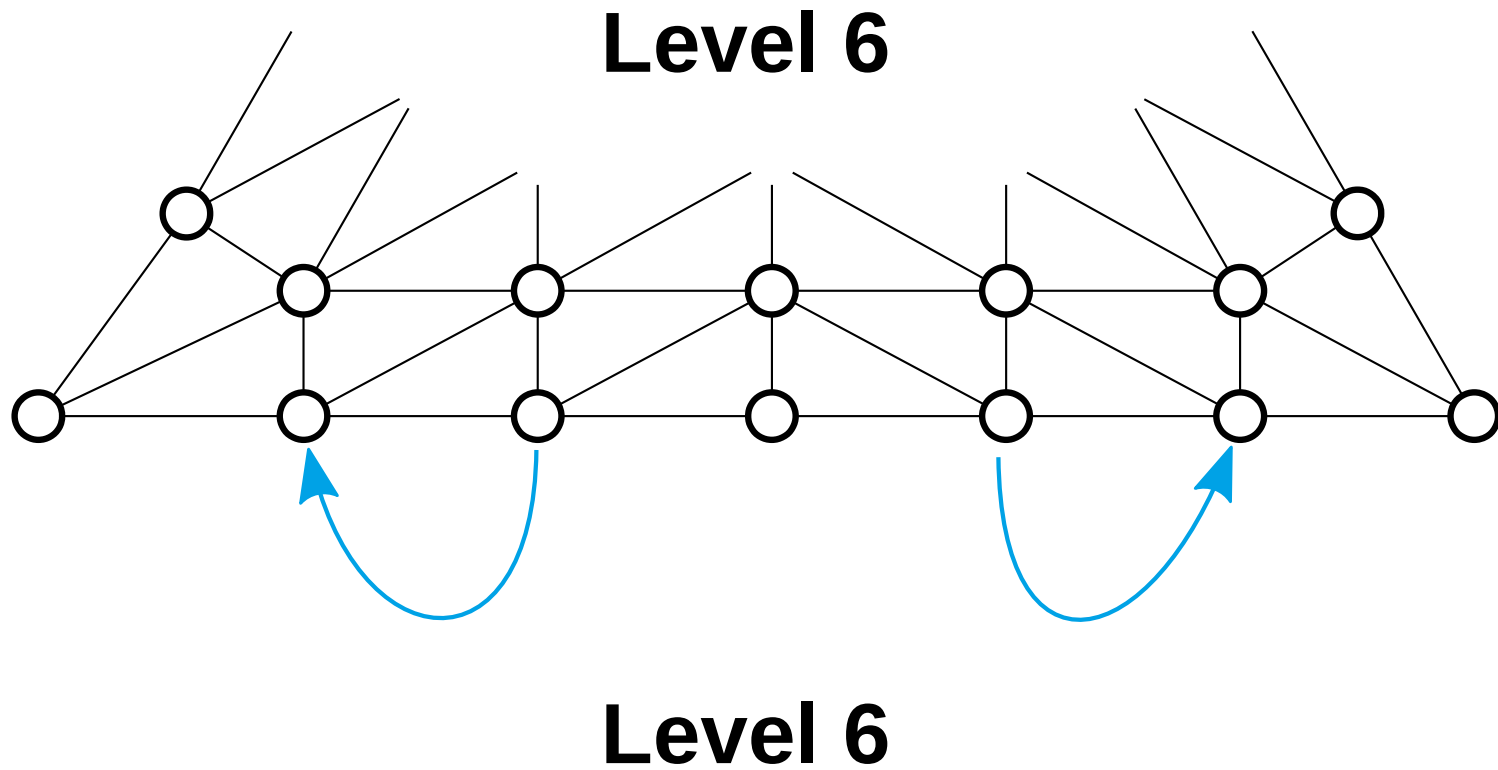
**Hardware Tessellation Stitching**



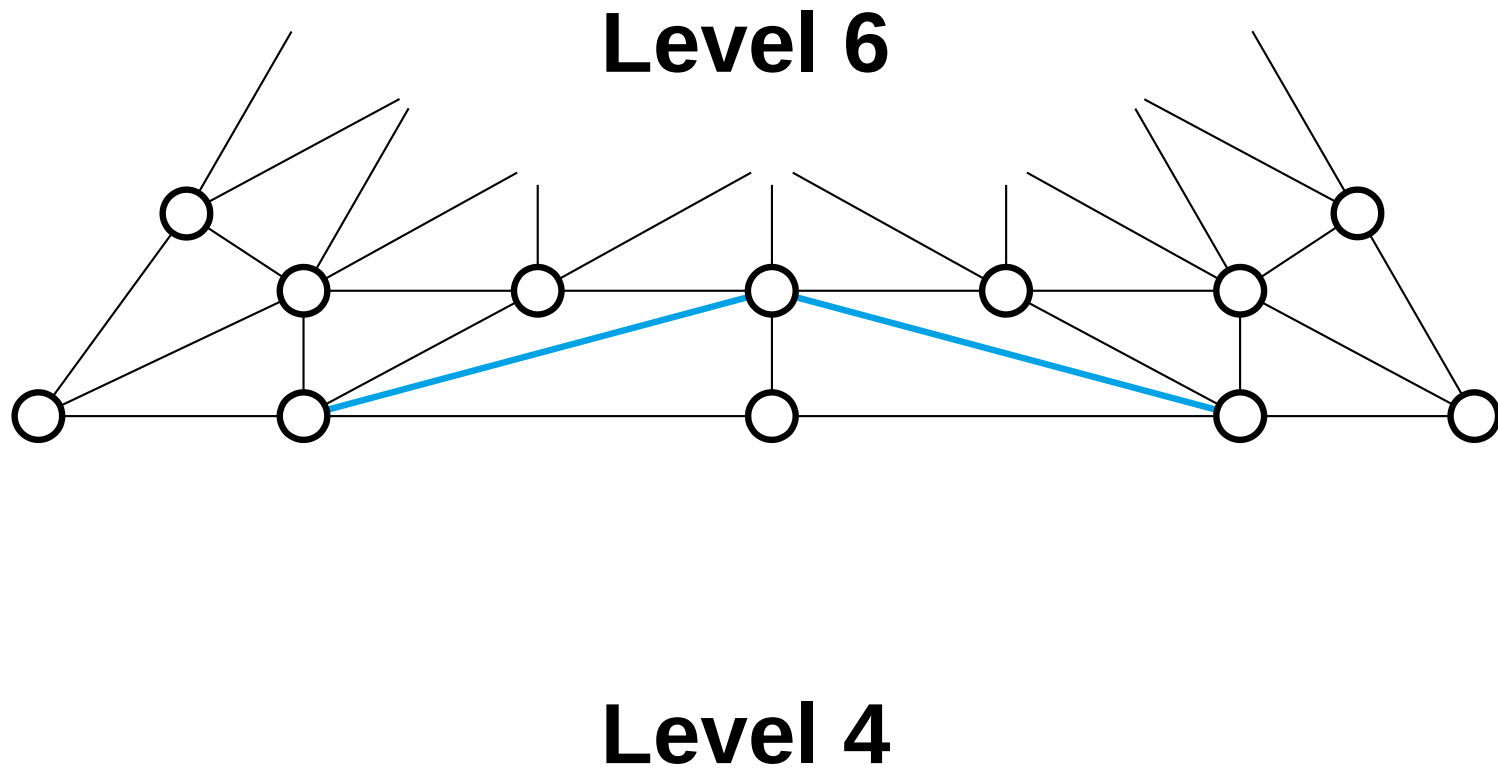
**Wanted stitching**



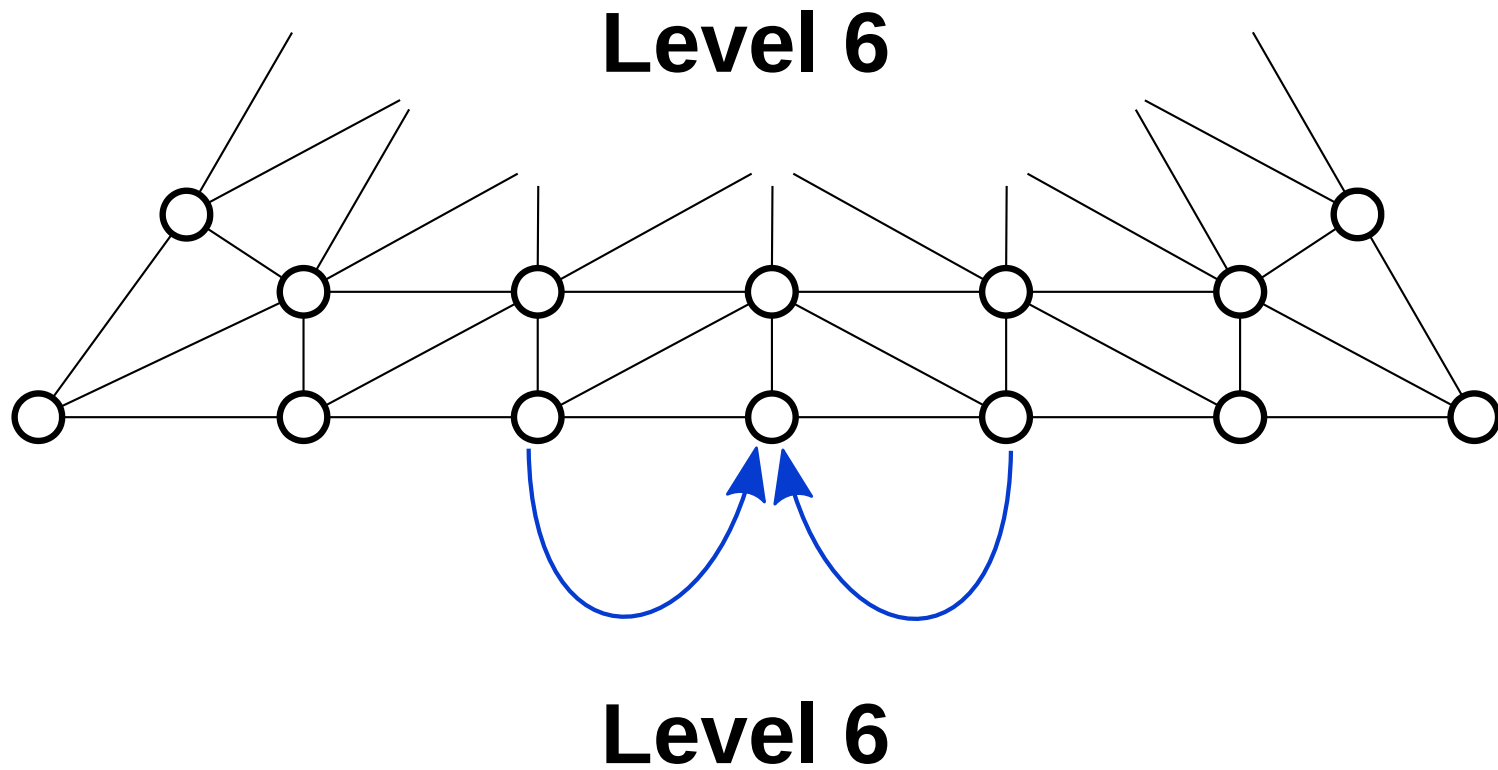
# Adaptive LOD



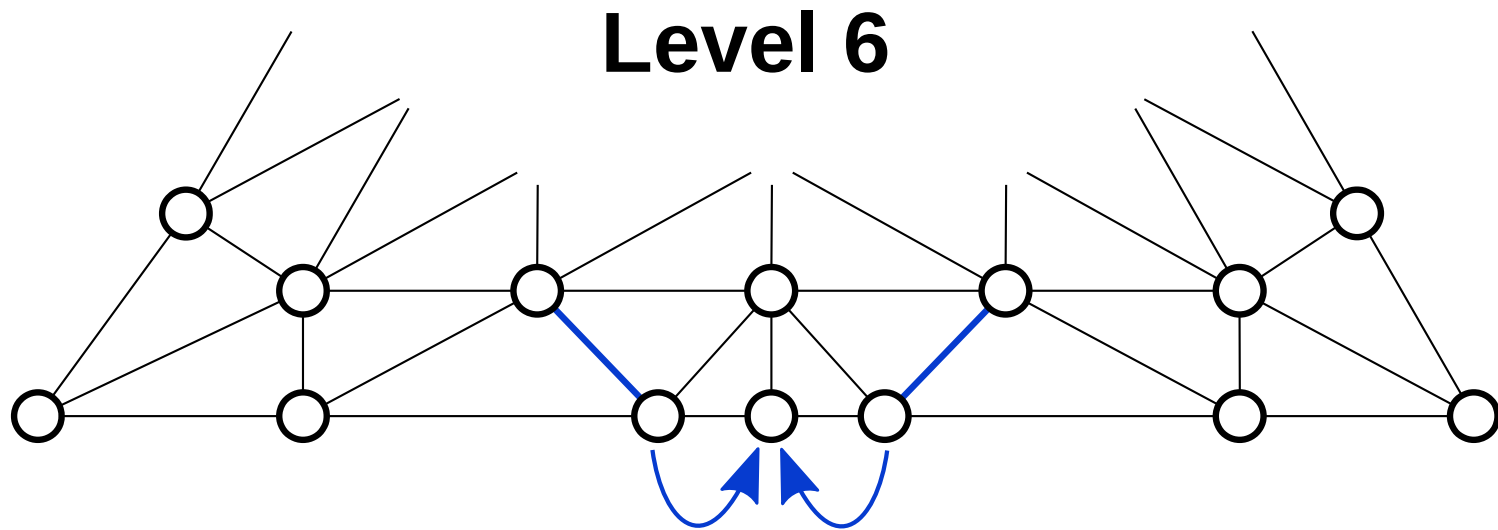
# Adaptive LOD



# Adaptive LOD

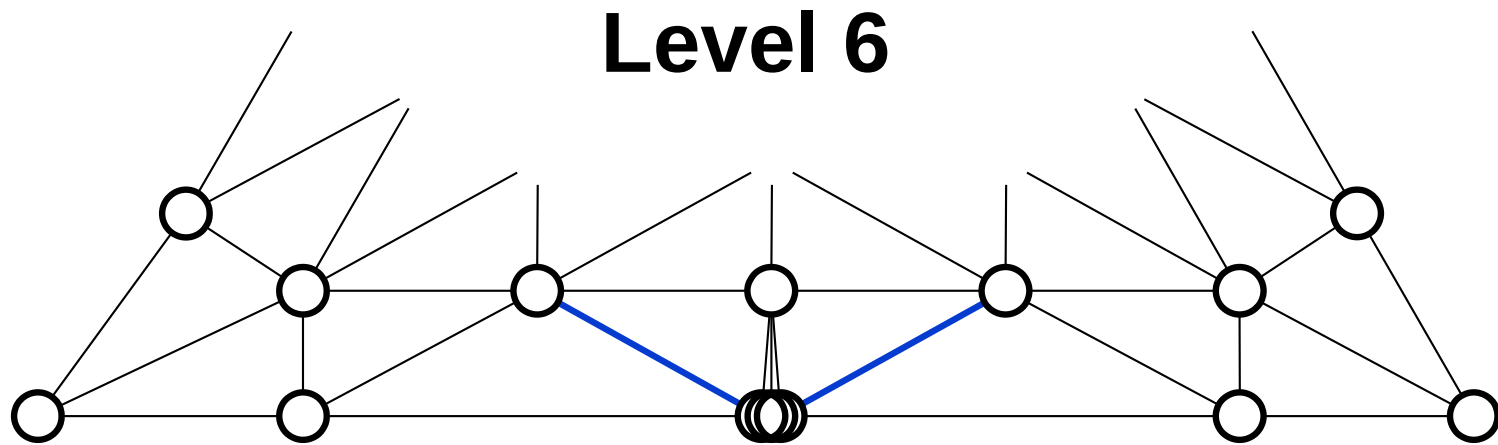


# Adaptive LOD



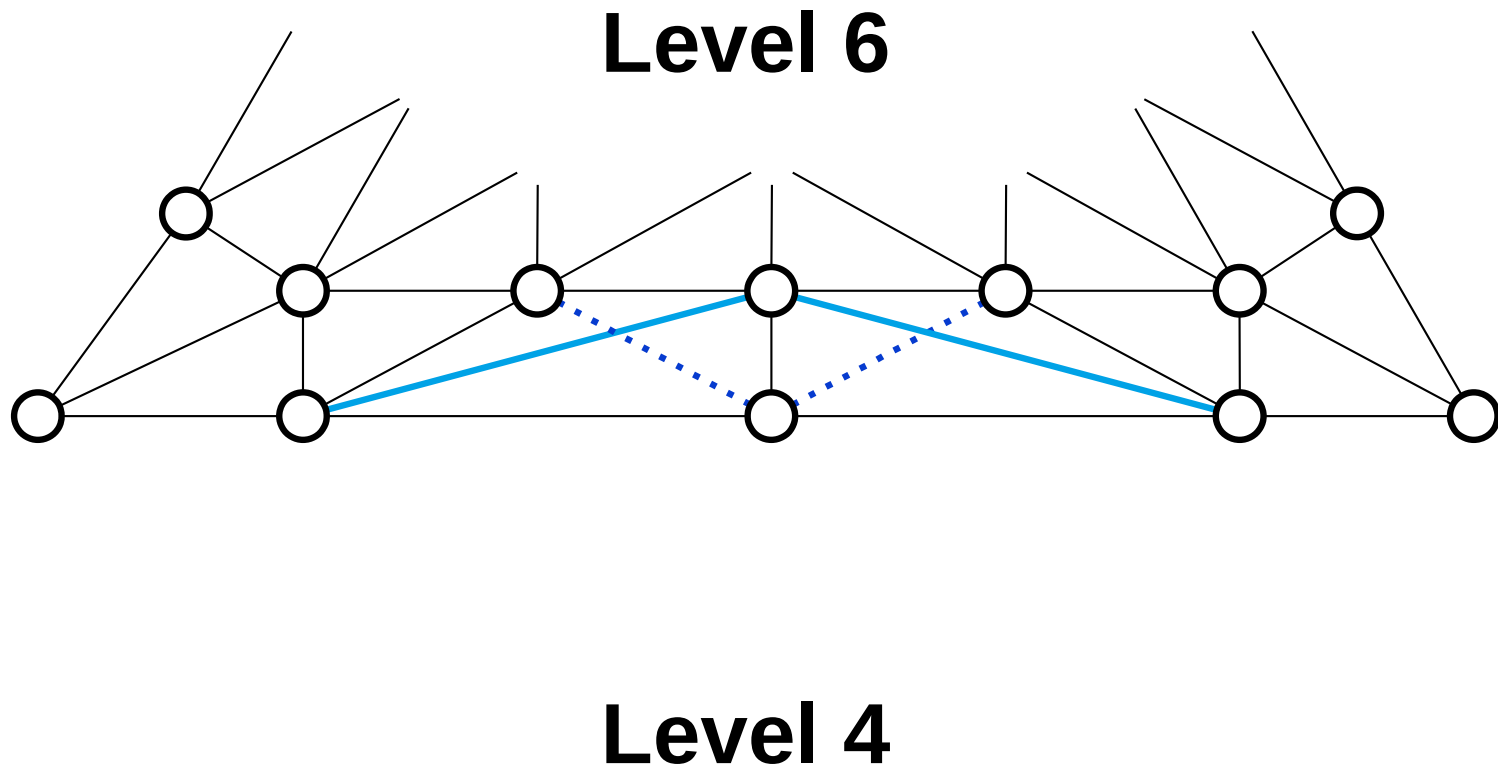
**Level 4.9**

# Adaptive LOD



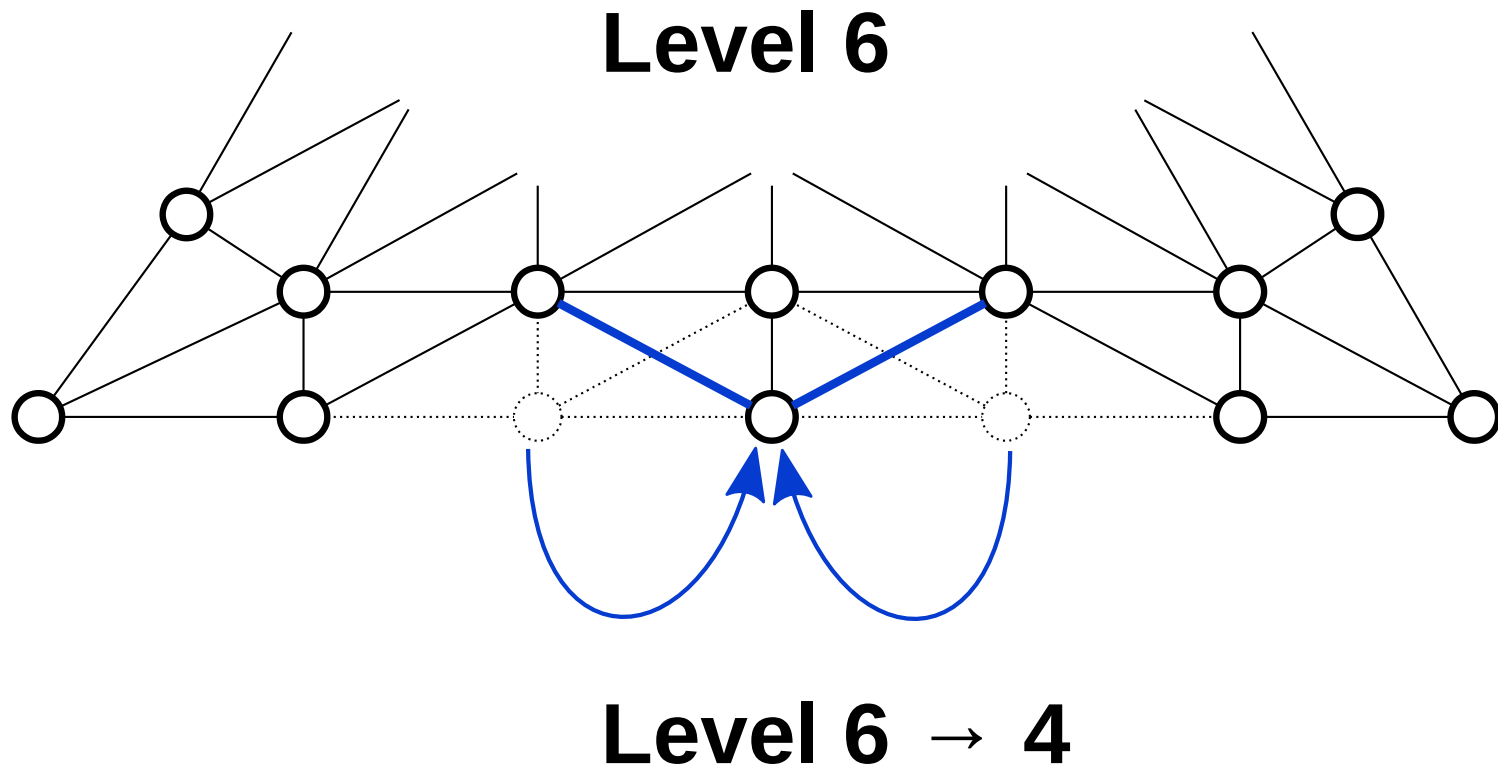
**Level 4.1**

# Adaptive LOD

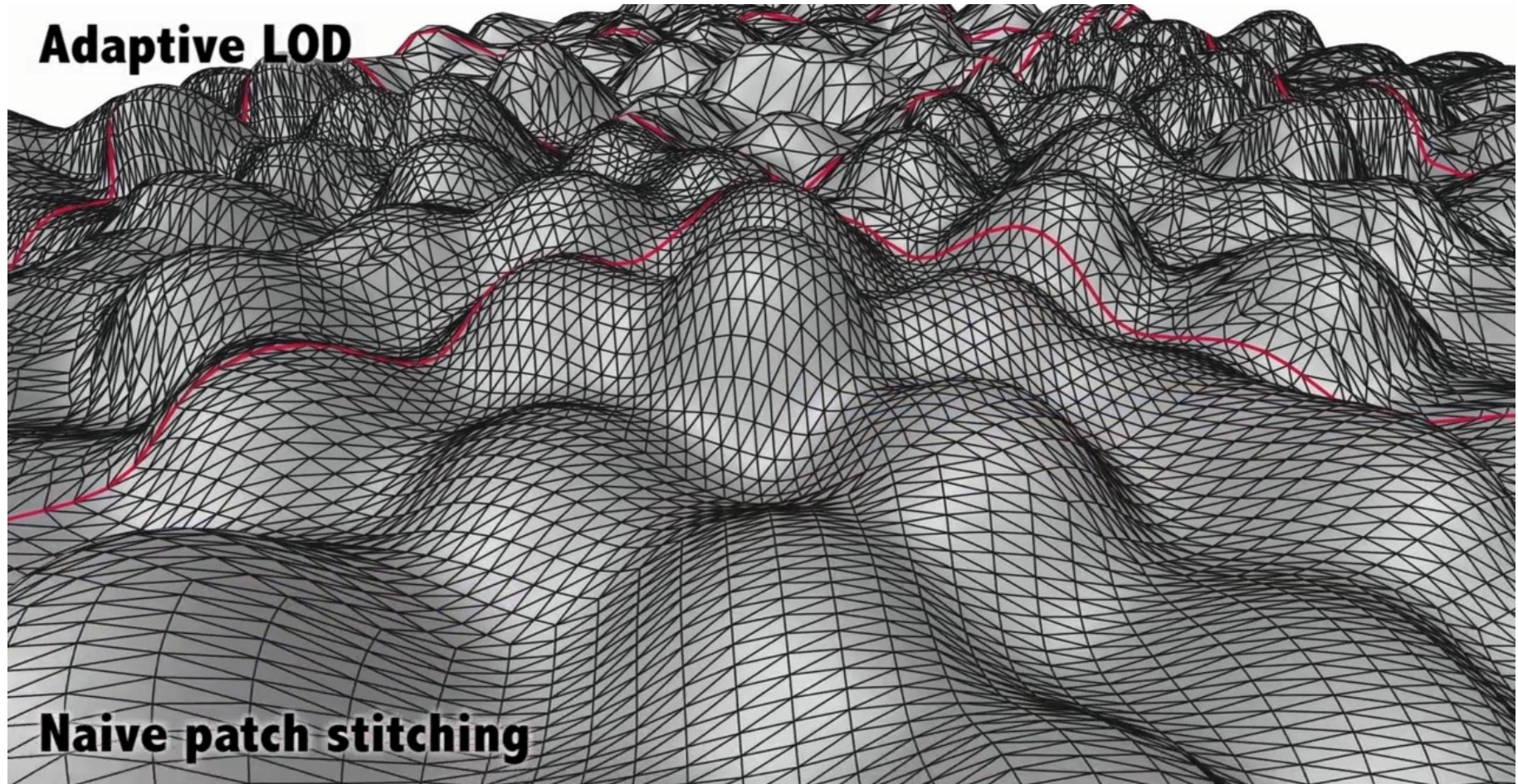




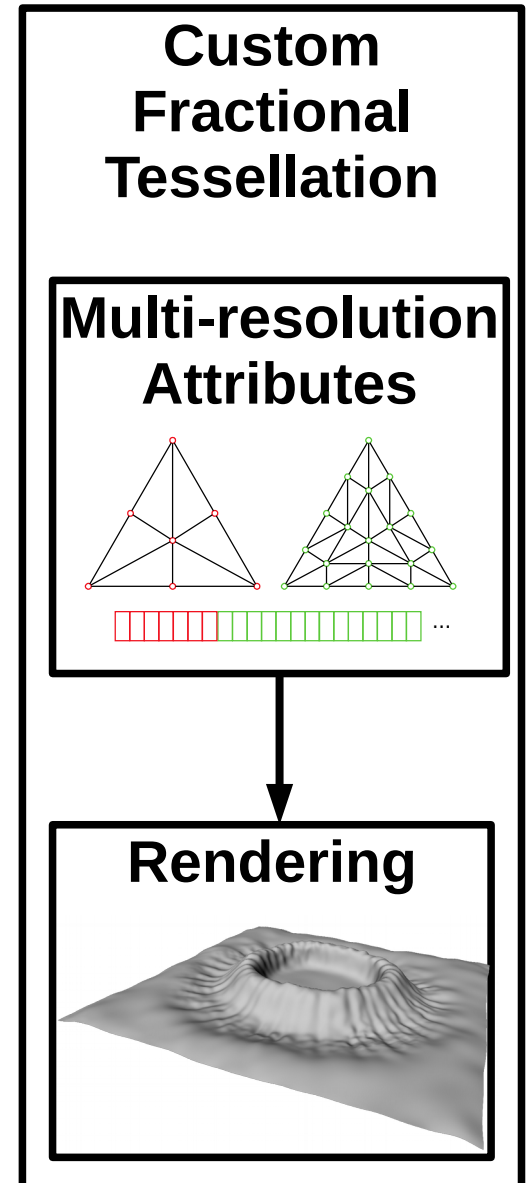
# Adaptive LOD



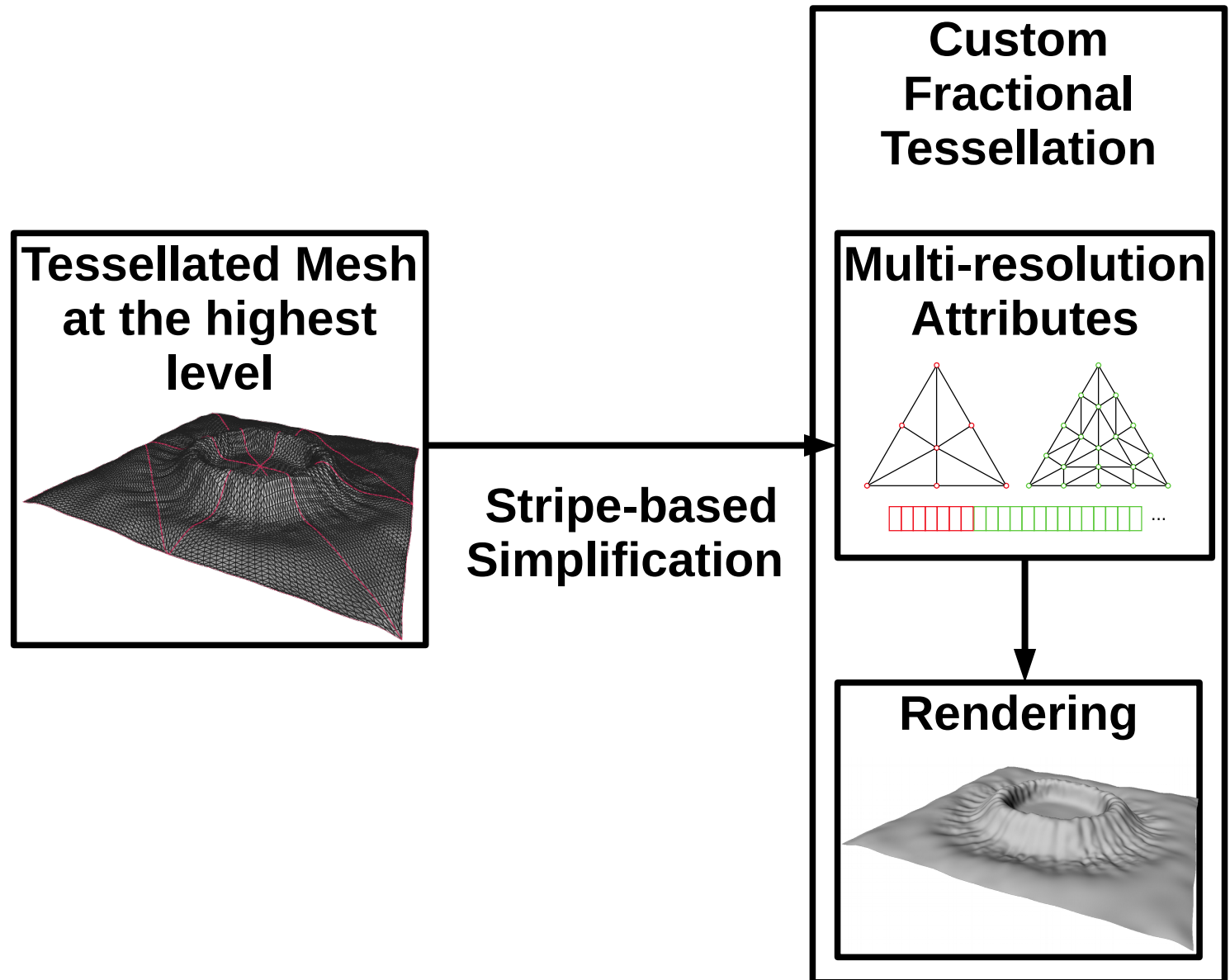
# Adaptive LOD



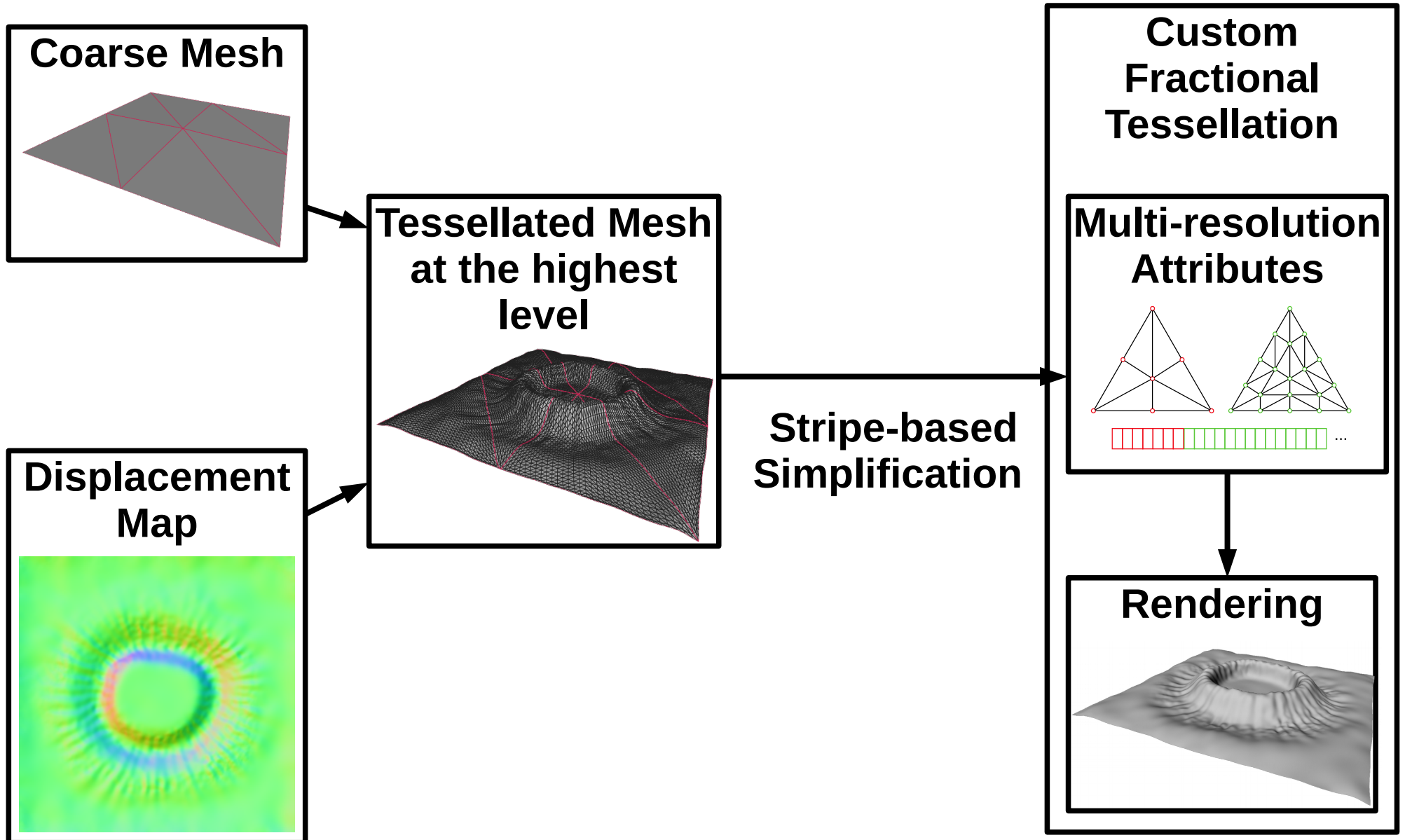
# Global Overview



# Global Overview



# Global Overview



# LOD Generation



## Approach #1

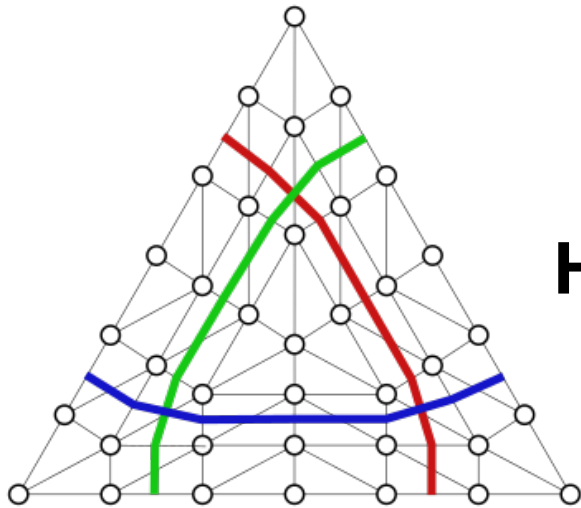
- Optimise each level independently
- Find transition between levels

## Approach #2

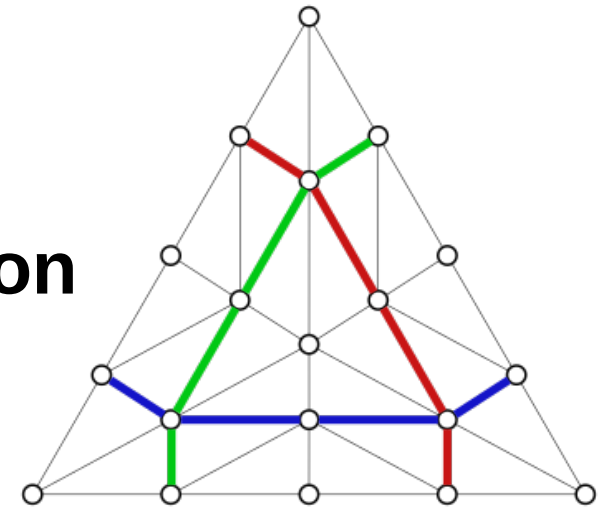
- Start from highest level
- Decimate the mesh while matching successive patterns

**NP  
hard**

# LOD Generation



Hardware Tessellation



## Approach #1

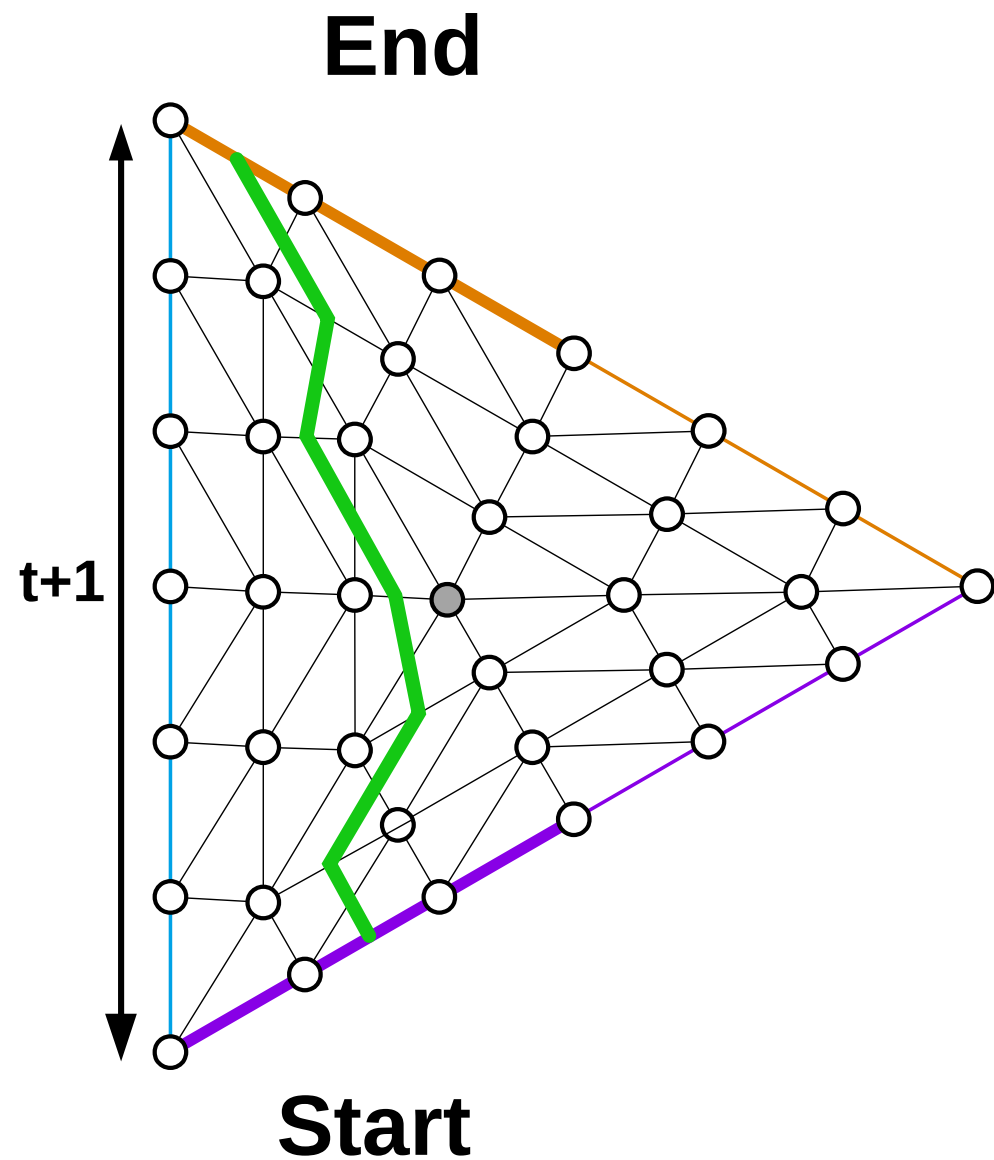
- Optimise each level independently
- Find transition between levels

## Approach #2

- Start from highest level
- Decimate the mesh while matching successive patterns

**NP  
hard**

# Stripe-based mesh simplification

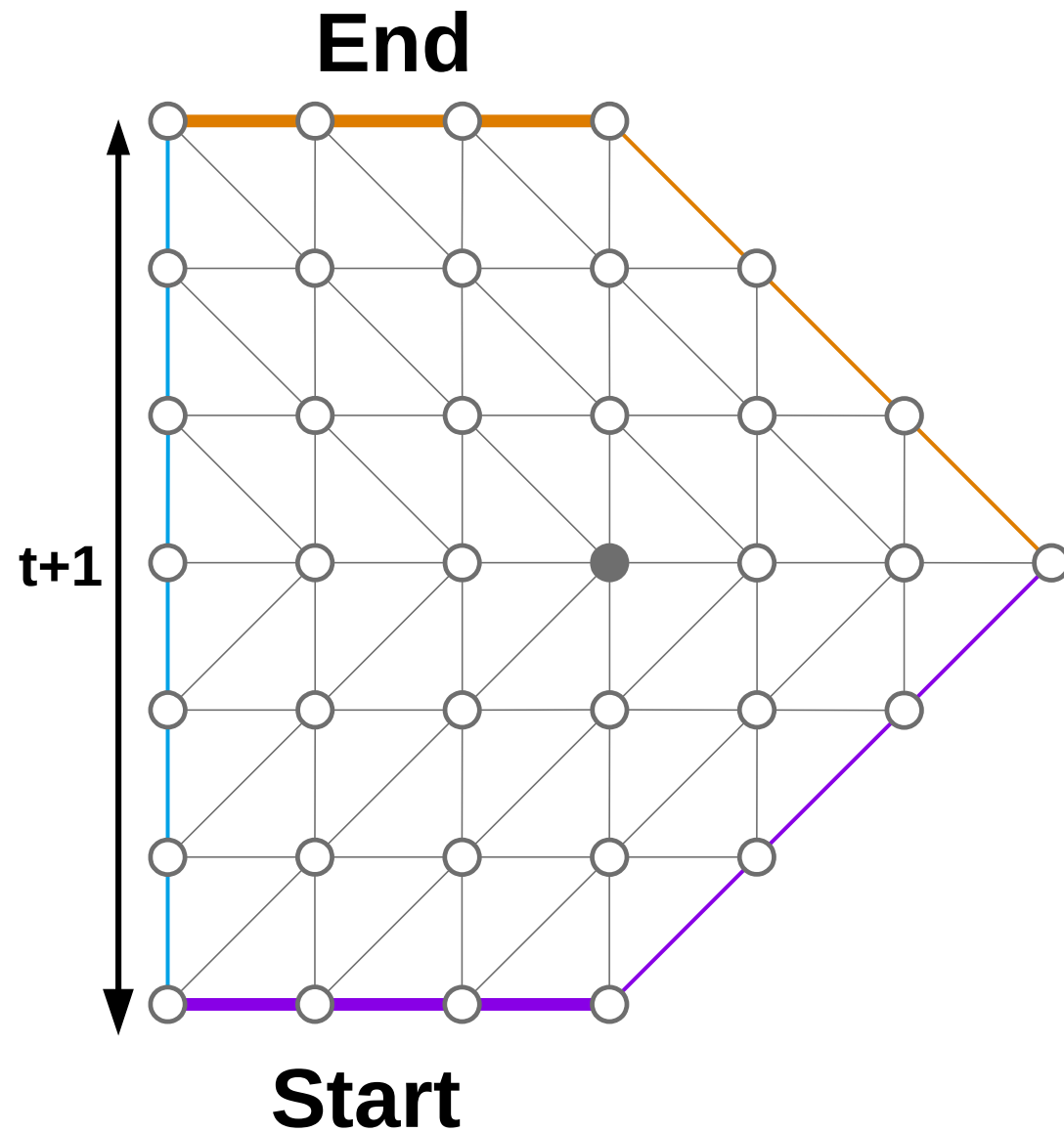


## Extended strips

- **Start:** half patch border
- **End:** opposite half
- **Length:** tessellation level + 1



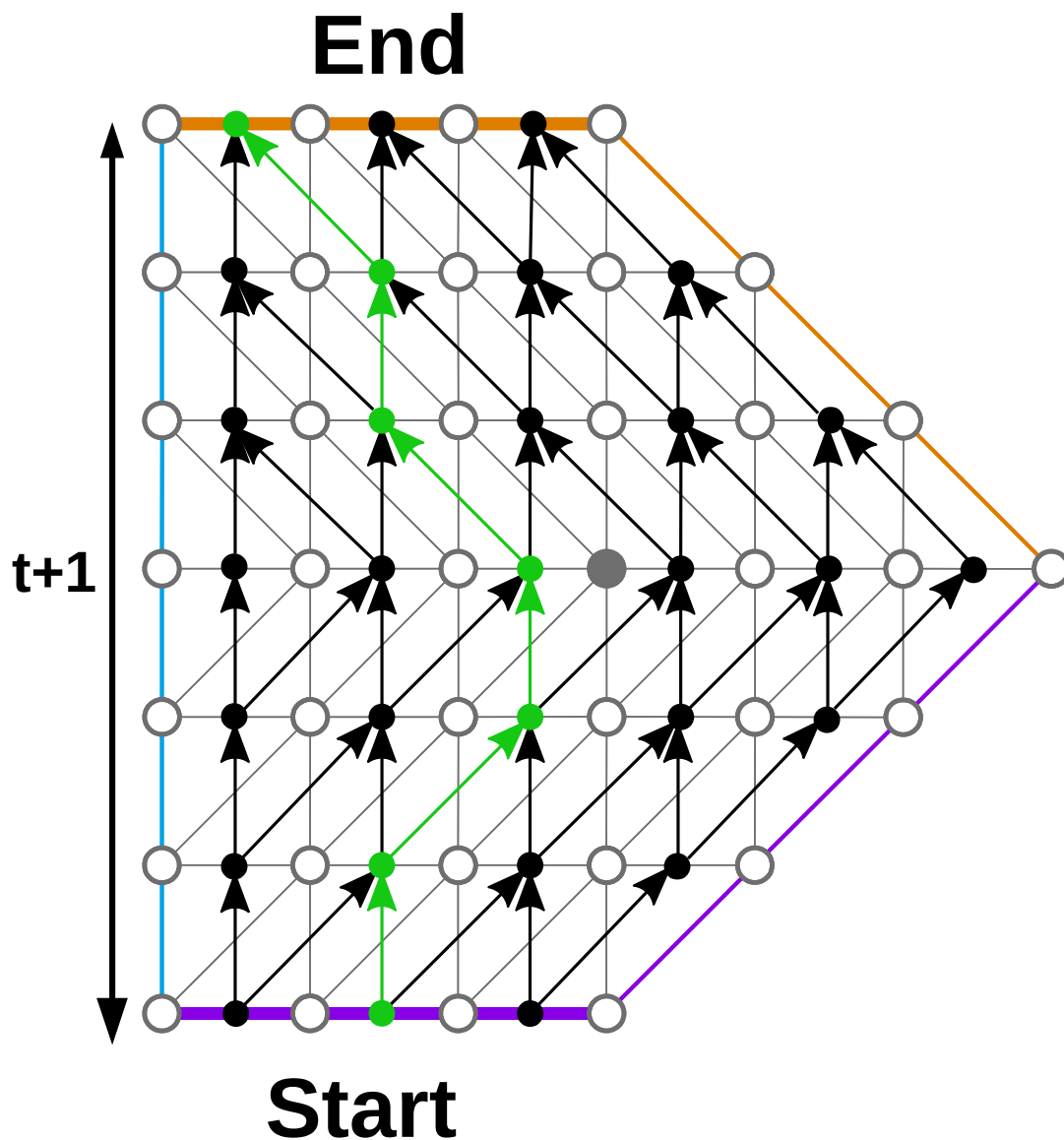
# Stripe-based mesh simplification



## Extended strips

- **Start:** half patch border
- **End:** opposite half
- **Length:** tessellation level + 1

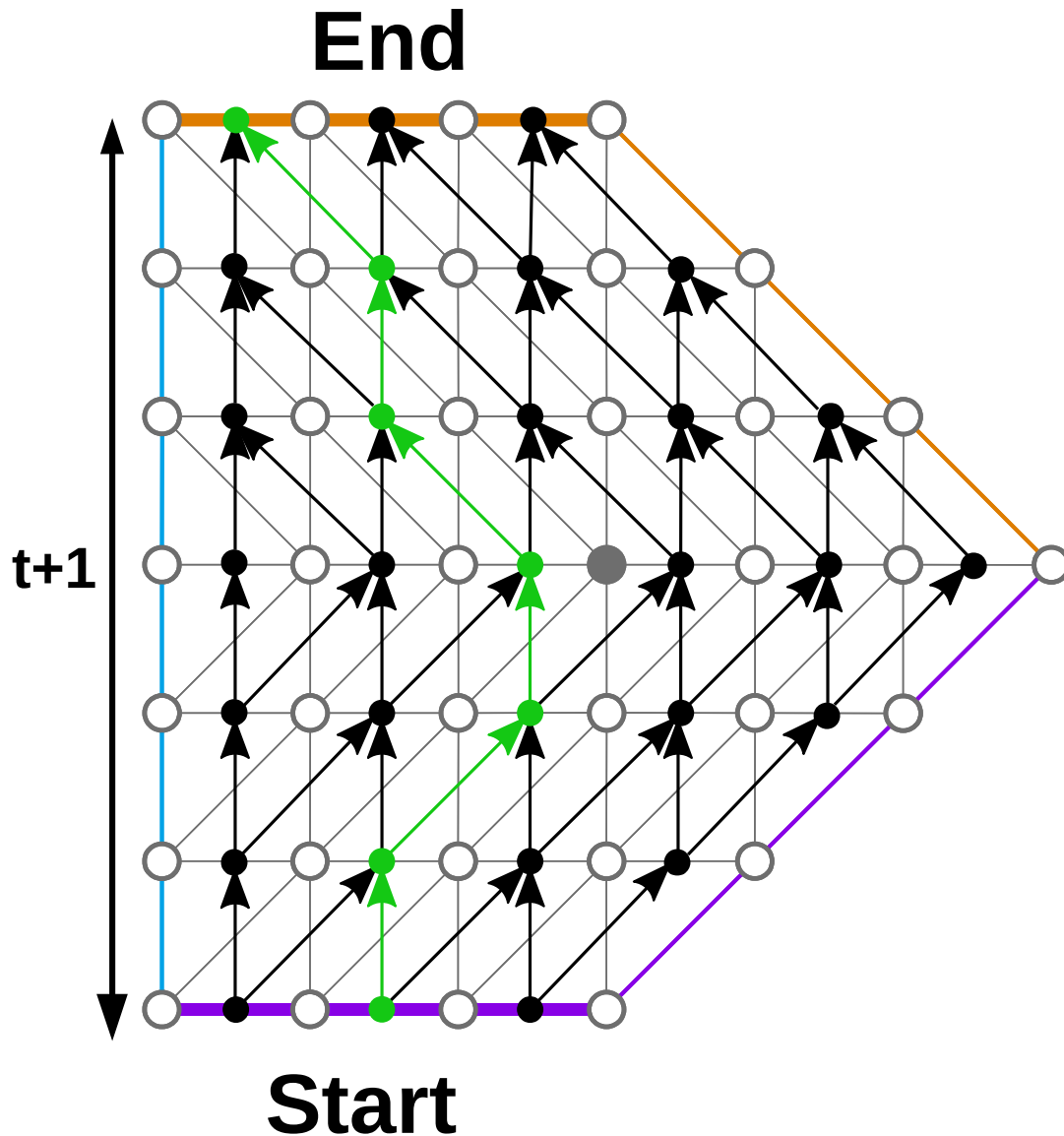
# Stripe-based mesh simplification



## Extended strips

- **Start:** half patch border
- **End:** opposite half
- **Length:** tessellation level + 1

# Stripe-based mesh simplification



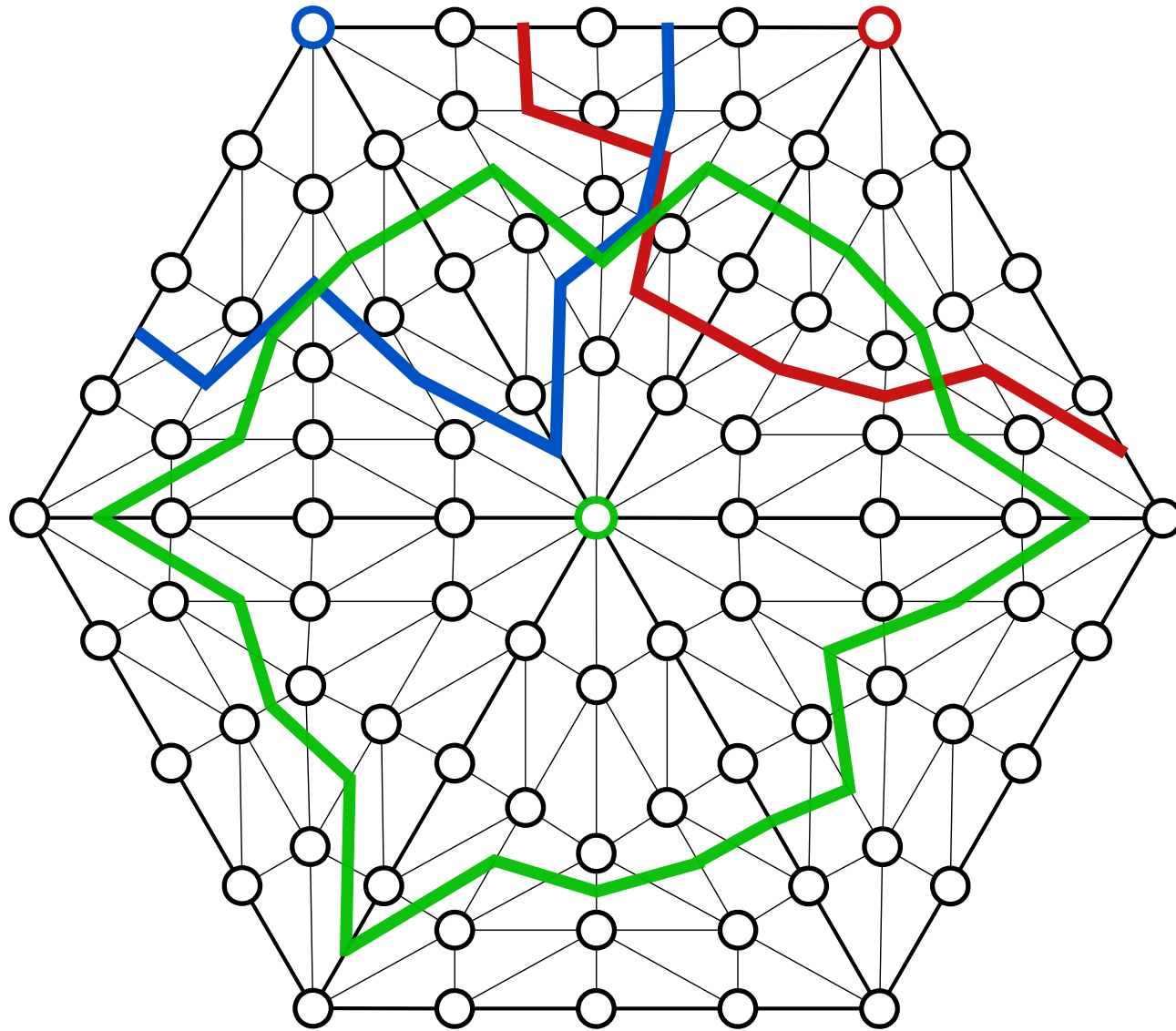
## Extended strips

- **Start:** half patch border
- **End:** opposite half
- **Length:** tessellation level + 1

## Choose strip

- Shortest paths
- Quadric Error [GH97]

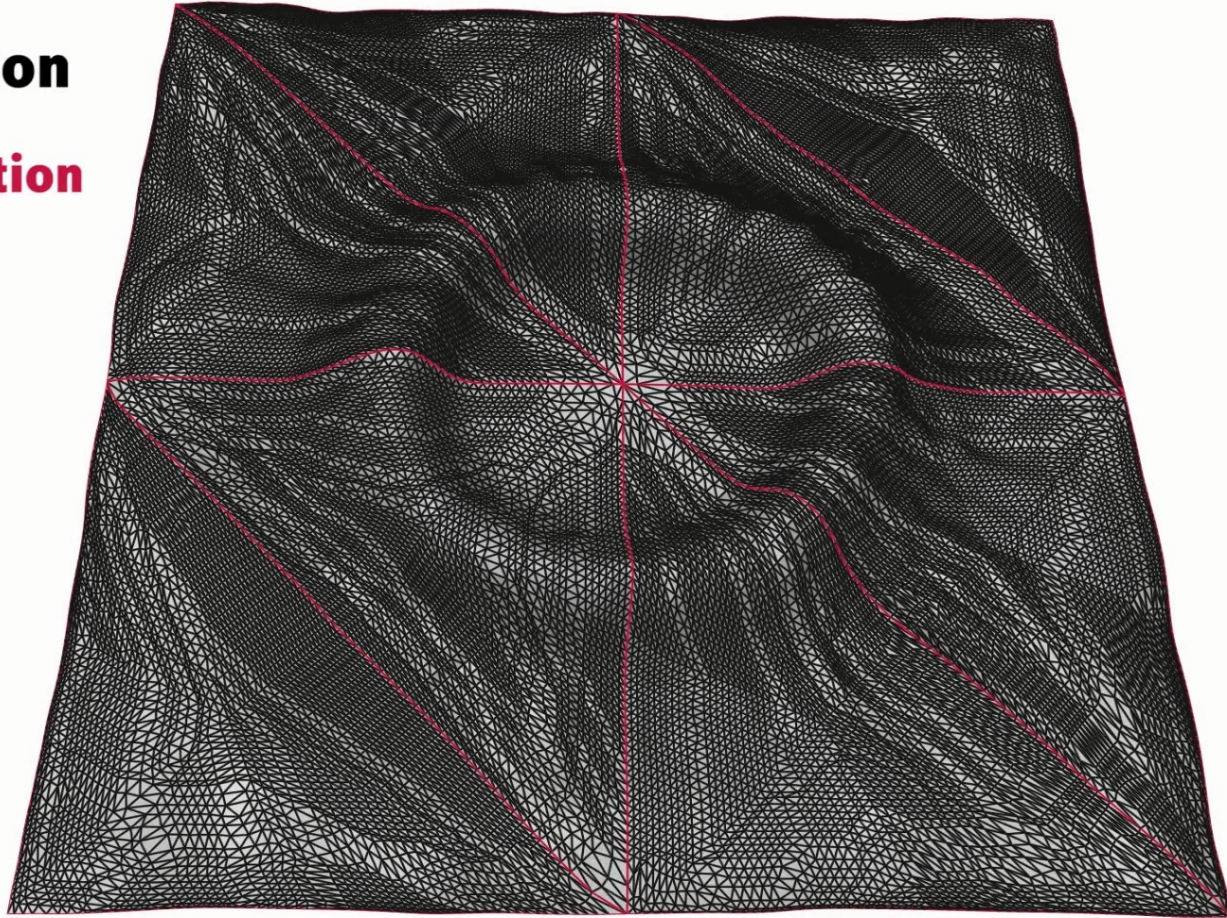
# Stripe-based mesh simplification



**Connect stripes : rings**

# Stripe-based mesh simplification

**LOD generation**  
**strip-collapse**  
**error minimization**



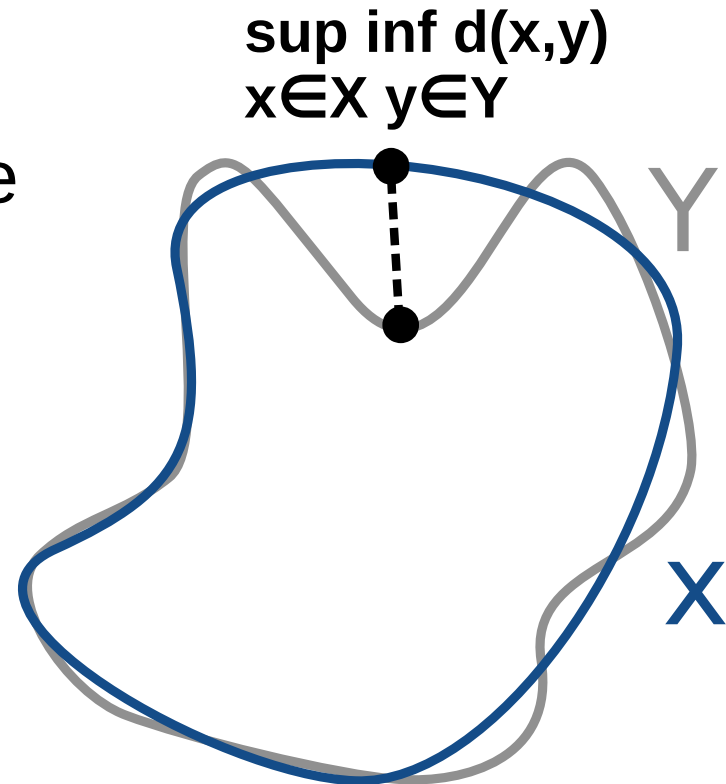
# Rendering Details

## LOD Selection

- Based on Hausdorff distance
- Computed for each level

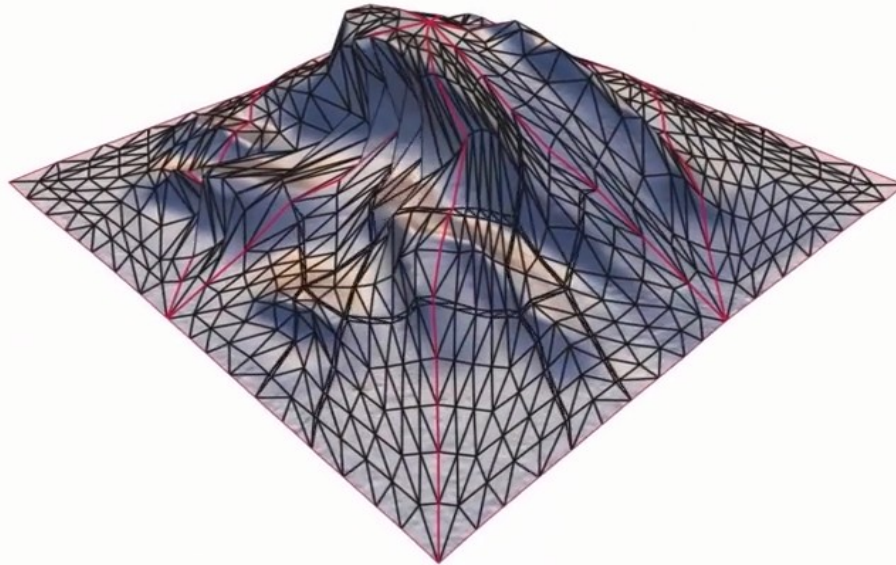
## Power of two levels

- Less memory



# Results

## "Snowdrifts"



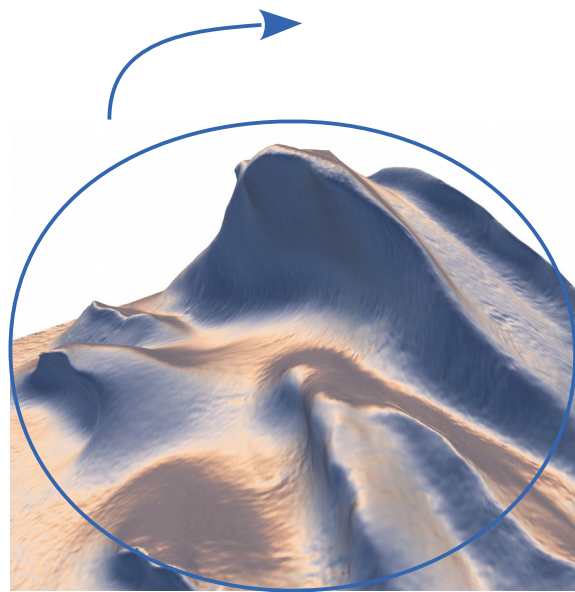
**fractional hw. tess.**

# Results

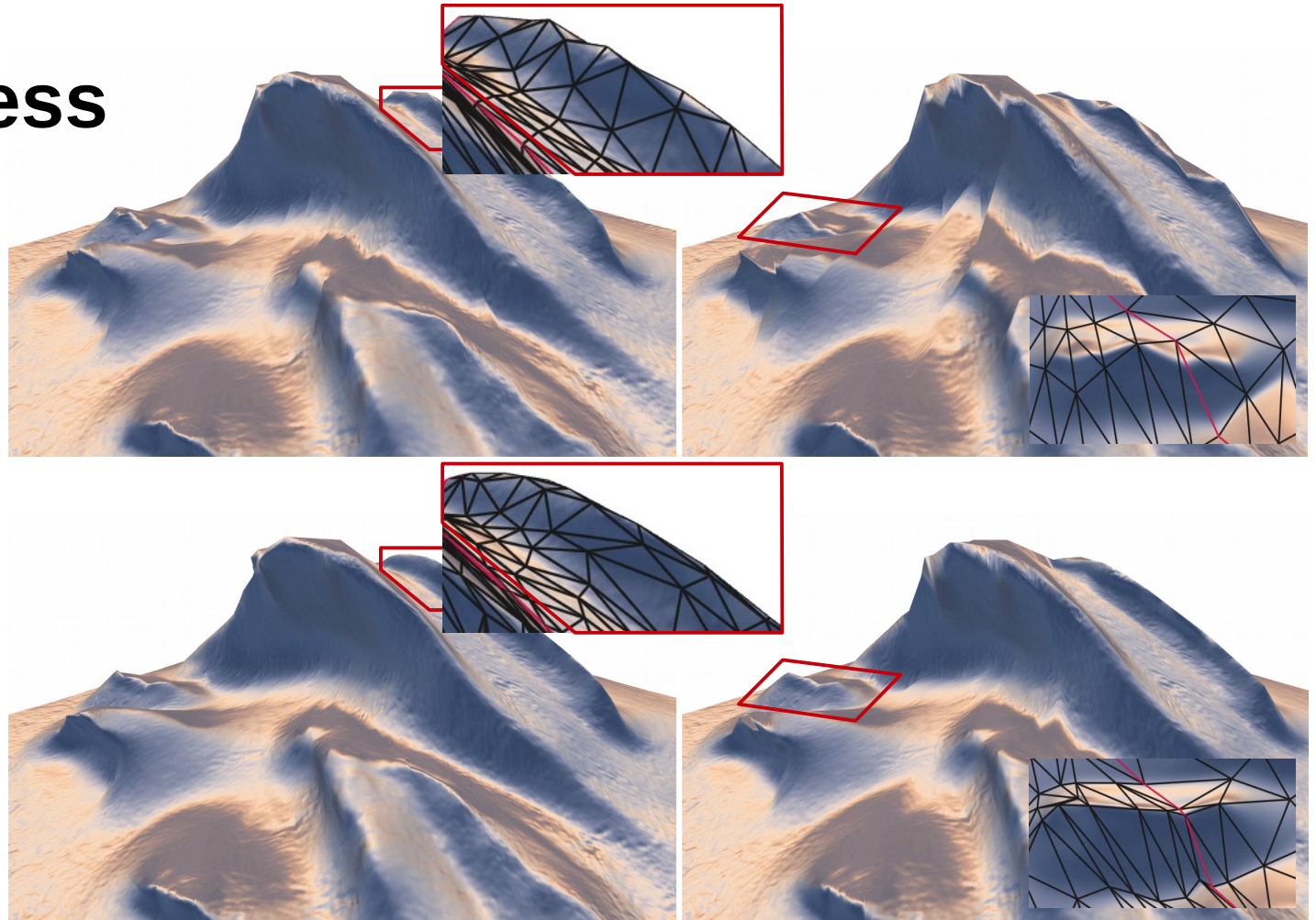
Level 16

Level 8

Uniform hw. tess



Our method





# Results

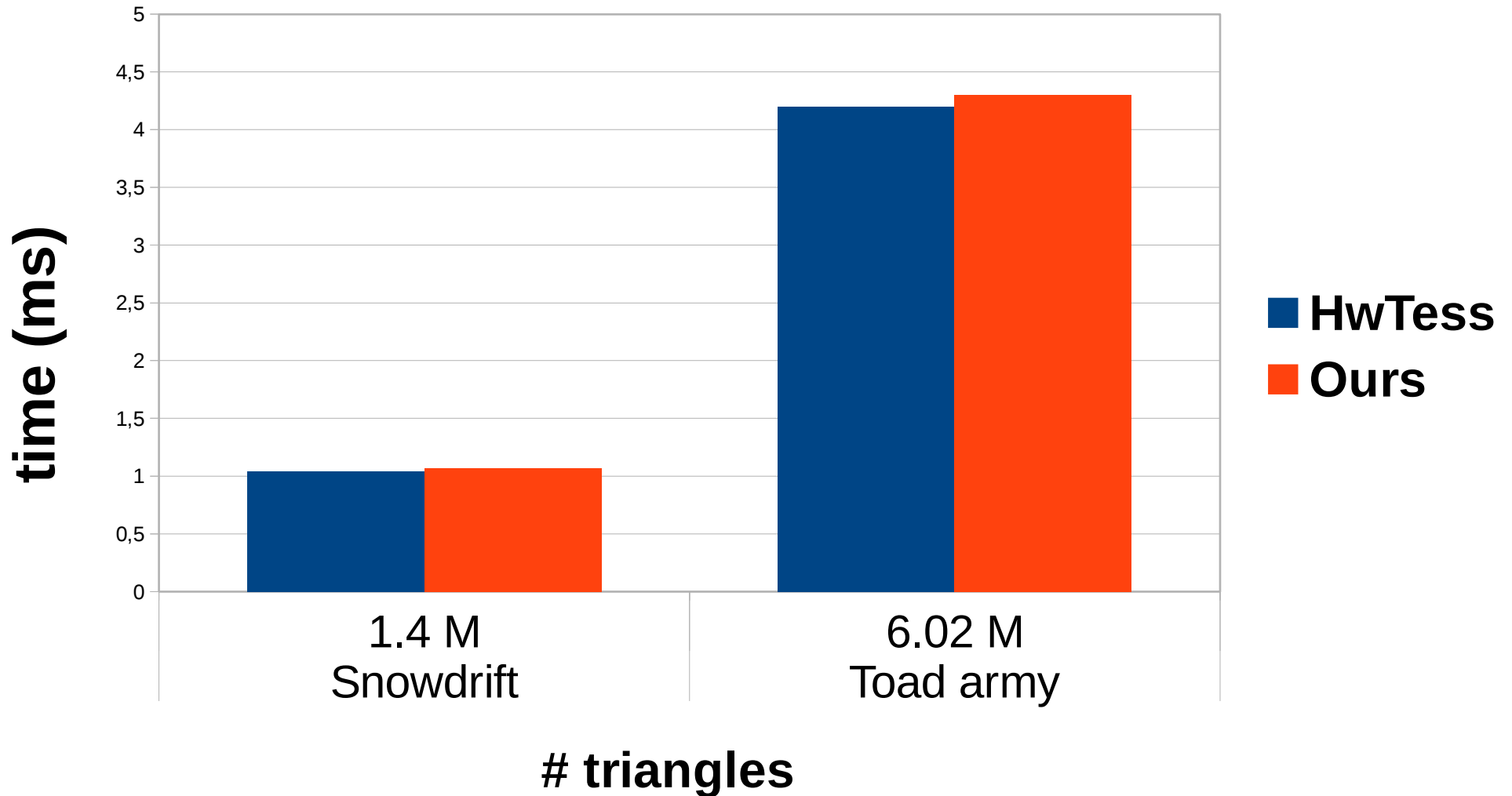
**"Toad"**



**fractional hw. tess.**

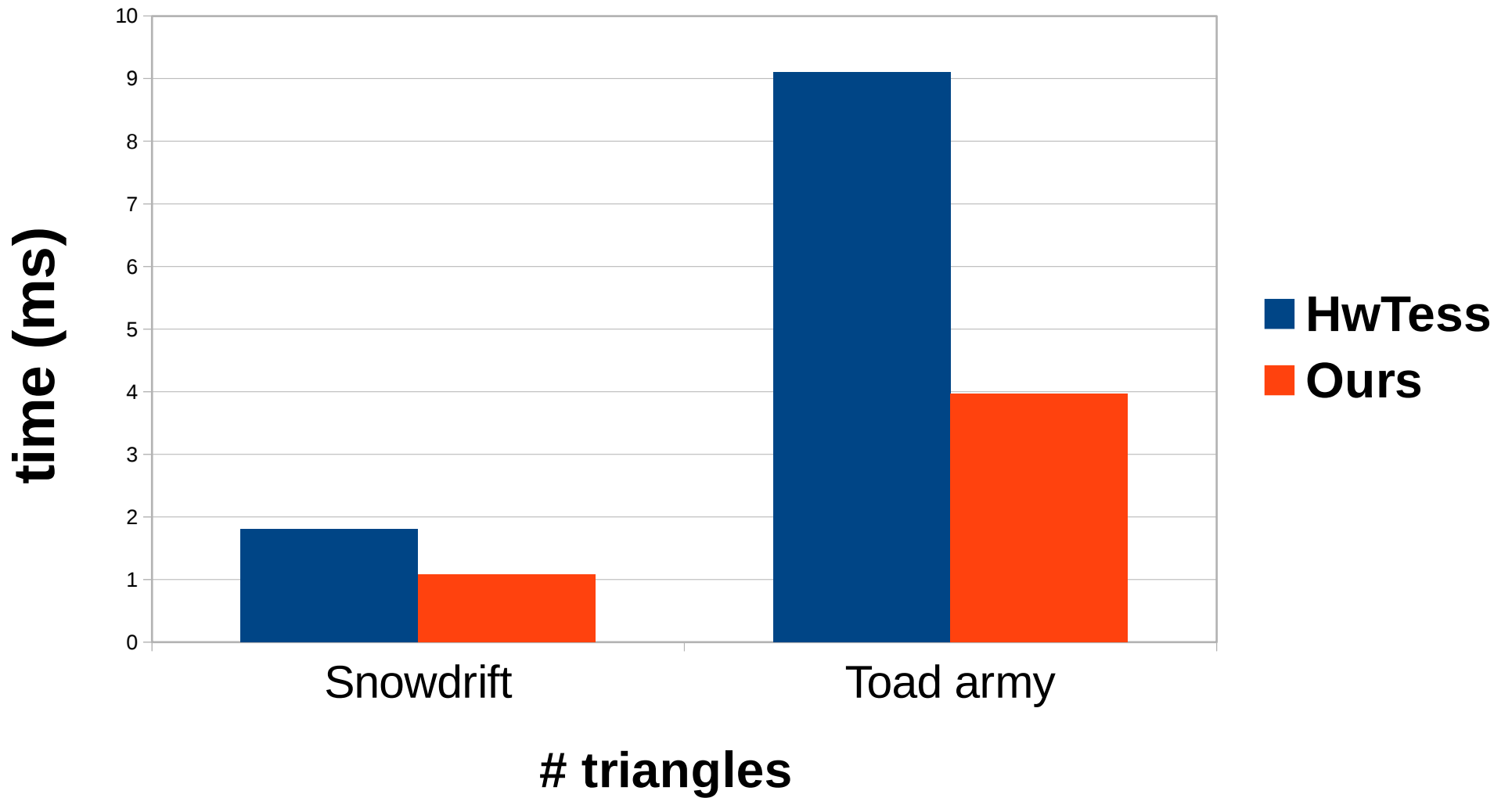
# Performance

## Equal number of triangles



# Performance

## Equal visual quality

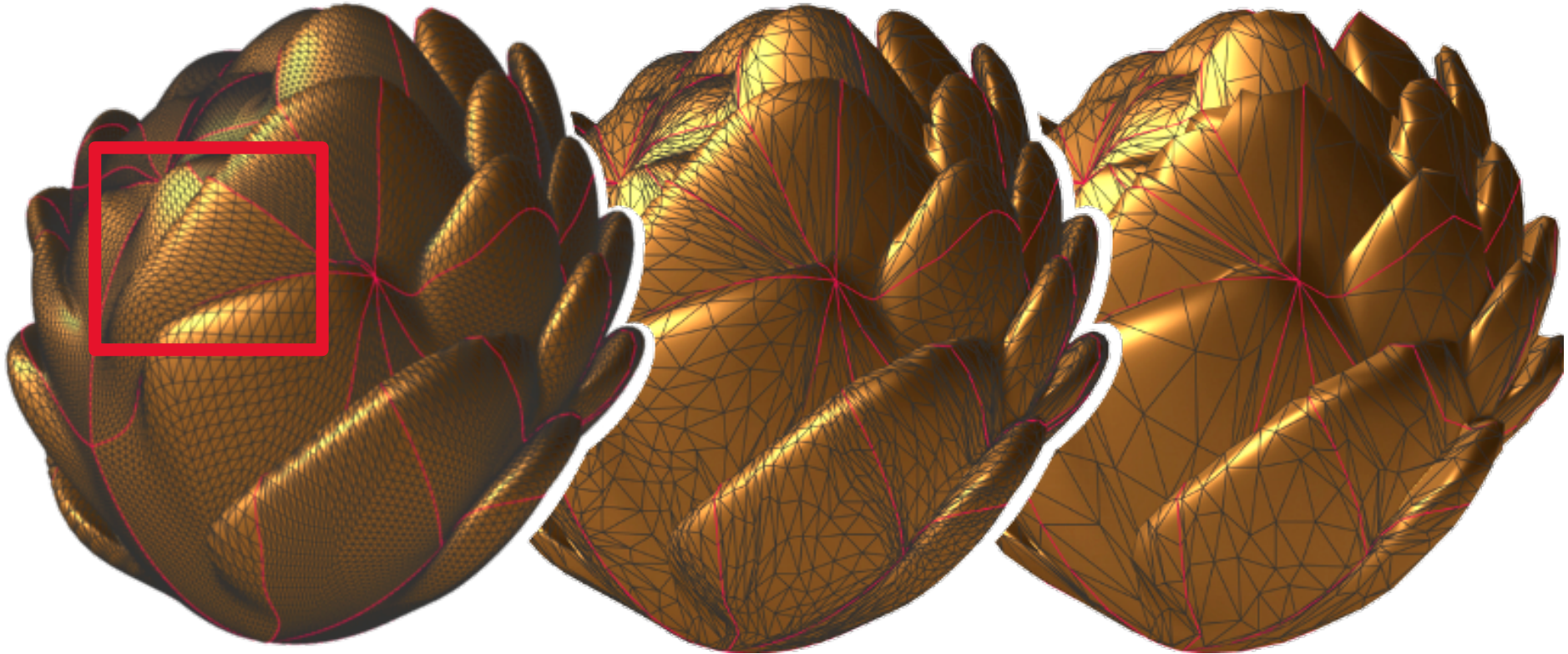


# Future works

Level 64

Level 16

Level 8



## Optimize initial solution

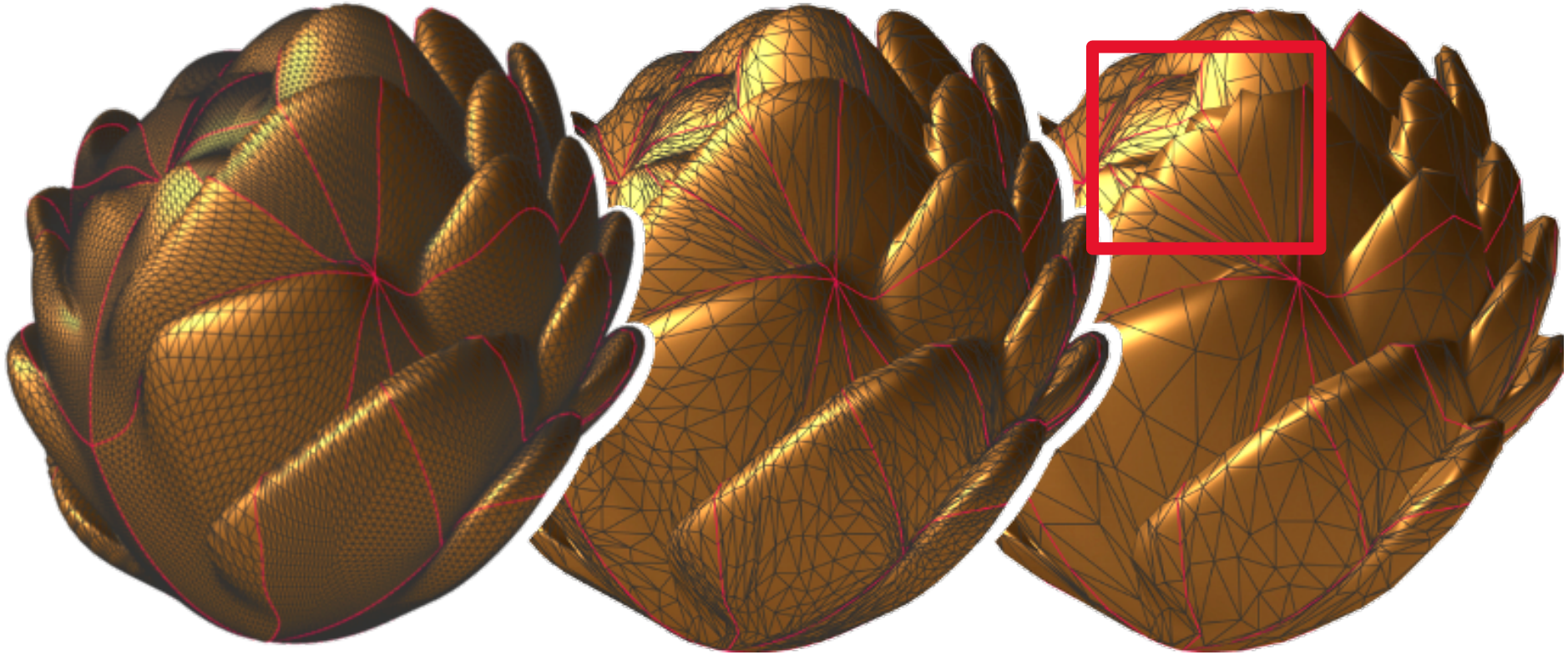
- Surface fitting

# Future works

Level 64

Level 16

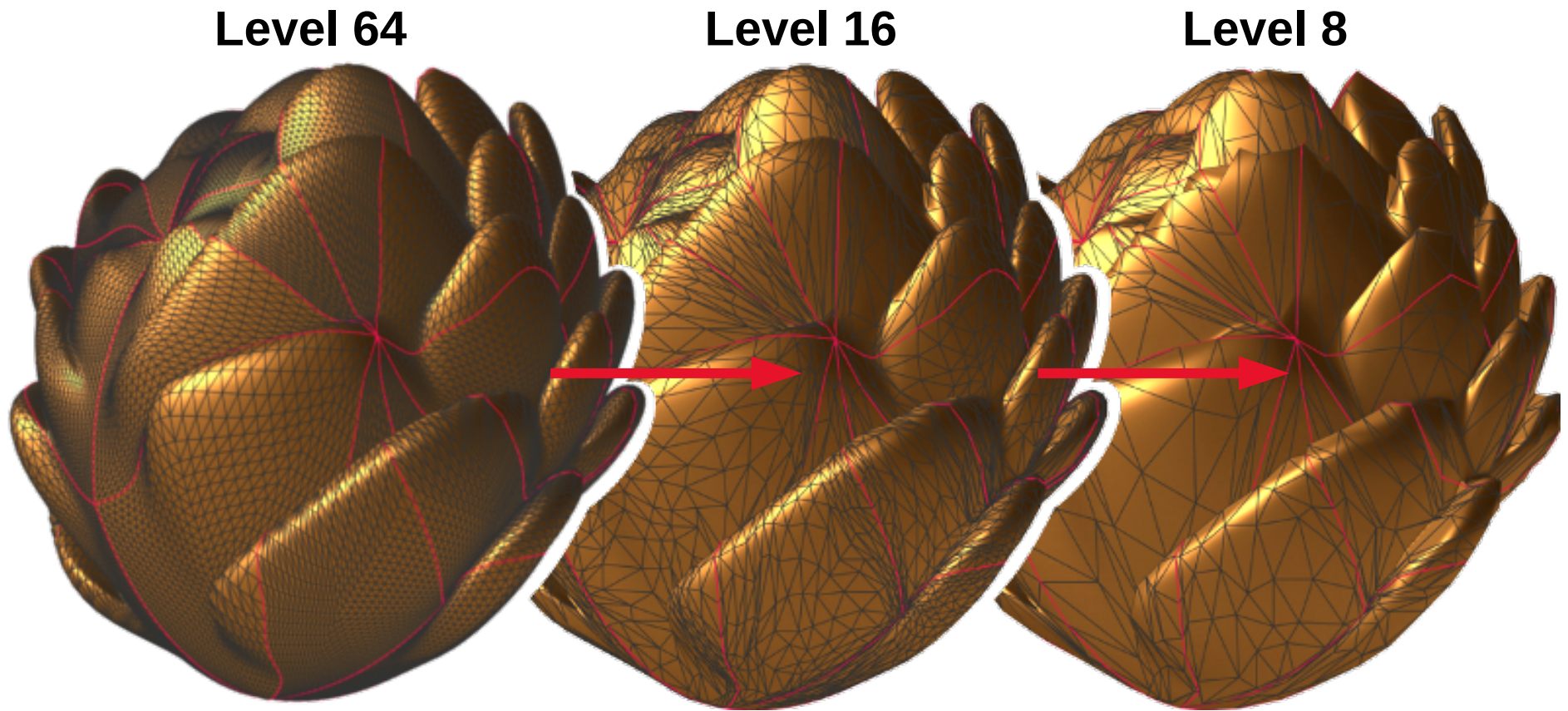
Level 8



## Smarter metric for simplification

- Global metric

# Future works



**Extend set of feasible contractions**

# Conclusion

## Contributions

- Novel interpolation scheme between tessellation levels
  - + Relaxed transition
  - + Swimming free
- Edge-collapse decimation
  - + Adapted to tessellation
  - + Feature-preserving

# Thank you

## **Multi-Resolution Meshes for Feature-Aware Hardware Tessellation**

