TU Delft Department of Management in the Built Environment
BK Bouwkunde

Multi-levels of details terrain construction for navigation

Qiuxian Wei 5801737

1st supervisor: Stelios Vitalis

2nd supervisor: Ken Arroyo Ohori

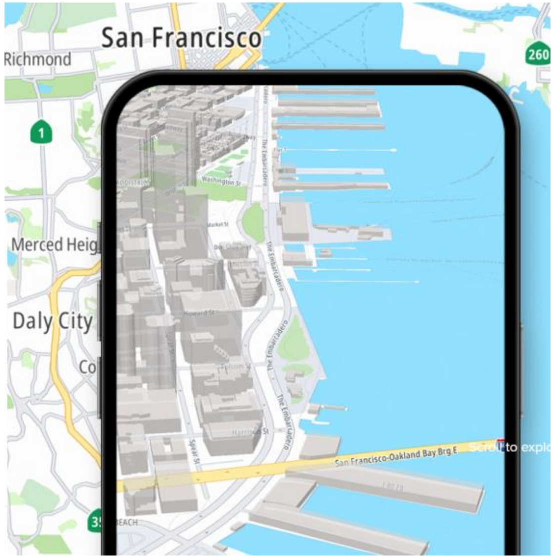
co-reader: Martijn Meijers

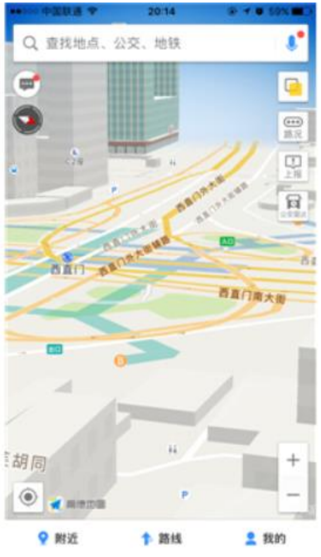
Delegator: Herman de Wolff

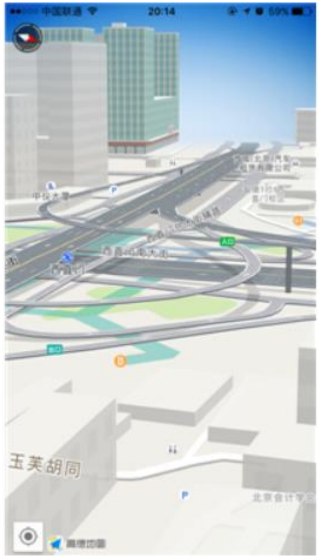
TU Delft

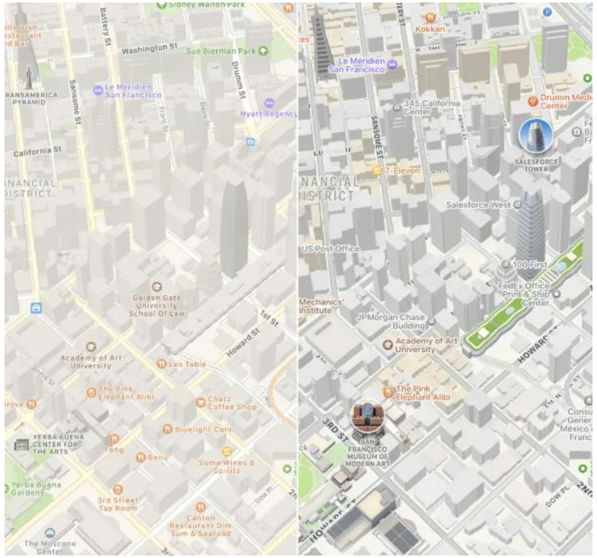
Catalog

1. INTRODUCTION —— 3D map for navigation








Washington St
Le Meridien San Francisco
California St
Golden Gate University School Of Law
Academy of Art University
The Pink Elephant Alibi
345 California Center
Salesforce Tower
JPMorgan Chase Building
San Francisco Museum of Modern Art

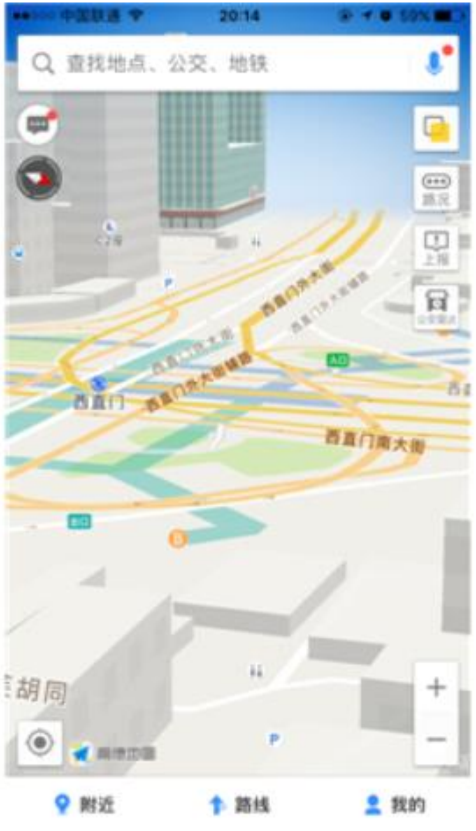
TomTom Map

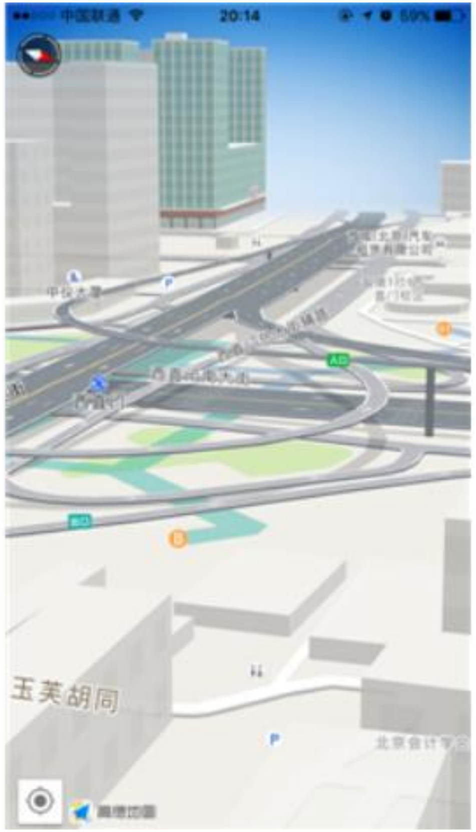
Gaode Map

Apple Map

1. INTRODUCTION —— 3D map for navigation

Gaode Map

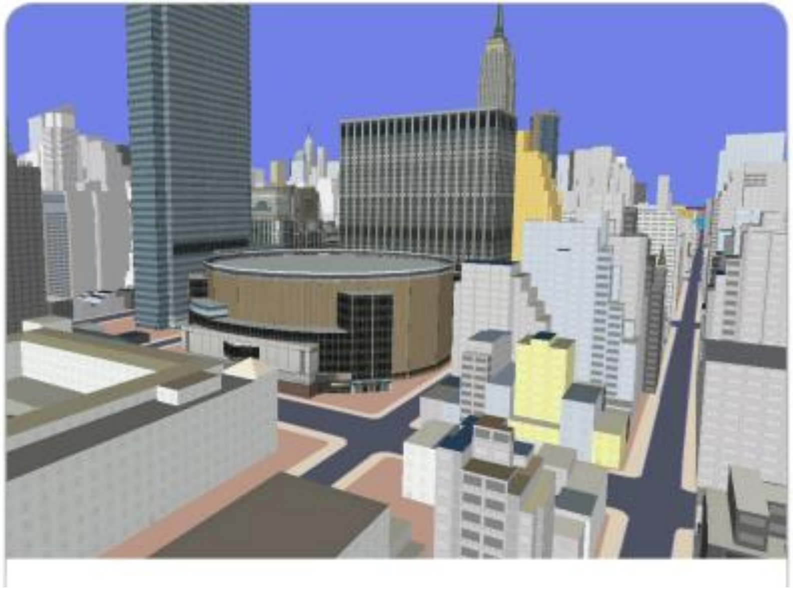




Complicated roads:
Flyovers, tunnels

Better visualization:
Landmarks

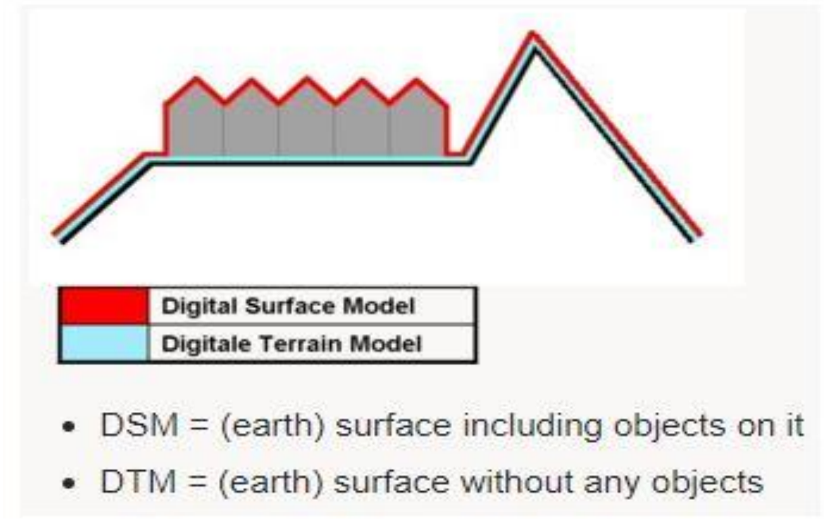
TomTom Map



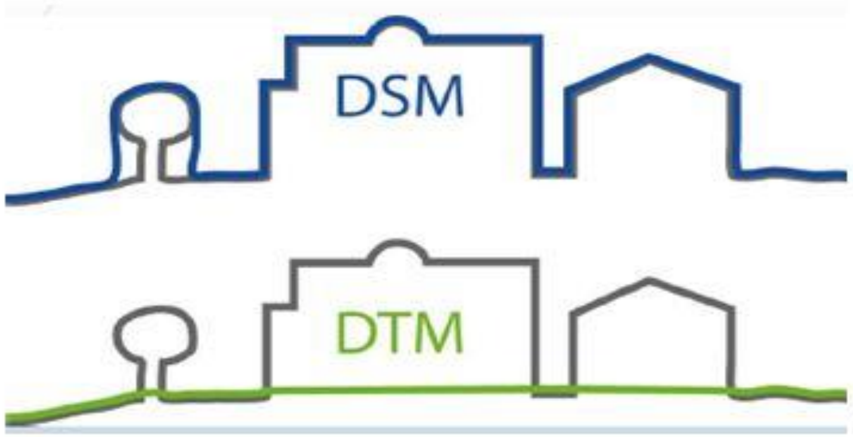
1. INTRODUCTION

Fundamental element:

Large-scale Terrain

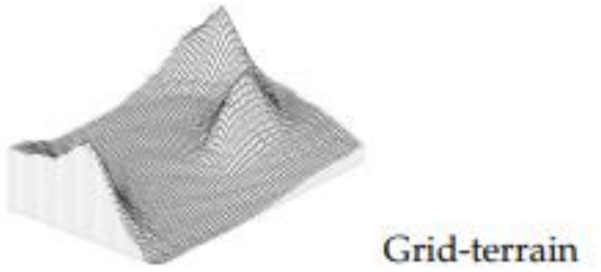


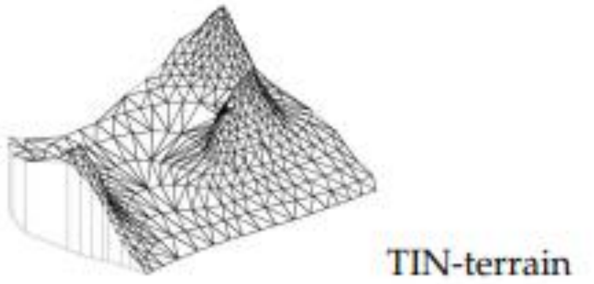
- DSM = (earth) surface including objects on it
- DTM = (earth) surface without any objects



1. INTRODUCTION

Two types of terrain representation:





Data redundancy in areas of uniform terrain.

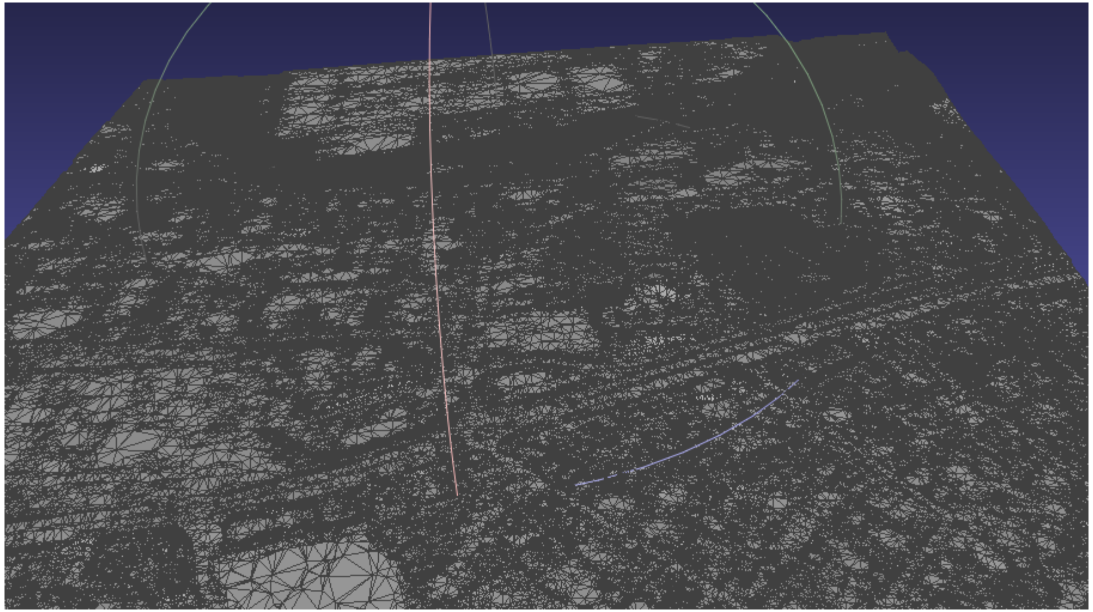
Inability to adapt to areas of differing relief complexity.

Non-redundant data

Allows extra data in complex areas and less data in non-complex areas

1. INTRODUCTION

Bottleneck: Huge data



Problem of such world-scale 3D modeling and visualization:

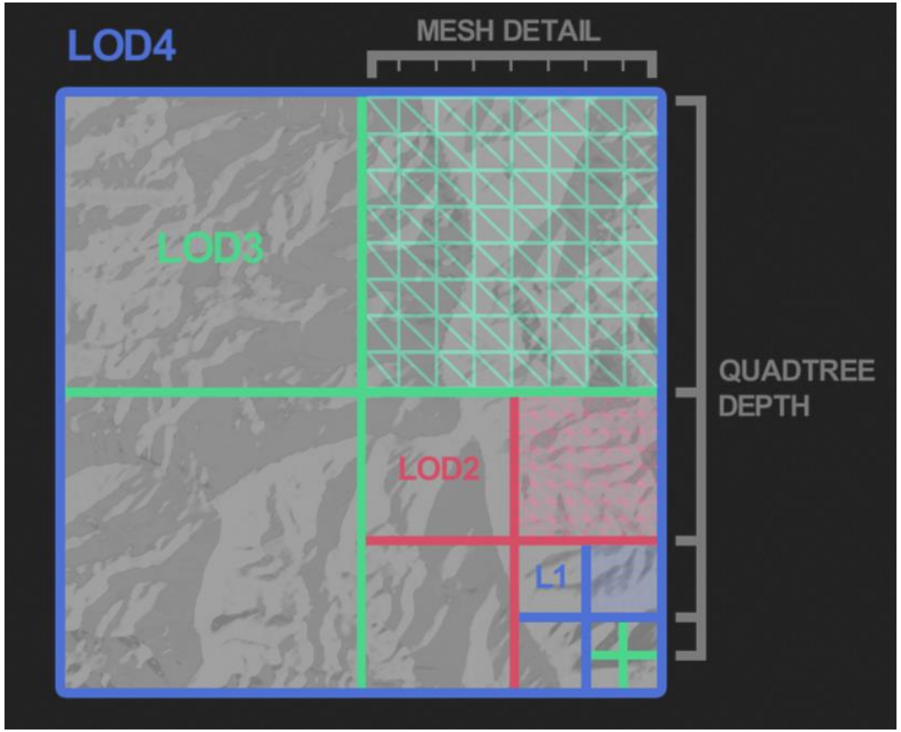
size of high-accuracy data > memory size and graphics processing capability

384,510 vertices for 1km * 1km terrain

1. INTRODUCTION —— Bottleneck

Solution for the huge dataset :

Requires hierarchical **multi-levels of details (LoDs)** terrain

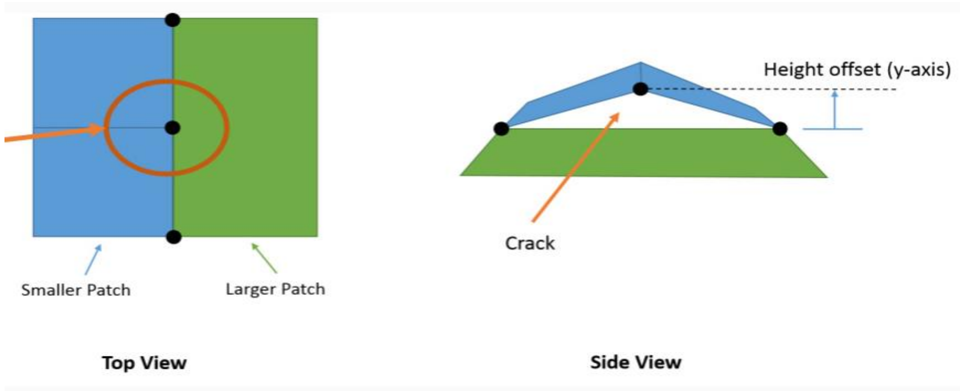


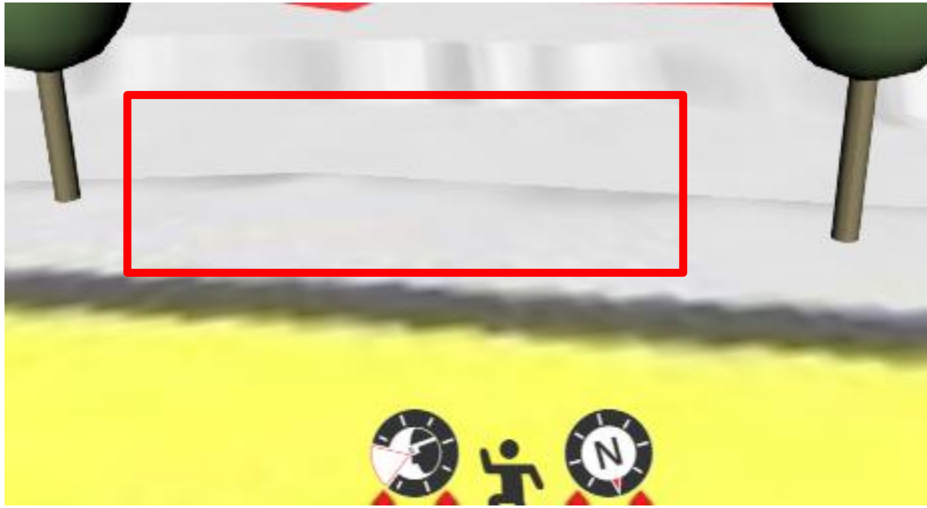
(https://svnte.se/cdlod-terrain)

1. INTRODUCTION —— Multi-LoDs problem

Multi-level terrain construction problem:

- Gap / Crack between different levels of detail patch/ tile — **T-juction.**



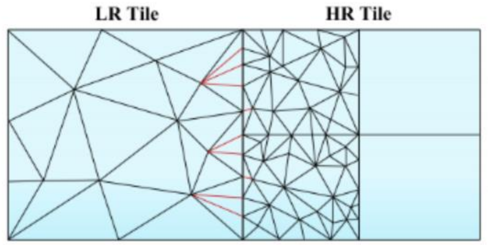


(https://victorbush.com/2015/01/tessellated-terrain/)

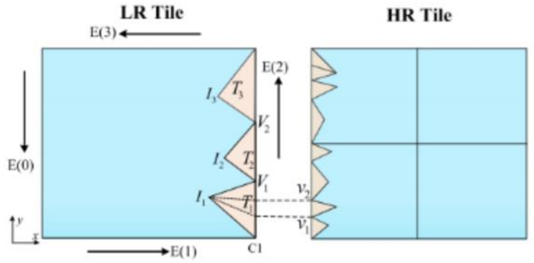
1. INTRODUCTION —— Multi-LoDs problem

T-junction problem: Locked boundaries

- Constrain the points on boundaries locked and remain unchanged during multi-LoD construction.



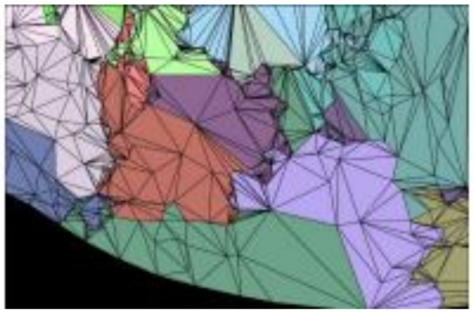
(a)

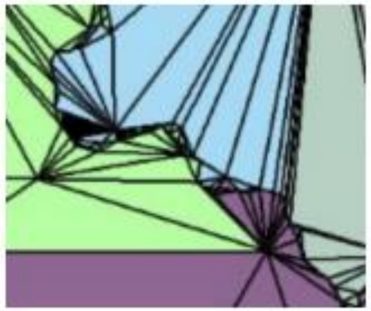


(b)

Locked boundaries: Complicated edges

The locked boundaries remain unchanged during simplification and cause triangle density.



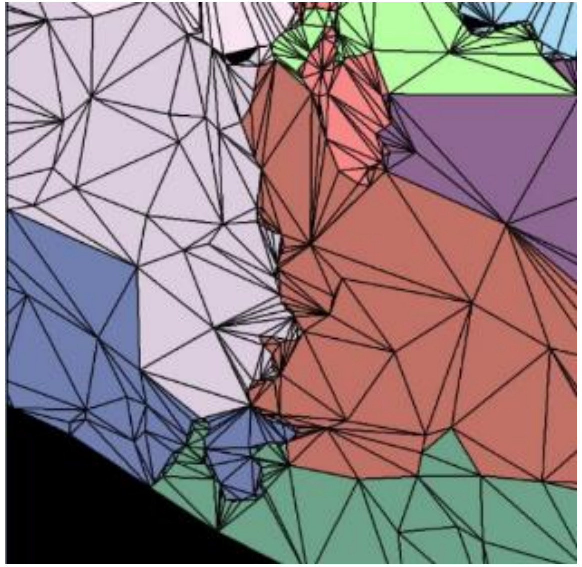


1. INTRODUCTION —— Multi-LoDs problem

Locked boundaries: Complicated edges

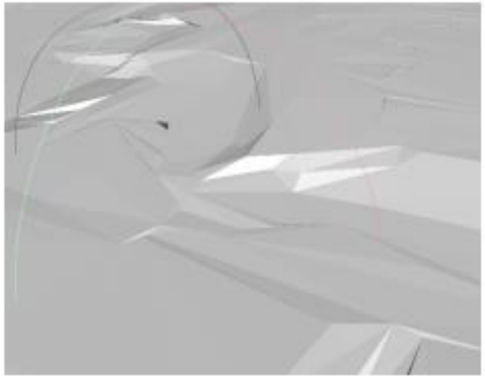
Simplification inefficiency

The points on the boundaries can never be simplified.

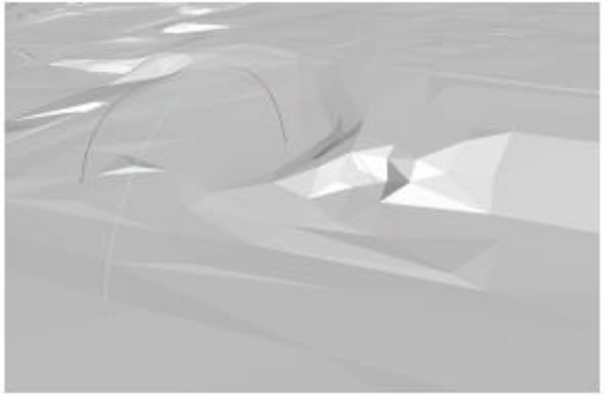


Visualization artifacts

The complicated edges on the left has a more rigid transition compare to the right visualization result.

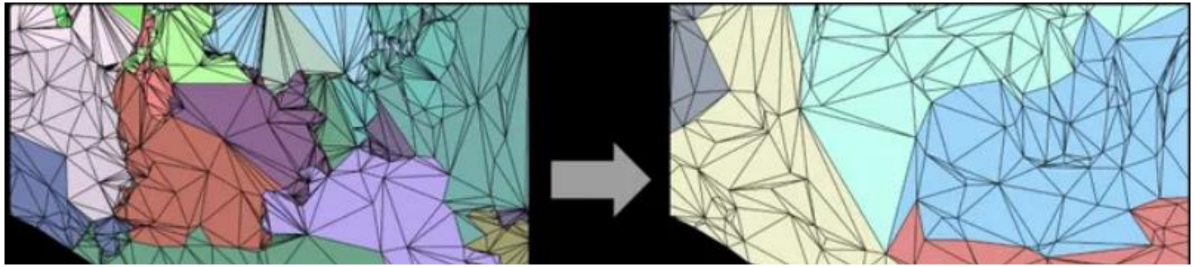


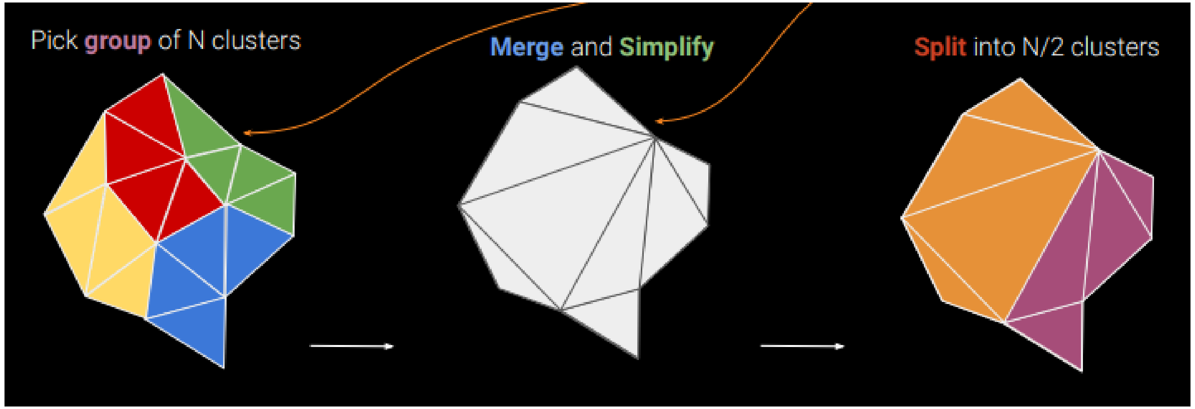
[illegible]



1. INTRODUCTION —— Multi-LoDs Solution

Nanite: triangle clustering





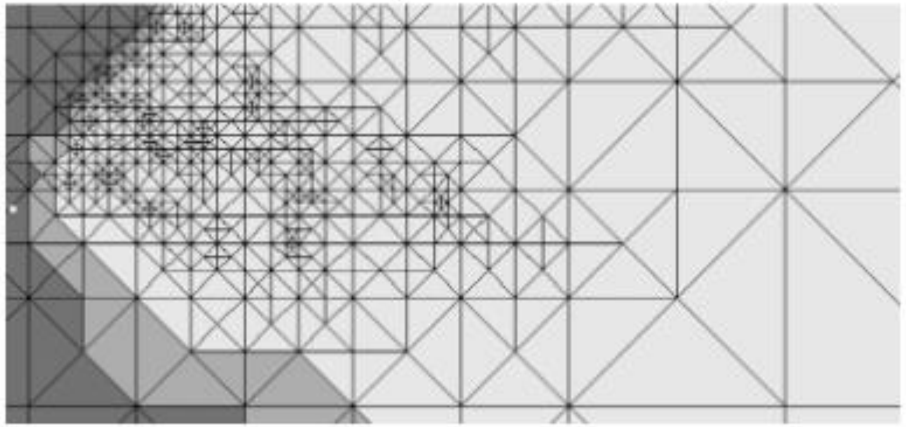
- Partition the triangles in the model mesh into different clusters.
- Group the triangle clusters, locked the boundaries of the clusters and simplify inside the clusters.
- Re-partition again for the next LoD

However, it’s not suitable for web mapping schema.

1. INTRODUCTION —— Multi-LoDs Solution

Web mapping — Pre-generated data relies on client-server architecture

- Real-time simplification?
- — Not for phones and navigation purposes
  - Large detailed data is costly either stored locally or transmitted online
  - For navigation, the rendering time of the visualization is vital, real-time simplification can be time-consuming



Real-time Optimally Adapting Meshes （ROAM）ing terrain

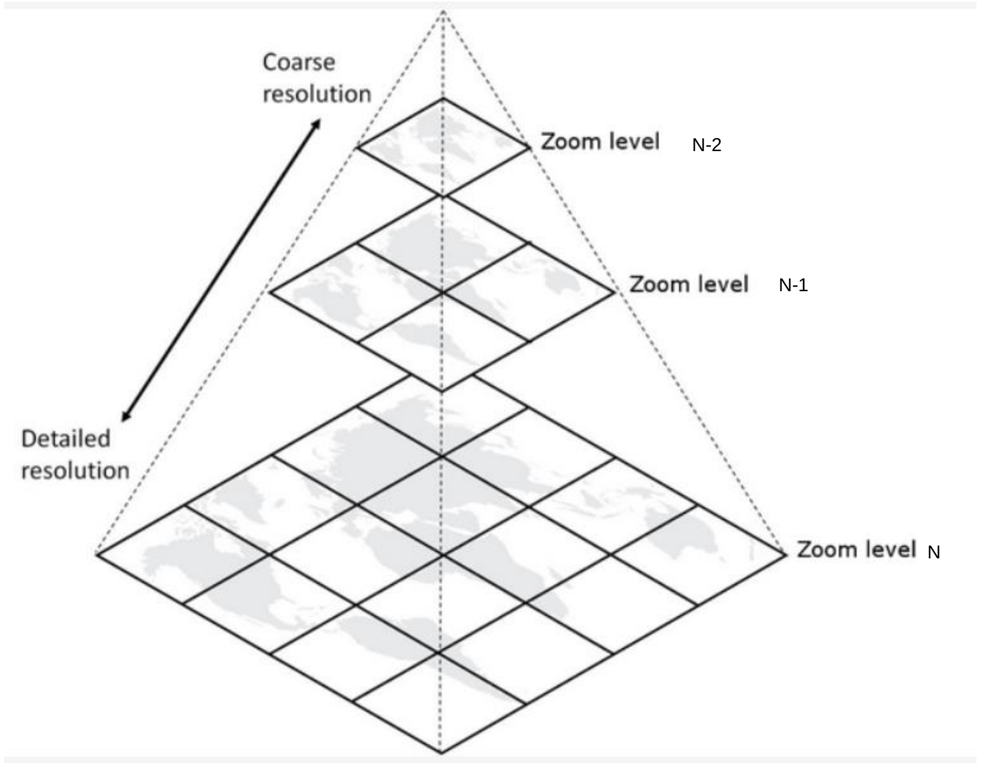
1. INTRODUCTION —— Multi-LoDs Solution

Solution for pre-generated web mapping:

- Tiled web map:

- Zoom level is the scale of the map.

- The zoom n contains **2^n** tiles.



https://support.plexearth.com/hc/en-us/articles/6325794324497-Understanding-Zoom-Level-in-Maps-and-Imagery

1. INTRODUCTION

Research question:

How to construct seamless large-scale multi-LoD tiled terrain, with consideration of constraints in the process of simplification?

1. INTRODUCTION – Sub questions

How to simplify the model to generate different LoD models?

- How to determine the LoDs threshold? Based on the model errors after simplification or geometry features such as vertices/edges/faces number?

- What kind of simplification methods? Edge-collapse? Greedy insertion? Point set simplification?

- **How to partition the mesh into different clusters?**

  - How does the number of clusters affect the simplification result?

  - How to partition the mesh so that the locked boundaries have less influence on the simplification efficiency?

1. INTRODUCTION – Sub questions

- **How to preserve boundaries of the clusters during simplification?**

The boundaries of the clusters need to remain unchanged during simplification, otherwise there would be cracks between different LoD models.

- Set constraints to terrain and simplified them together?

- **How to assign the clusters to the tiles?**

- **How to assess the quality of the simplification methods?**

  - Size of the mesh geometry (vertices, edges, faces) after simplification?

  - How to calculate the error of the simplified terrain?

  - How to ensure the connection of the different LoDs tiles is visually seamless?

1. INTRODUCTION – Research scope

The partition of the triangles: METIS - Serial Graph Partitioning and Fill-reducing Matrix Ordering (Karypis and Kumar, 1999).

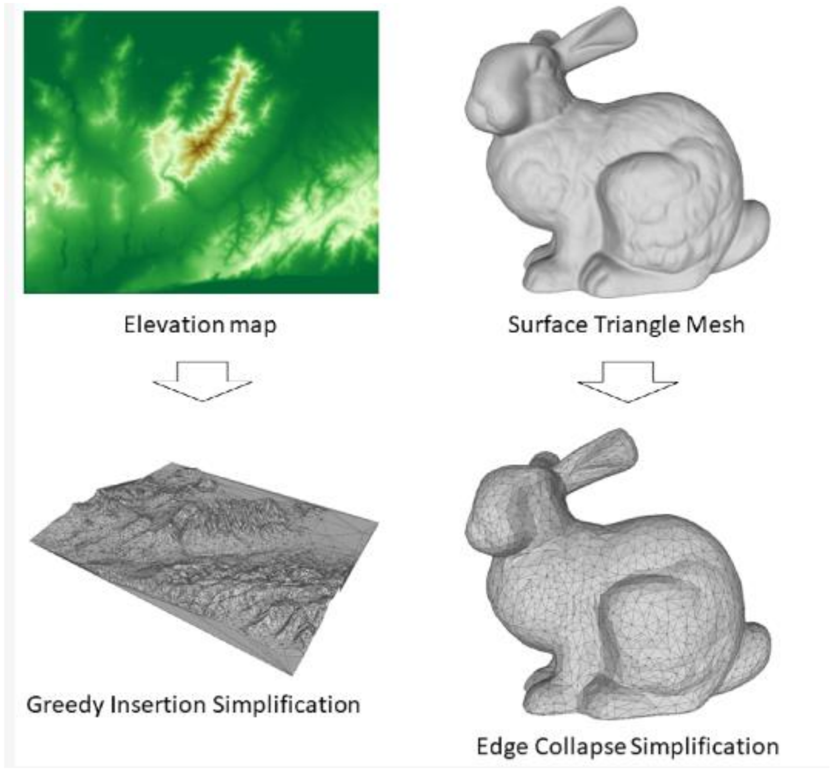
The implementation of mesh simplification: The Computational Geometry Algorithms Library (CGAL) (cga, 2021), and greedy insertion is implemented by tin-terrain (HERE Technologies, 2024).

The main challenge is adapting the algorithm choices to the **web tile schema**.

Note that this thesis only investigates the build of seamless multi-LoD terrain, the elements attached to the terrain, such as roads and building footprints, are not within the scope of this thesis.

2. Related work

Simplification methods:



Greedy insertion:

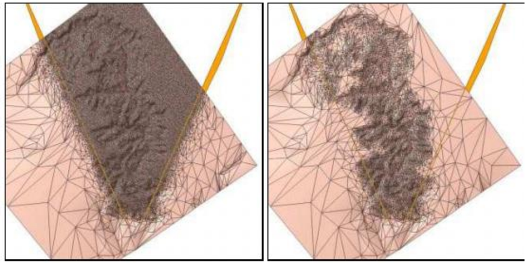
coarse-to-fine simplification. It starts with a very basic triangulation, They found the most effective way is to calculate the absolute height error added to the mesh at each iteration until all the points are within a userdefined tolerance.

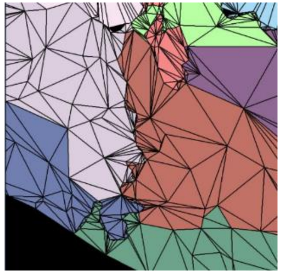
Edge collapse:

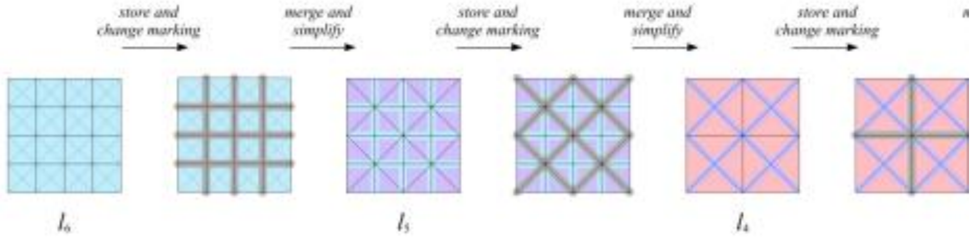
fine-to- coarse simplification. At each iteration, it collapses the edge that would induce the smallest error.

2. Related work – Summary

Multi-LoD Construction:

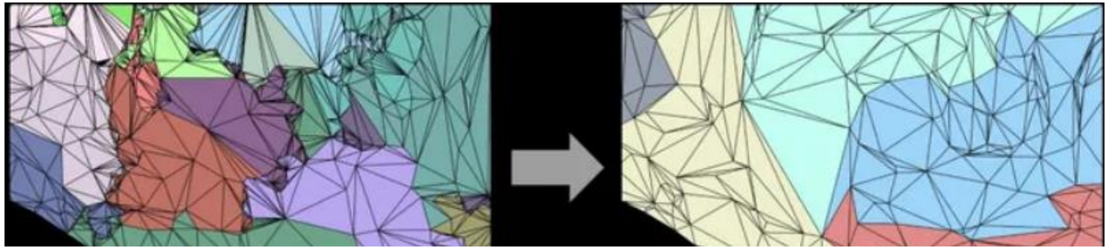






Real-time: Density accumulation

Regular subdivision structure: extra points needed.

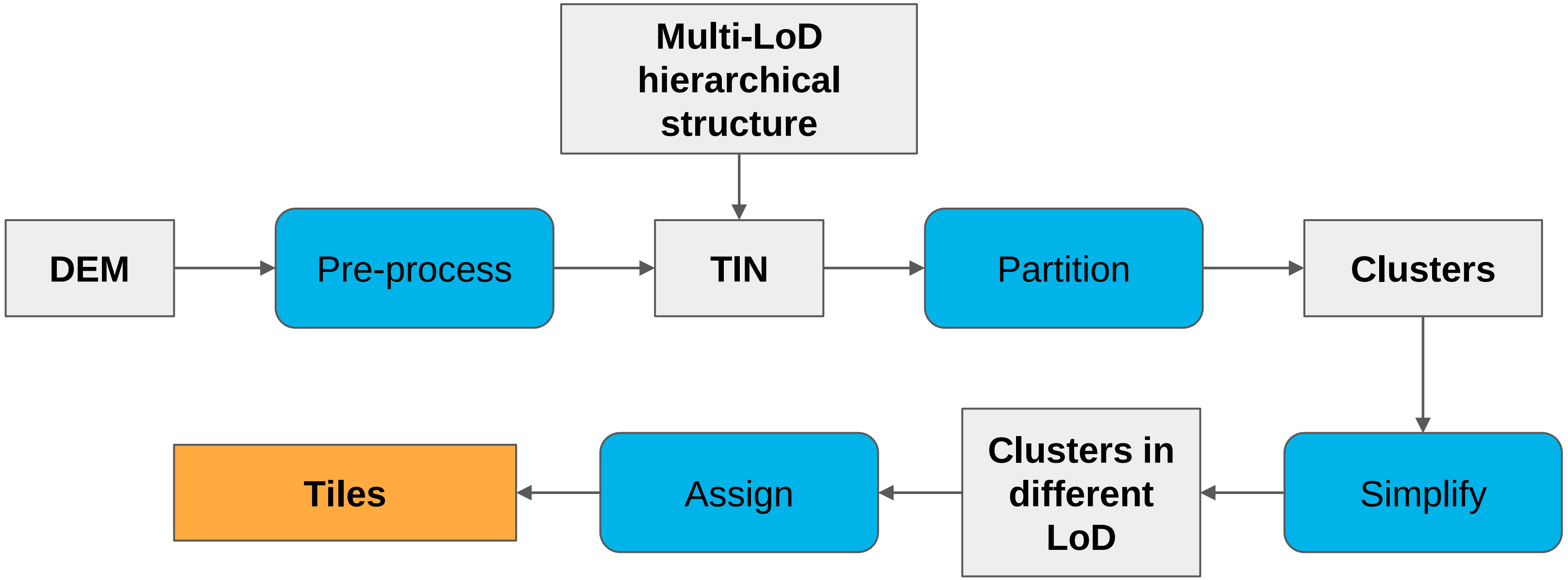


How to adapt it to tiled web map?

Triangle clustering: Based on local memory, not Client/Server architecture.

3. Methodology

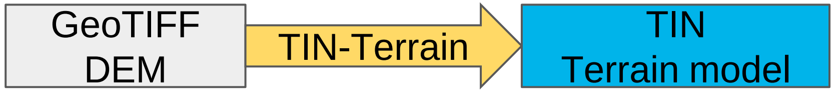
Implementation Steps

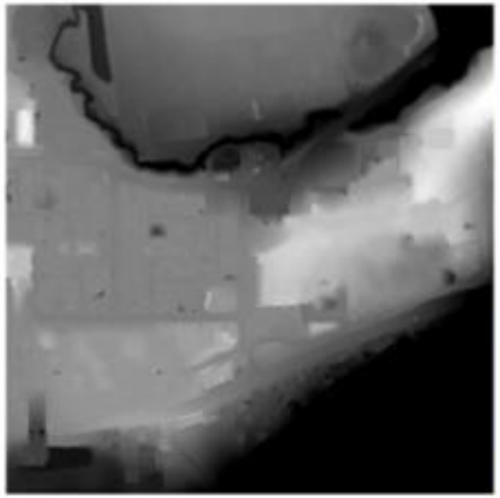


3. Methodology

Pre-process of the data: Generate TIN data

Tin-terrain (HERE Technologies, 2024) is used to derive a TIN model from DEM data. It utilizes height maps in GeoTIFF format and uses the Terra method for greedy insertion simplification

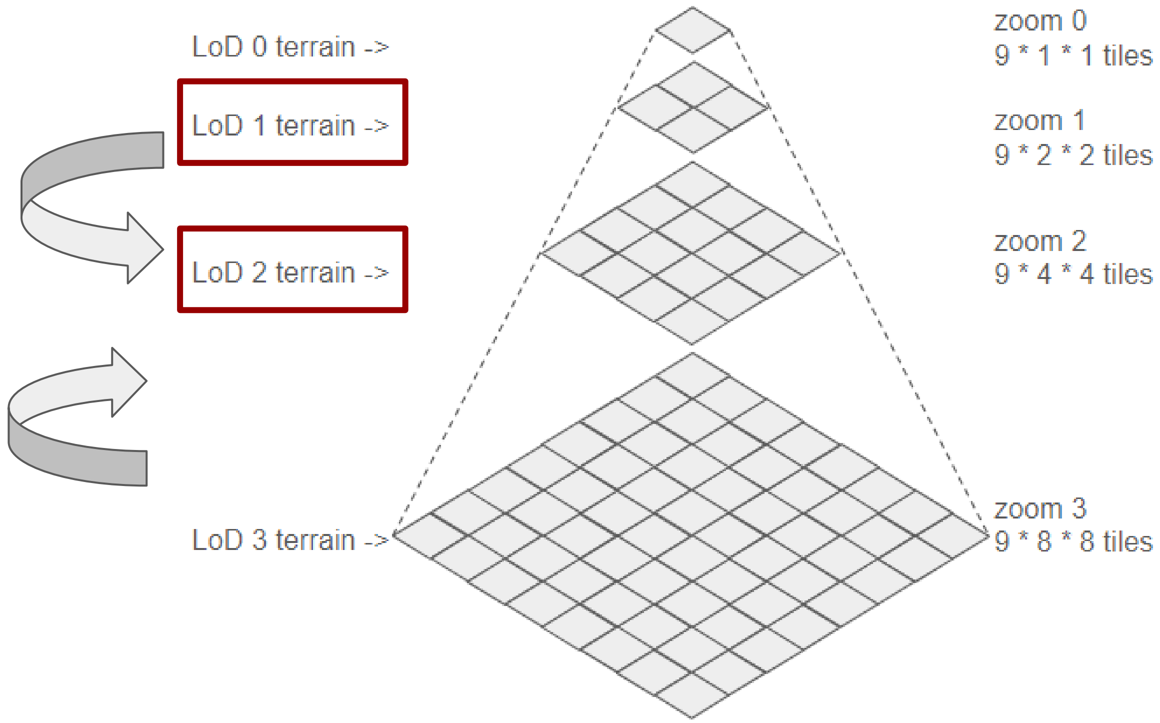




3. Methodology

Construction of Multi-zoom-level Tiles Hierarchy

To test the connection between tiles at different zoom levels, we establish a simple hierarchy with four zoom levels.

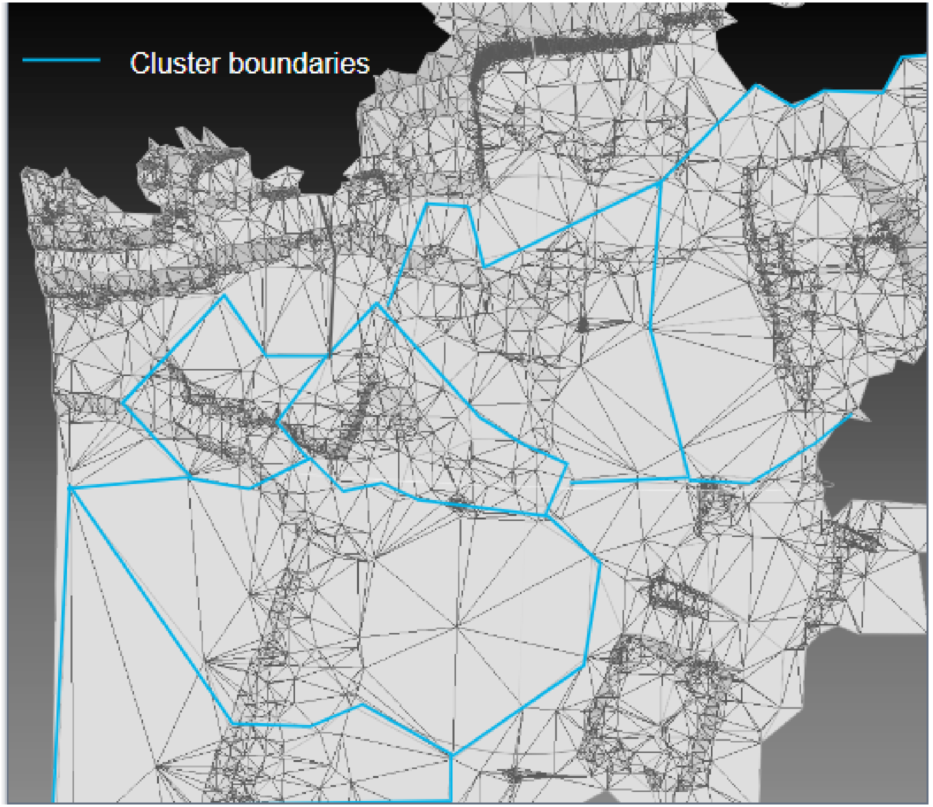


3. Methodology

Partition TIN into different clusters

The boundaries of the clusters is locked during simplification: the choice of locked boundaries influences the simplification result.

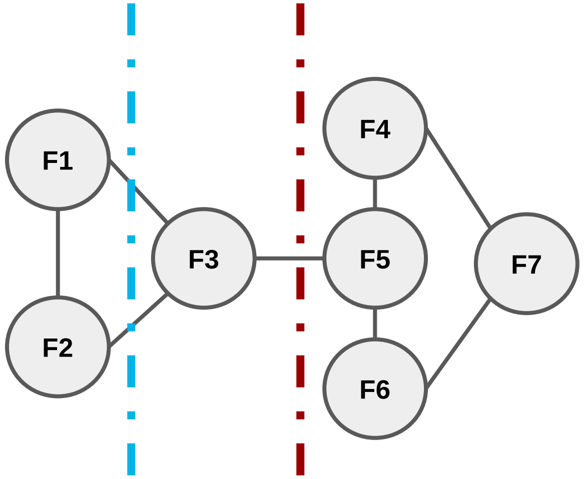
The fewer locked boundaries, the better.



3. Methodology

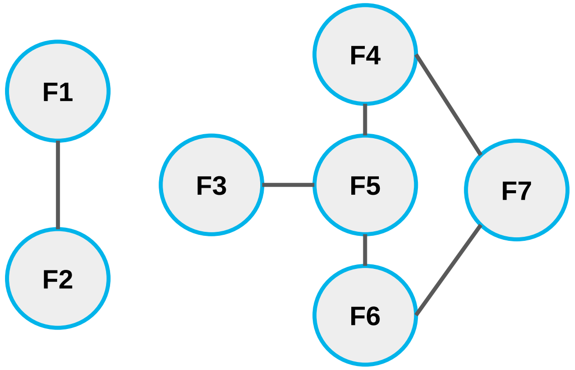
Partition TIN into different clusters

Edge Cut Minimization

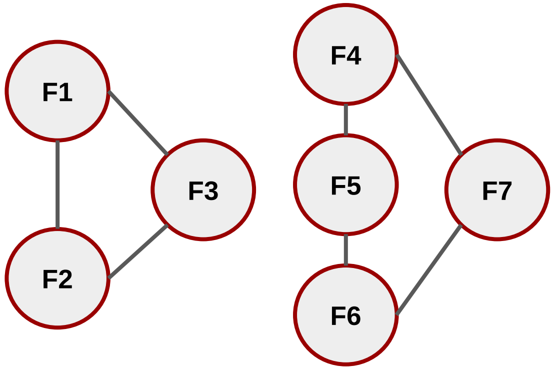


● Node: Face of the mesh

— Edge: Face connection



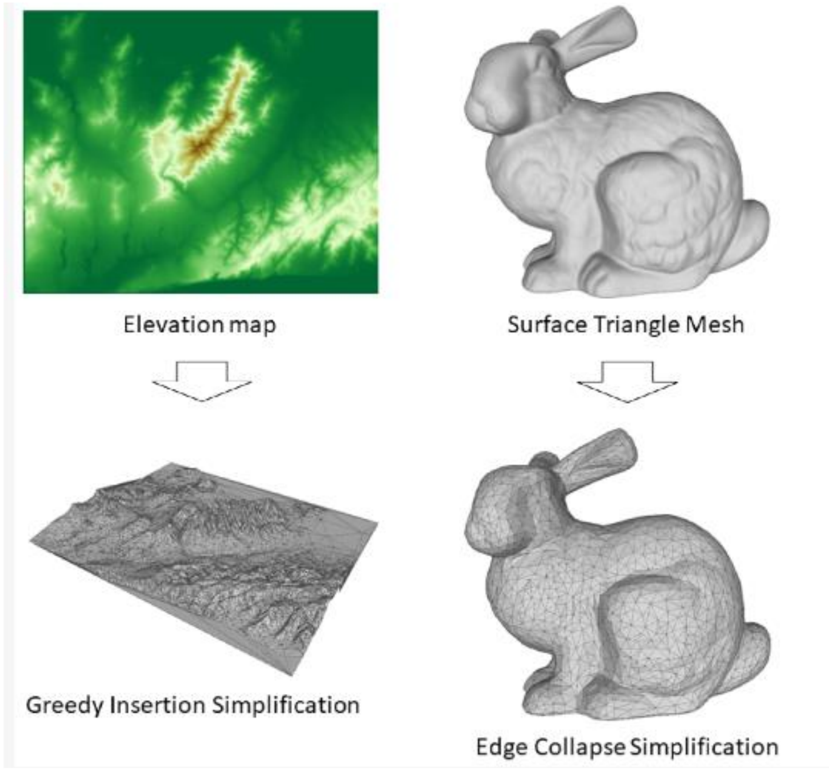
Edge-cut: 2
Boundaries: 9



Edge-cut: 1
Boundaries: 7

3. Methodology

Simplify the mesh with constraints

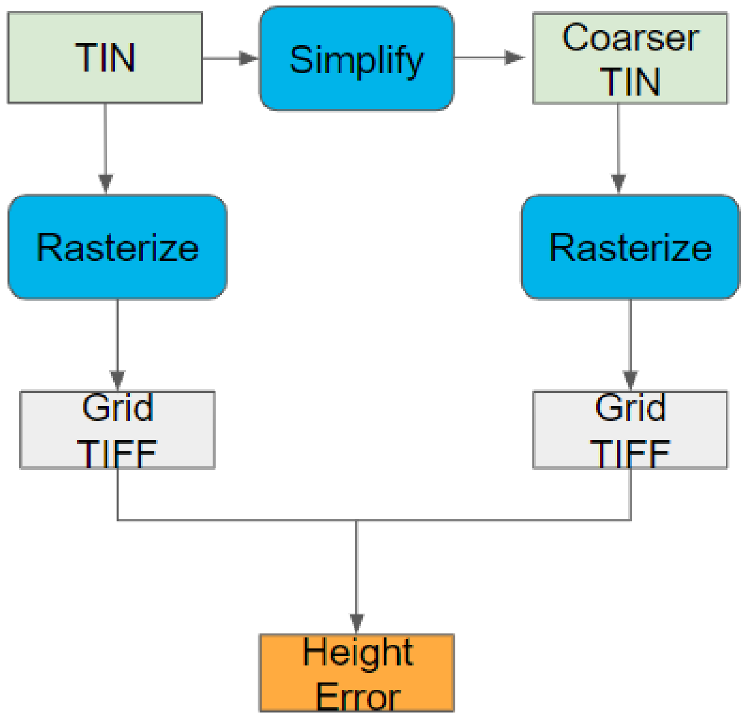


Greedy insertion: Performs coarse-to-fine simplification.

Edge-collapse simplification: At each iteration, it collapses the edge that would induce the smallest error.

3. Methodology

Simplify the mesh with constraints

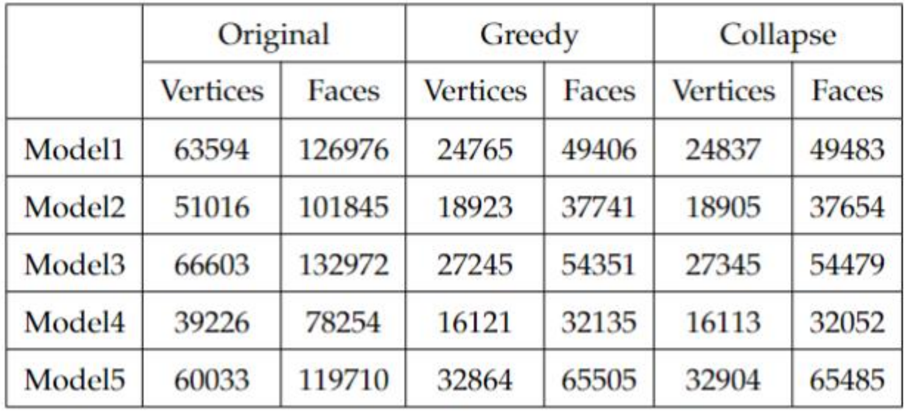


To calculate the height in each area, the TIN models are rasterized to a grid data with a resolution of 0.5 meters × 0.5 meters using interpolation, enabling the calculation of height per grid.

3. Methodology

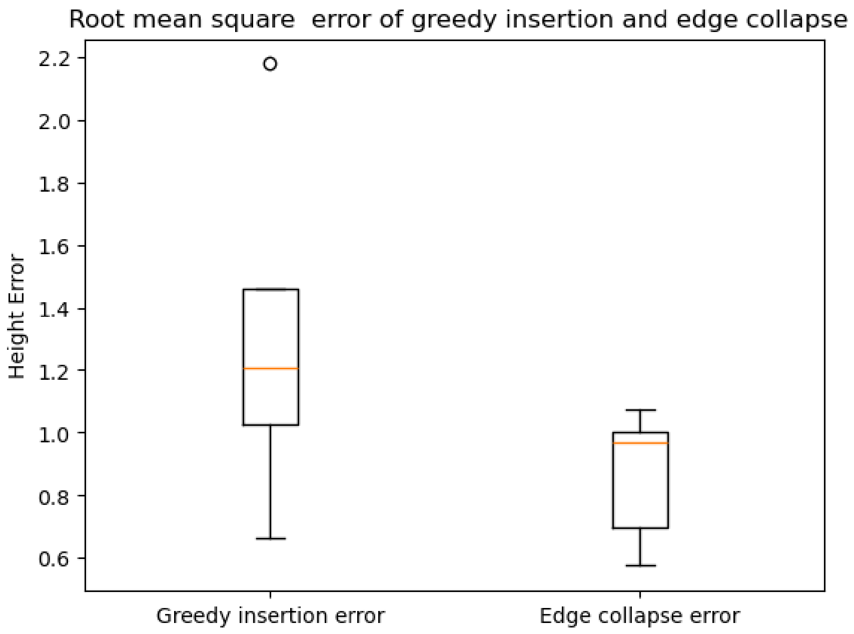
Simplify the mesh with constraints

Perform two simplification on five different terrain, the results are similar. Greedy insertion is more sensitive to height and likely to produce more triangles.



	Original		Greedy		Collapse	
	Vertices	Faces	Vertices	Faces	Vertices	Faces
Model1	63594	126976	24765	49406	24837	49483
Model2	51016	101845	18923	37741	18905	37654
Model3	66603	132972	27245	54351	27345	54479
Model4	39226	78254	16121	32135	16113	32052
Model5	60033	119710	32864	65505	32904	65485

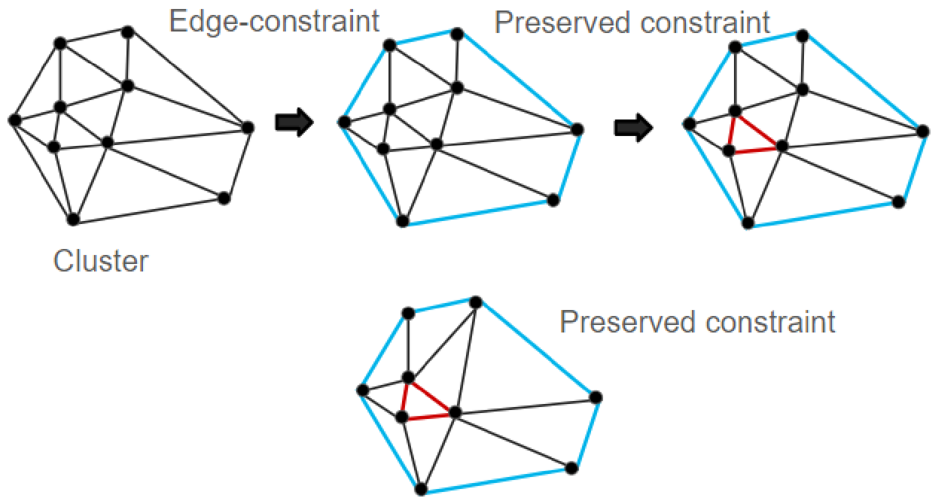
Table 1: Vertices and faces number of the models



3. Methodology

Simplify the mesh with constraints

Potential elements on the terrain, such as roads and building footprints that need preservation during simplification, are set as **constraints**. These constraint edges are considered **unremovable** during edge-collapse simplification.

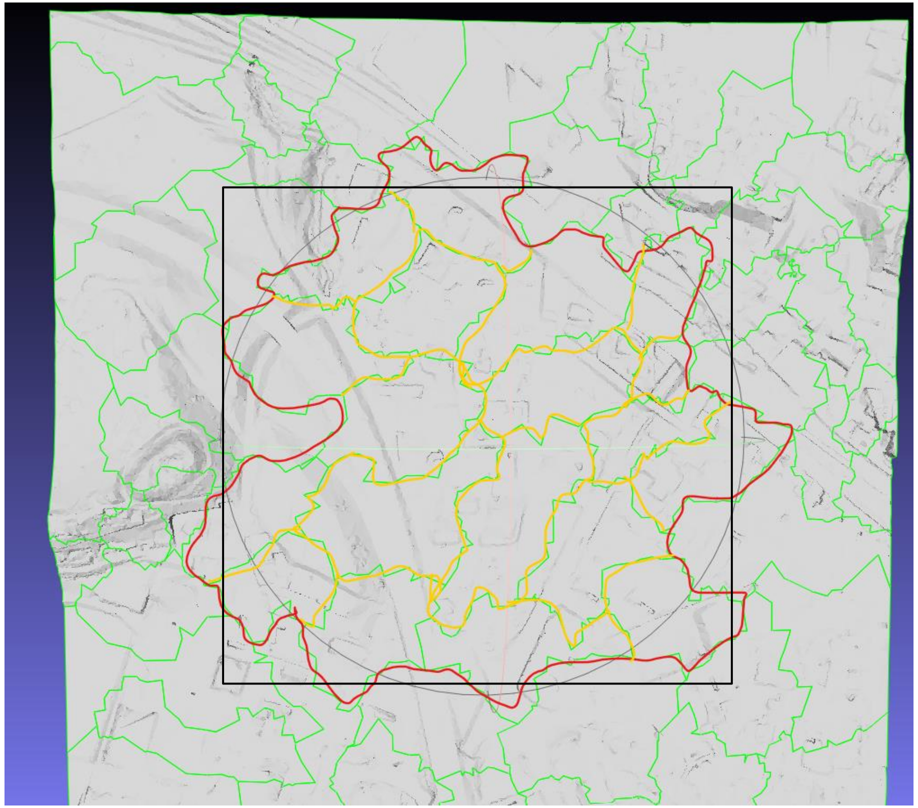


3. Methodology

Assign clusters to tiles and generate tiles

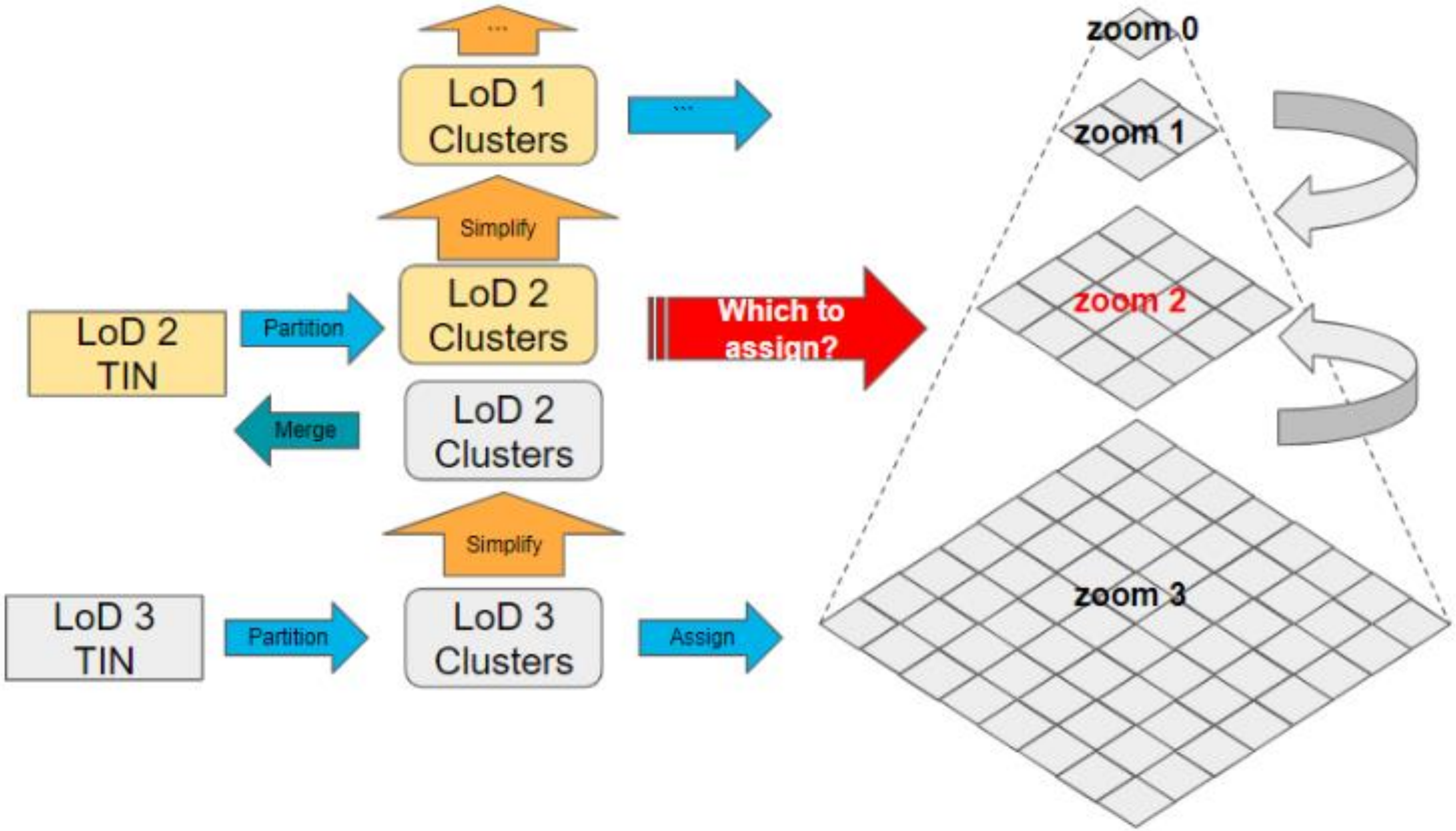
- After partitioned, the clusters are assigned to tiles to generate web tiles.
- Based on the centroid of clusters.
- After assigning the clusters, the result is double check so each tile has clusters. Otherwise the neighbouring tiles are partitioned again.

—	Tile boundaires
🟥	Outer clusters boundaires
🟨	Inner clusters boundaires
🟩	Other clusters boundaires not assigned to this tile



3. Methodology

Assign clusters to tiles and generate tiles

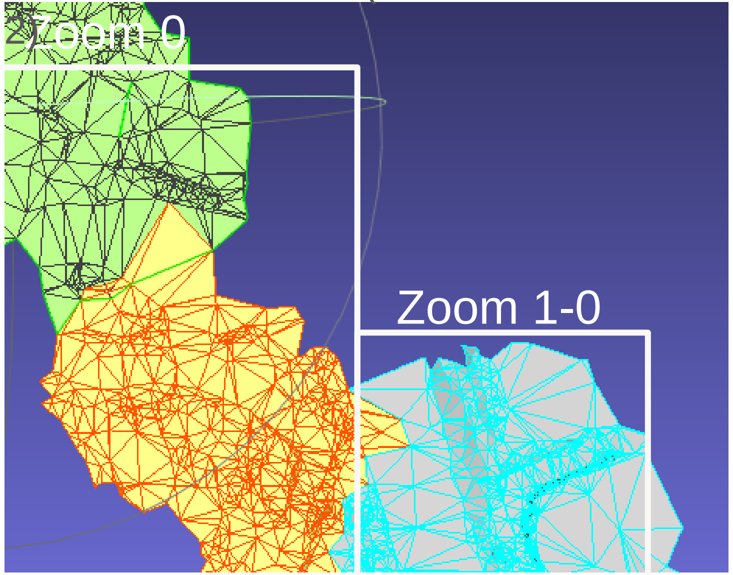


3. Methodology

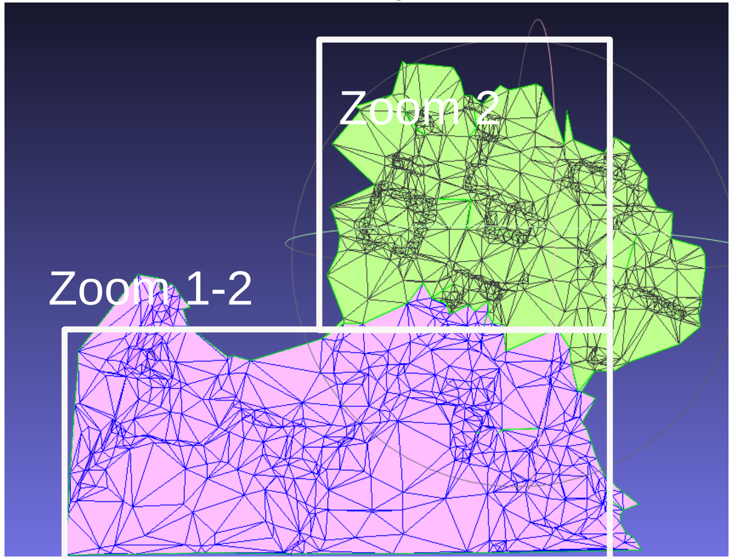
Assign clusters to tiles and generate tiles

With same cluster boundaries, the differen zoom level tiles can be connected with boundaries completely aligned.

Zoom 0 + zoom 1-0 (Same cluster with zoom 0)



Zoom 2 + zoom 1-2 (Same cluster with zoom 2)

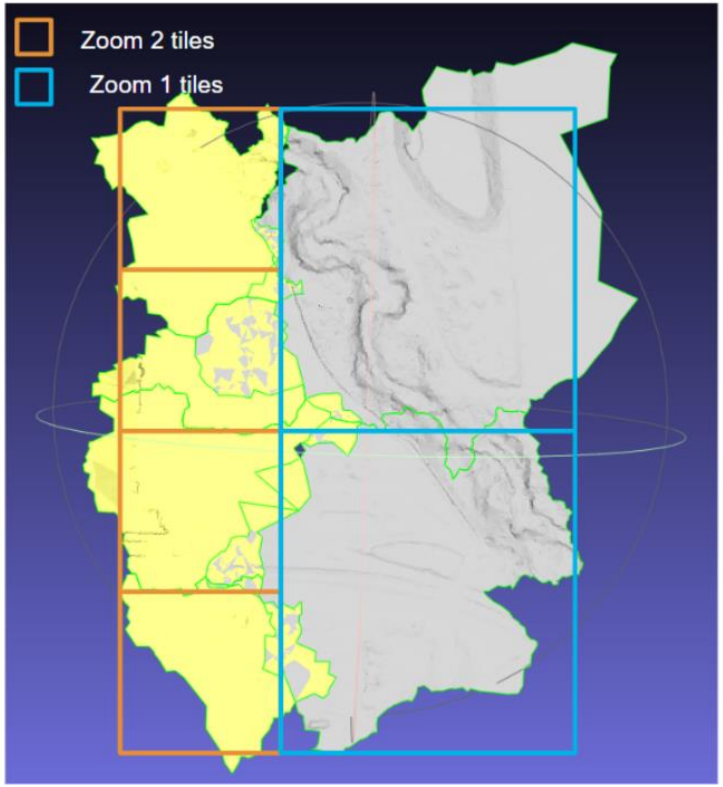


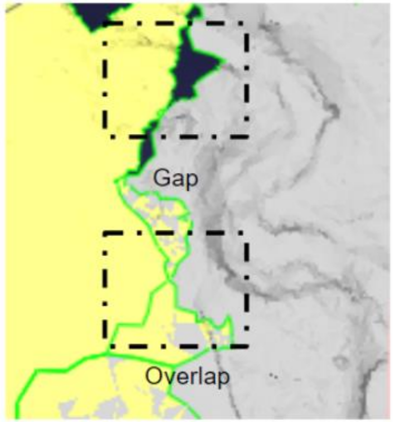
3. Methodology

Assign clusters to tiles and generate tiles

However, if the cluster boundaries are not the same, there are gaps and overlaps between the different zoom tiles.

The picture shows zoom 2 tiles connect with zoom 1-0 tiles.

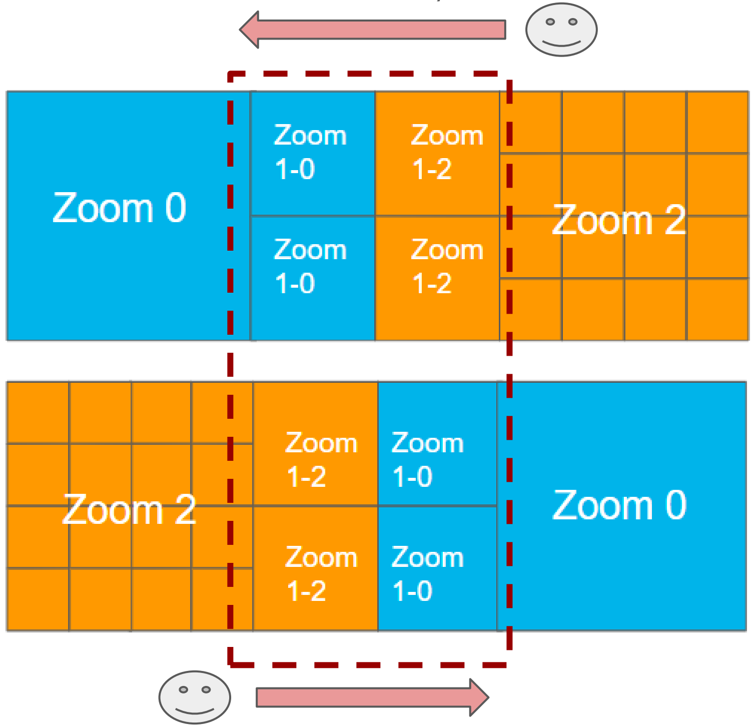


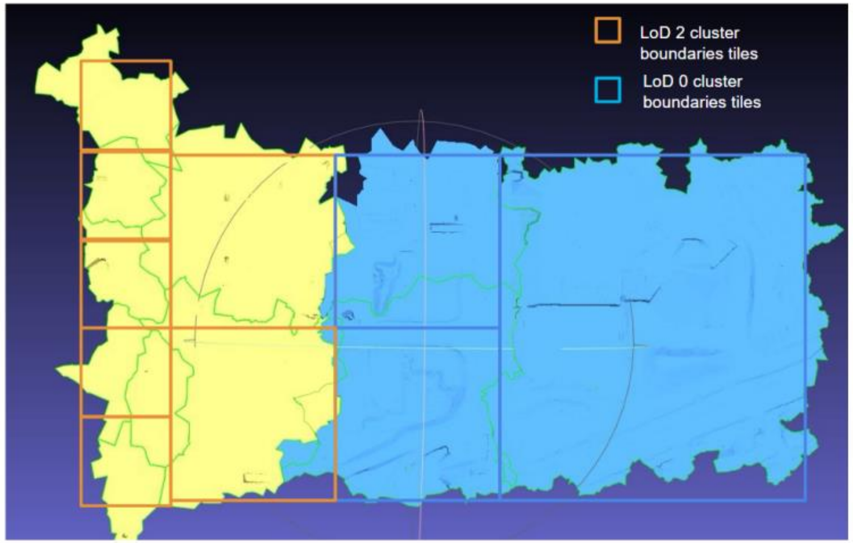


3. Methodology

Assign clusters to tiles and generate tiles – Double Zoom Tiles

However, although different zoom tiles can connect with each other with completely aligned. This method is **costly** for web mapping. Similarly in navigation, If the user navigates back and forth within the same area, the same tiles will be retrieved from the server **twice.**





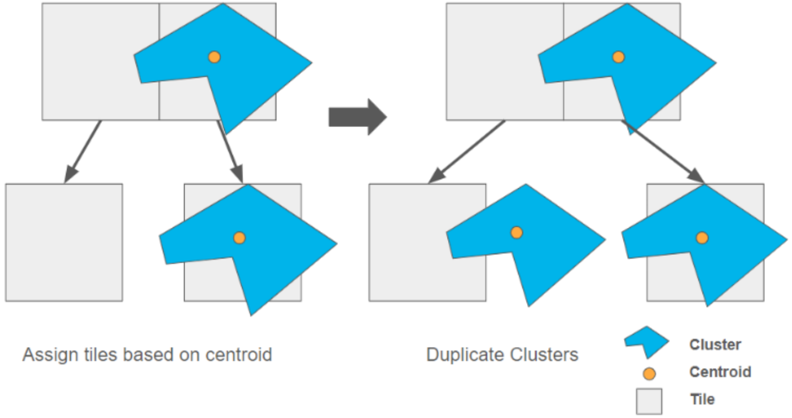
LoD 2 cluster boundaries tiles
LoD 0 cluster boundaries tiles

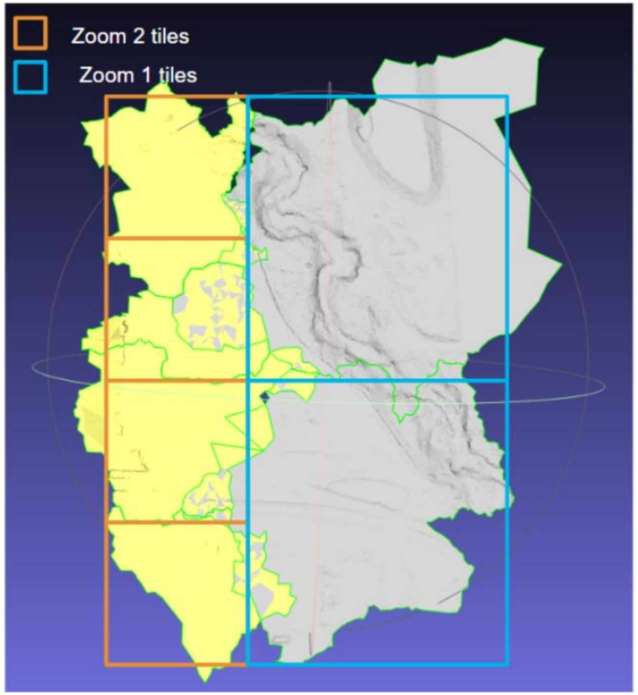
3. Methodology

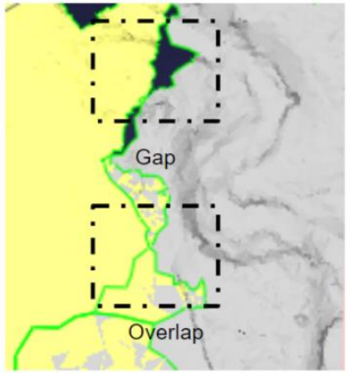
Assign clusters to tiles and generate tiles – Duplicate clusters

Overlap: data redundancy, but doesn't influence seamless visualization.

Gaps: The clusters in the gap areas are assigned to neighbouring tiles. Thus if it's assigned to both tiles, there wouldn't be gaps.

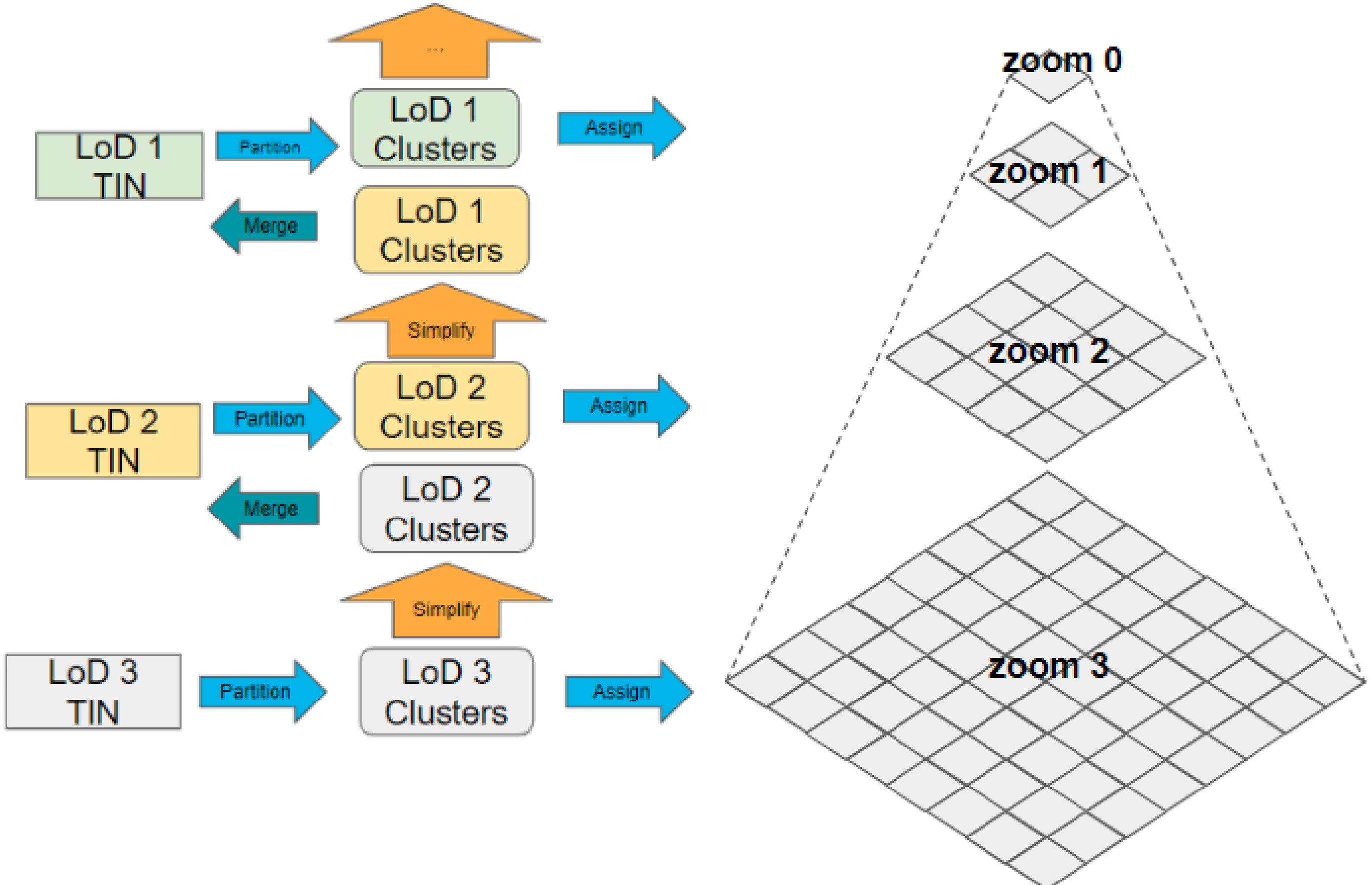






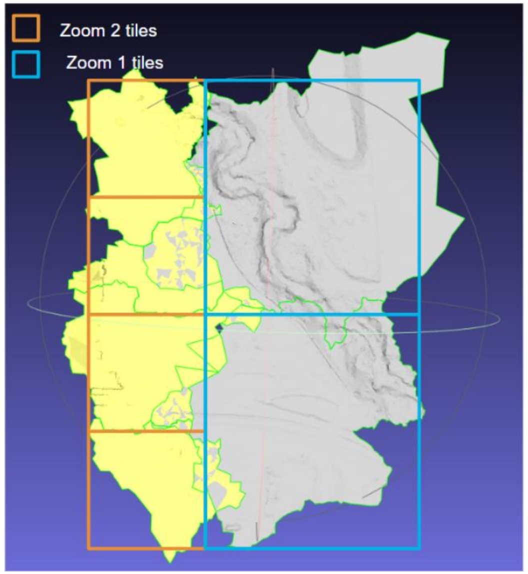
3. Methodology

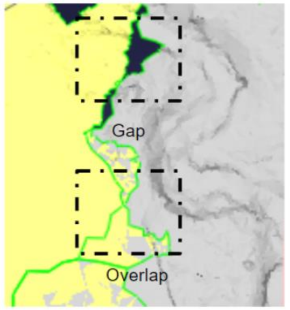
Assign clusters to tiles and generate tiles – Duplicate clusters

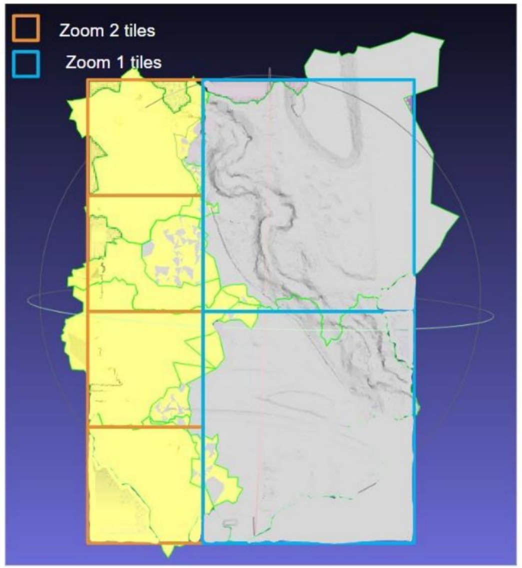


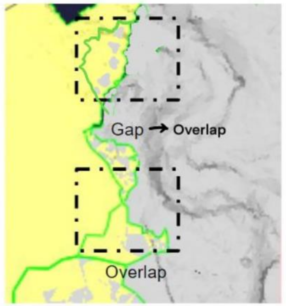
3. Methodology

Assign clusters to tiles and generate tiles – Duplicate clusters



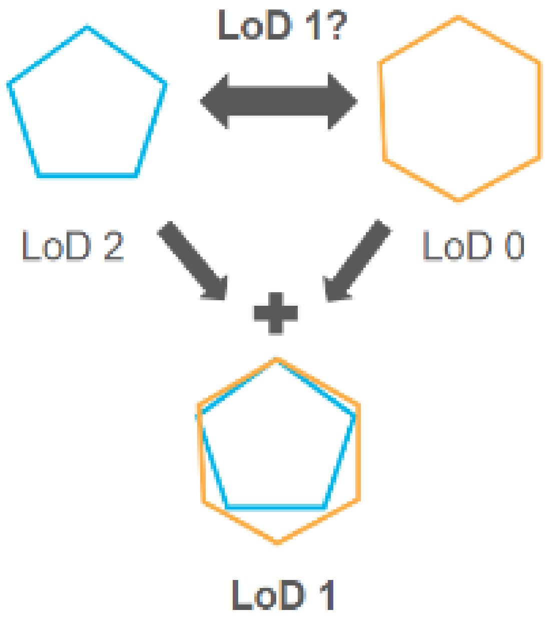


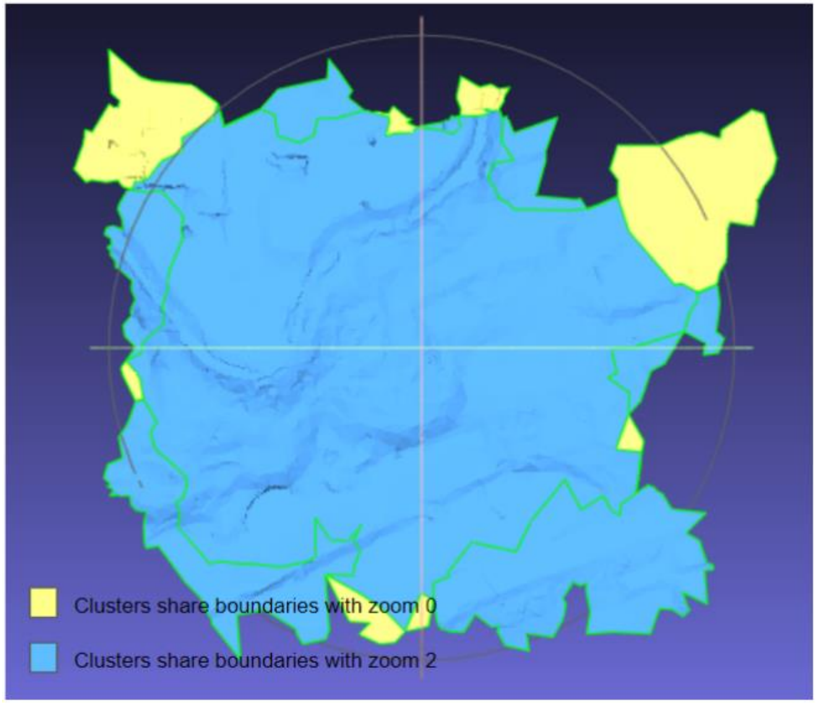


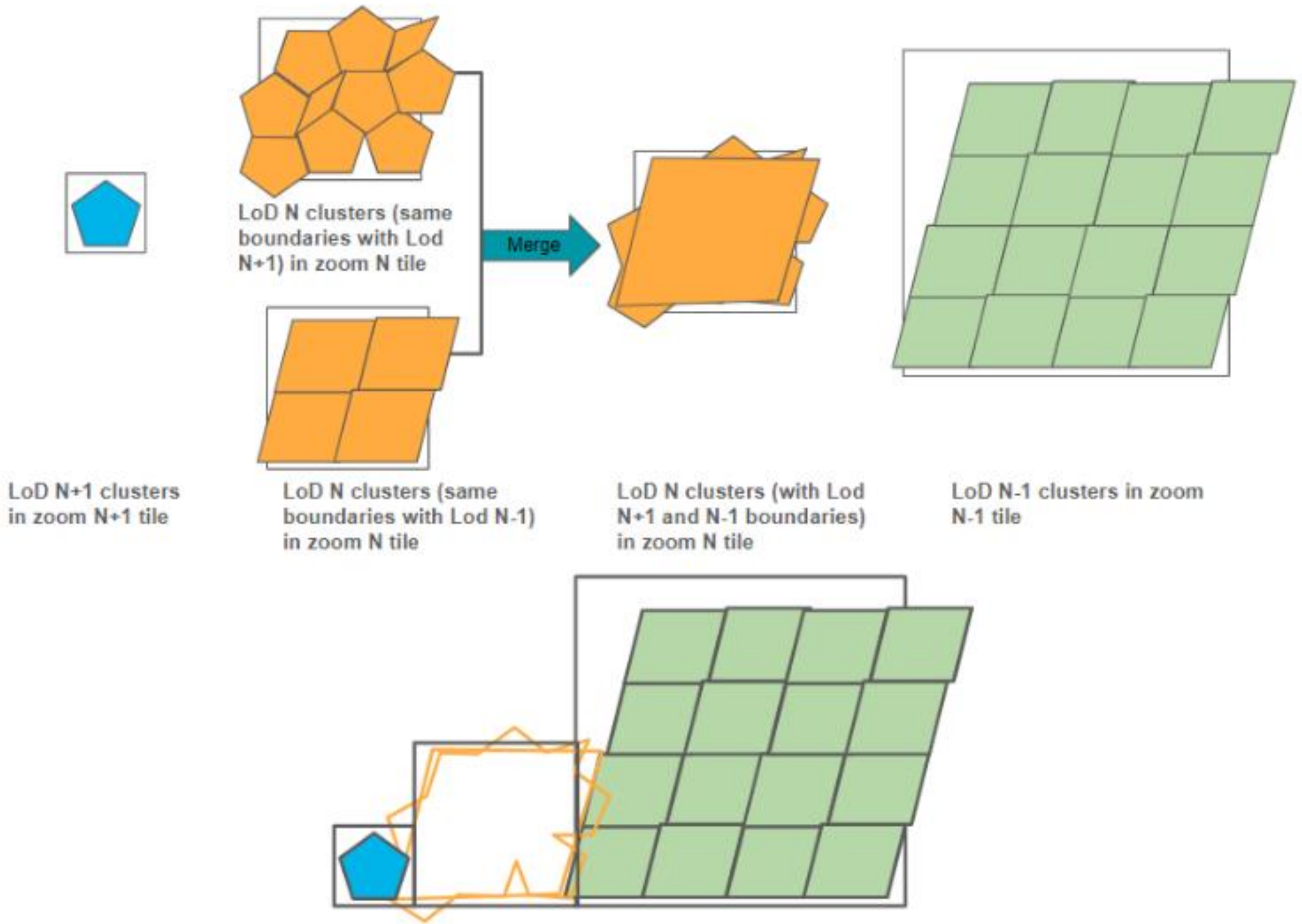


3. Methodology

Assign clusters to tiles and generate tiles – Boundaries Merge

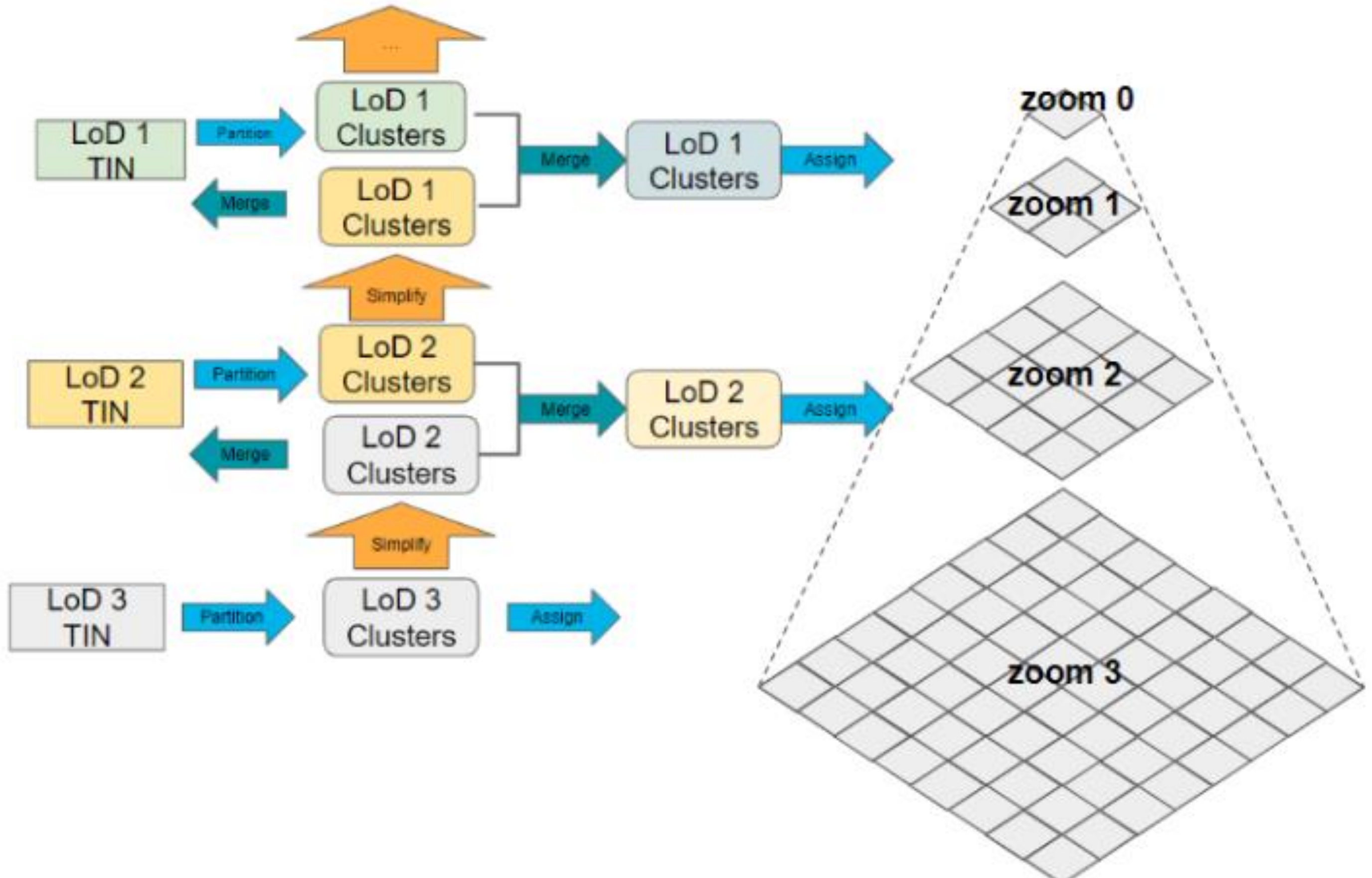



🟨 Clusters share boundaries with zoom 0
🟦 Clusters share boundaries with zoom 2


LoD N clusters (same boundaries with Lod N+1) in zoom N tile
Merge
LoD N+1 clusters in zoom N+1 tile
LoD N clusters (same boundaries with Lod N-1) in zoom N tile
LoD N clusters (with Lod N+1 and N-1 boundaries) in zoom N tile
LoD N-1 clusters in zoom N-1 tile

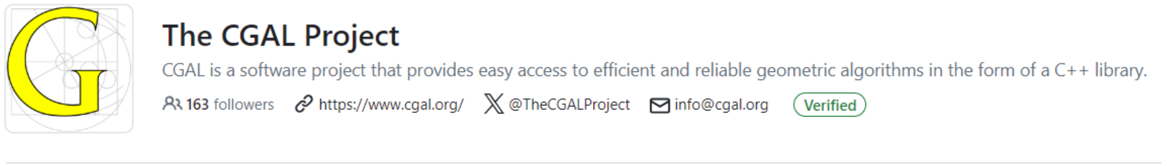
3. Methodology

Assign clusters to tiles and generate tiles – Boundaries Merge



4. Implementation

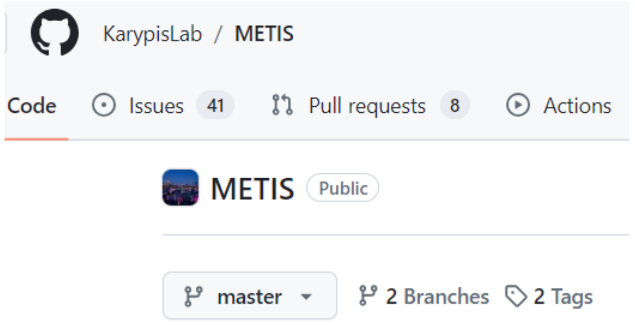
- A prototype is developed using C++ as the programming language.
- The implementation utilizes the CGAL library (cga, 2021) and METIS (Karypis and Kumar, 1999).
- To assess the quality of models after simplification, Python is utilized in conjunction with the Rasterio library (Gillies et al., 2013–2023) to compute the height error of the models.



The CGAL Project

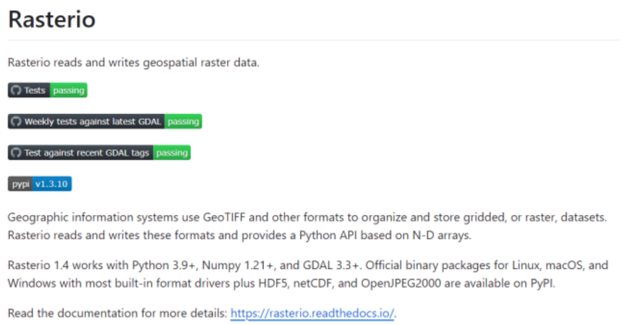
CGAL is a software project that provides easy access to efficient and reliable geometric algorithms in the form of a C++ library.

163 followers https://www.cgal.org/ @TheCGALProject info@cgal.org Verified



KarypisLab / METIS

METIS Public



Rasterio

Rasterio reads and writes geospatial raster data.

Tests passing
Weekly tests against latest GDAL passing
Test against recent GDAL tags passing
pypi v1.3.10

Geographic information systems use GeoTIFF and other formats to organize and store gridded, or raster, datasets. Rasterio reads and writes these formats and provides a Python API based on N-D arrays.

Rasterio 1.4 works with Python 3.9+, Numpy 1.21+, and GDAL 3.3+. Official binary packages for Linux, macOS, and Windows with most built-in format drivers plus HDF5, netCDF, and OpenJPEG2000 are available on PyPI.

Read the documentation for more details: https://rasterio.readthedocs.io/.

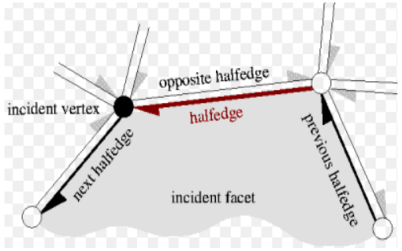
4. Implementation – Data Formats

- **DEM data**
  - The original DEM data is provided in GeoTIFF format, representing a height map model. This DEM is transformed into a TIN using TIN-Terrain (HERE Technologies, 2024).

- **TIN data**
  - The 3D TIN models in this thesis are stored in the Object File Format (OFF). OFF format is chosen for its simple data structure, similar to Wavefront .obj (OBJ) format, but with more official standardization.

4. Implementation – Data Structures

- **TIN Mesh: CGAL::Surface mesh**
  This structure is a class from CGAL (cga, 2021) specifically designed for representing 3D meshes. It can be utilized as a **halfedge** (Kettner, 2024) data structure or a polyhedral surface. Note that a halfedge data structure is an edge-centered data structure capable of **maintaining incidence information** of vertices, edges, and faces.



- **Partition Information: Face id map**
  - Property map (provides functions for accessing a reference to a value object.)

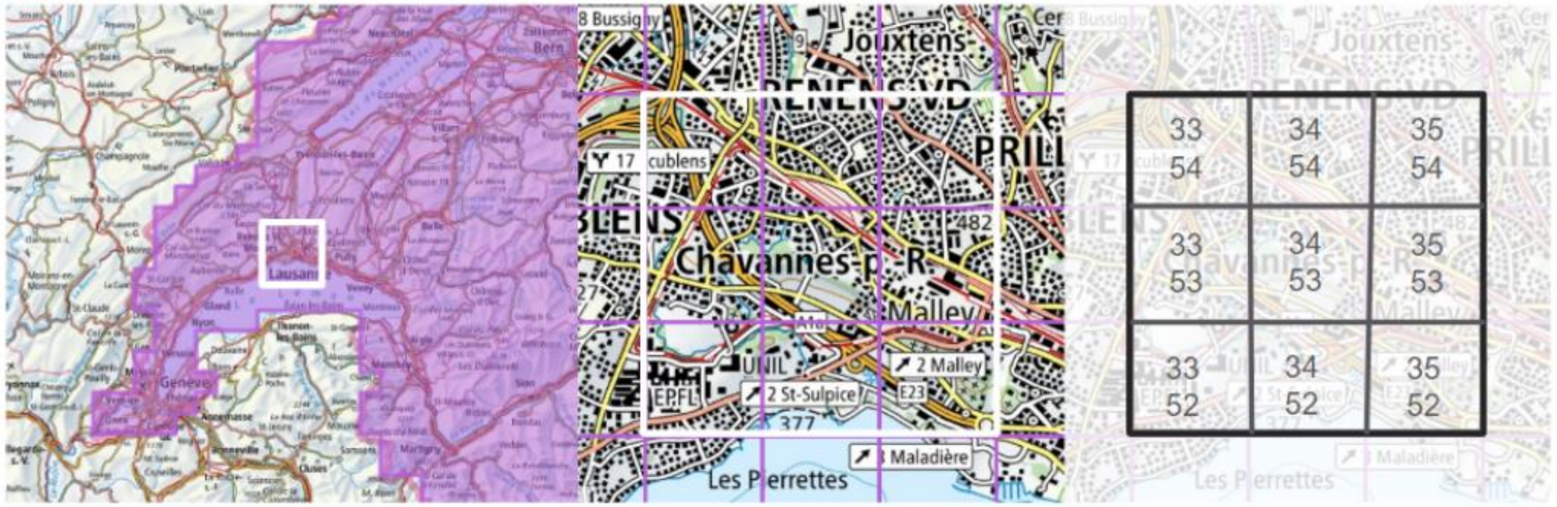
- **Constrained Elements:**
  - ***is constrained edge map*** It's a struct inherited from the Property map. It is composed of a boolean value to indicate whether an edge is constrained and a reference to the edge.

- **halfedge descriptor**
  - It is an identifier defined by graph traits (Libraries, 2021). graph traits provides associated types for the graph, and halfedge descriptor is used to identify a halfedge.

5. Result and Analysis

Datasets

The experiment data selected for this thesis consists of 9 tiles covering a 3 km × 3 km area in Lausanne, Switzerland from swissALTI3D. This area is located in the center of Lausanne city, and is an urban environment with multiple roads and significant topological variation (height differences). The maximum height in the area is **522.83** meters, while the minimum height is **367.68** meters.

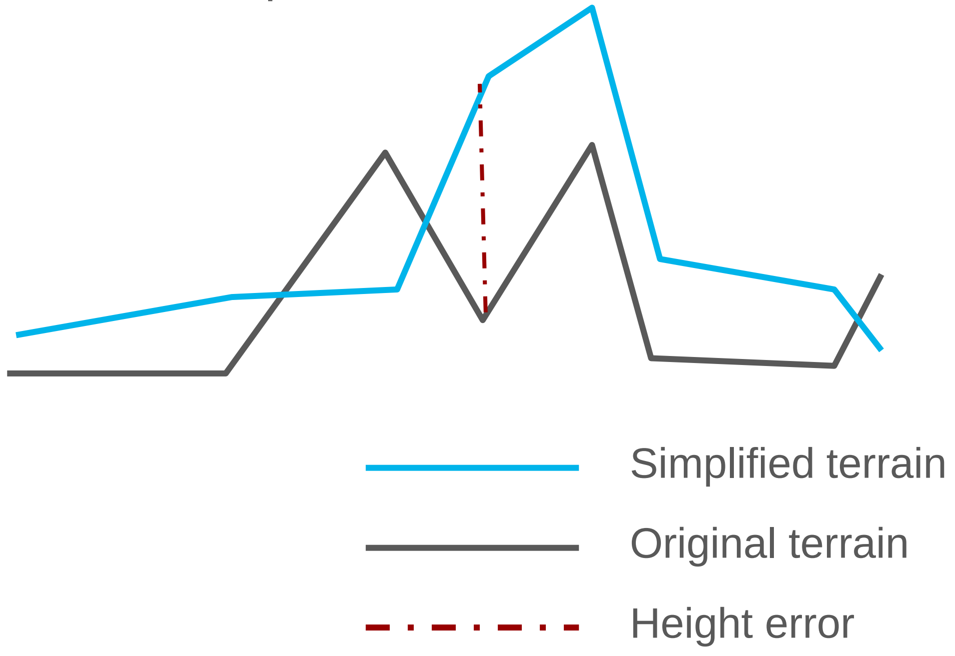


5. Result and Analysis – Evaluation

Simplification result

Simplification error, visualization result, constraints preservation.

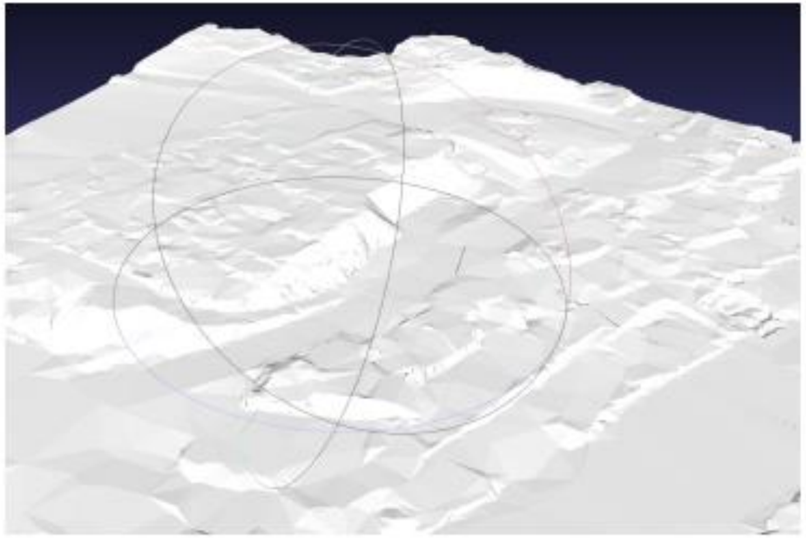
The evaluation of the simplification result is based on the maximum height errors between the simplified TIN models and the original models.



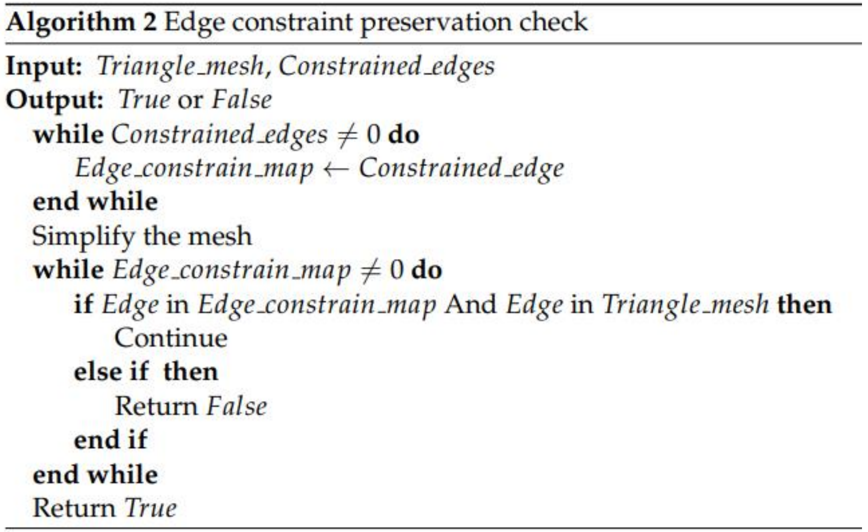
5. Result and Analysis – Evaluation

Simplification result

Visualization: Seamless? visualization effect?



Constraints preservation



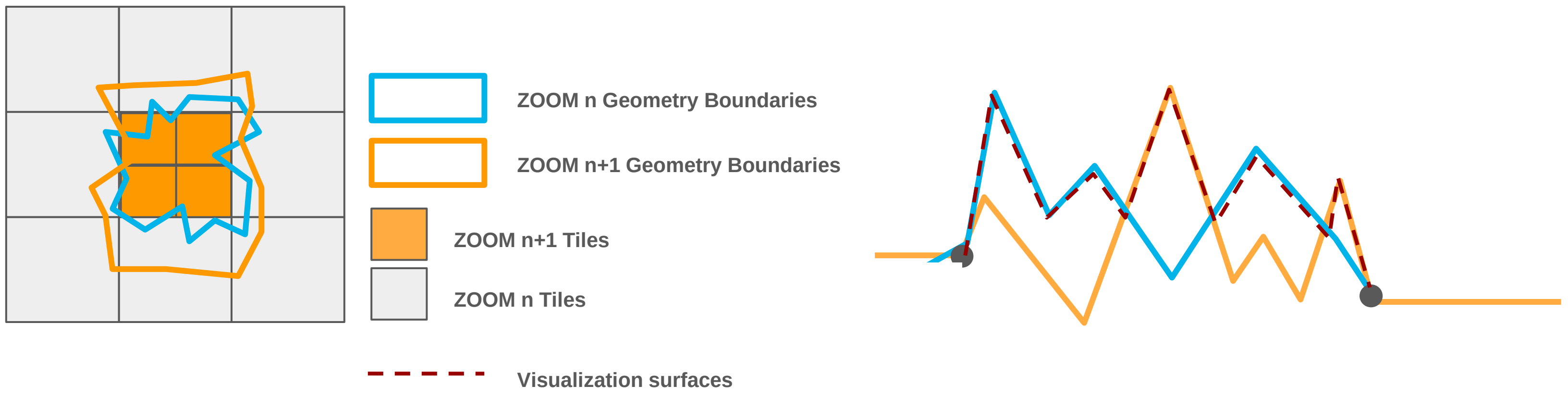
Algorithm 2 Edge constraint preservation check

```
Input: $Triangle\_mesh, Constrained\_edges$
Output: $True$ or $False$
  while $Constrained\_edges \neq 0$ do
      $Edge\_constrain\_map \leftarrow Constrained\_edge$
  end while
  Simplify the mesh
  while $Edge\_constrain\_map \neq 0$ do
      if $Edge$ in $Edge\_constrain\_map$ And $Edge$ in $Triangle\_mesh$ then
          Continue
      else if then
          Return $False$
      end if
  end while
  Return $True$
```

5. Result and Analysis – Evaluation

Simplification result – Visualization, seamless connection.

If the blue boundaries vertices can be found in zoom n+1, and the yellow boundary vertices can be found in zoom n, then it means they can be connected seamlessly.

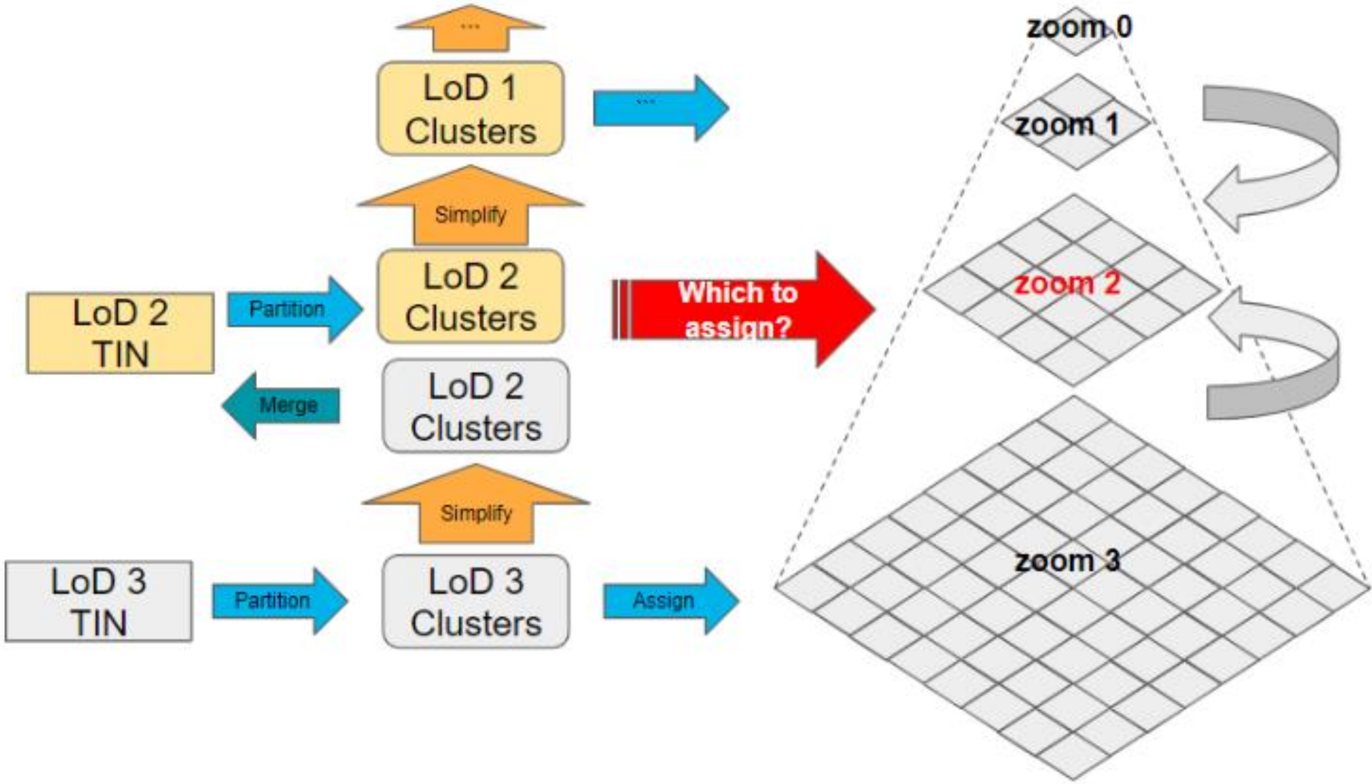


5. Result and Analysis

Construction result

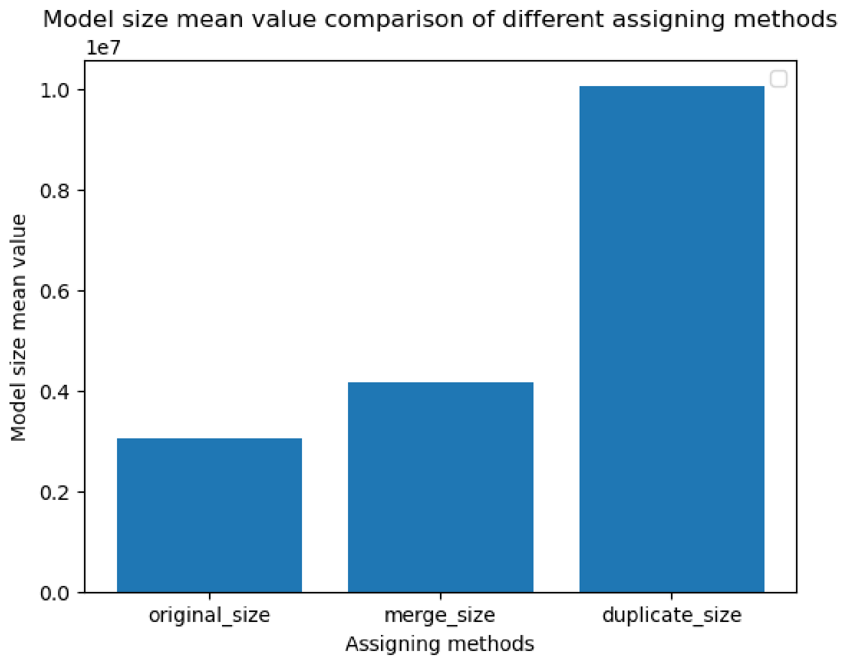
The experiment area in Lausanne generates **576** zoom 3 tiles, with 24 rows and 24 columns, **144** zoom 2 tiles, with 12 rows and 12 columns, **36** zoom 1 tiles, with 6 rows and 6 columns, and **9** zoom 0 tiles. Each tile is identified with index zoom level/row/column.

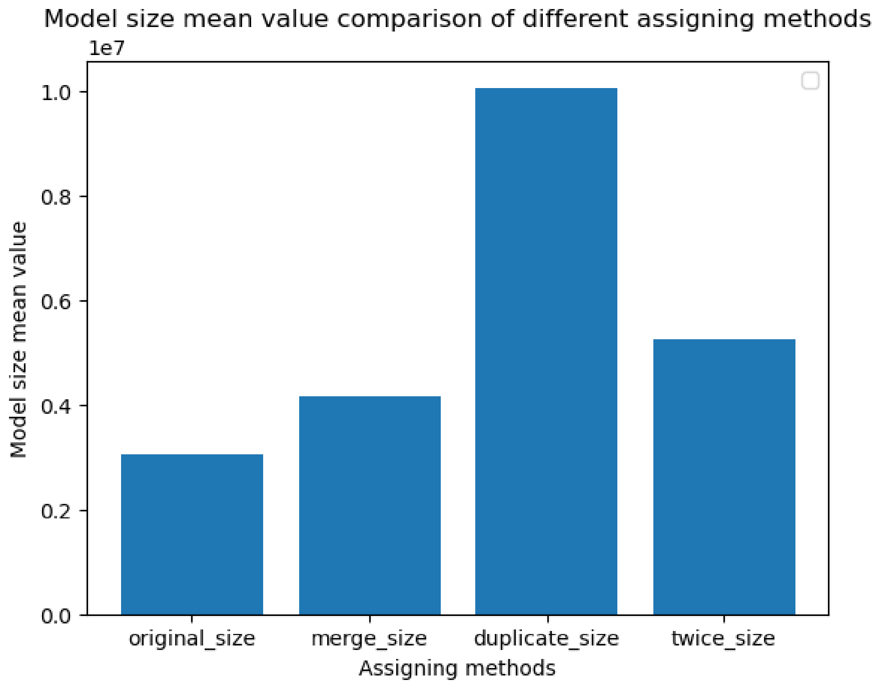
For example, the tile in zoom 2, row 0, column 0 is Tile 2/0/0.



5. Result and Analysis

Assign clusters to tiles and generate tiles – Redundancy evaluation

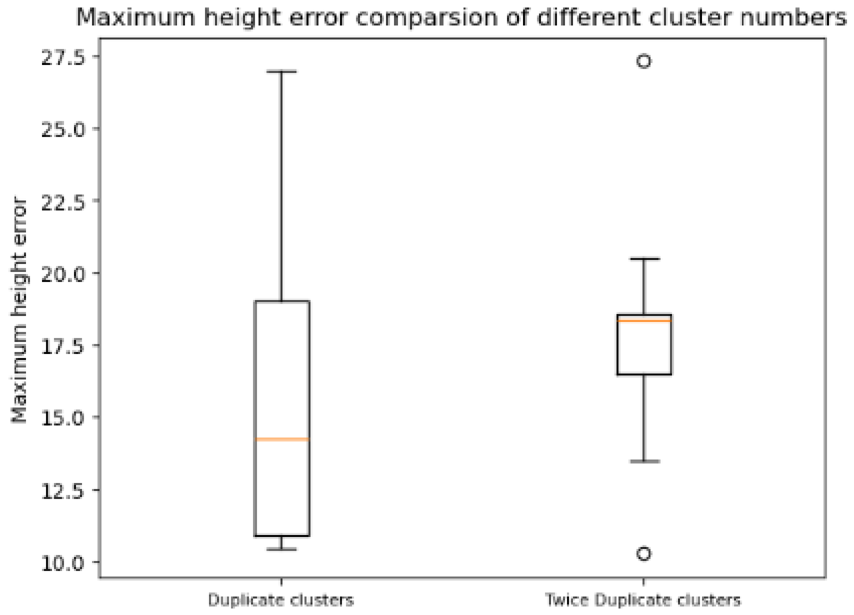




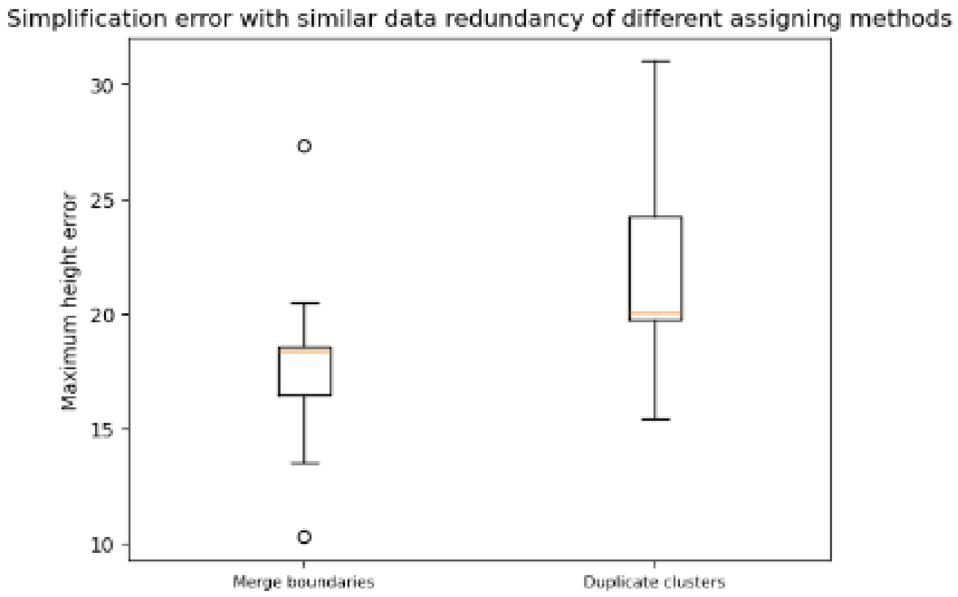
5. Result and Analysis

Assign clusters to tiles and generate tiles – Redundancy evaluation

However, with twice clusters number, the error rises:



And when the redundancy sizes are similar, the merge boundaries error is smaller than duplicate clusters.

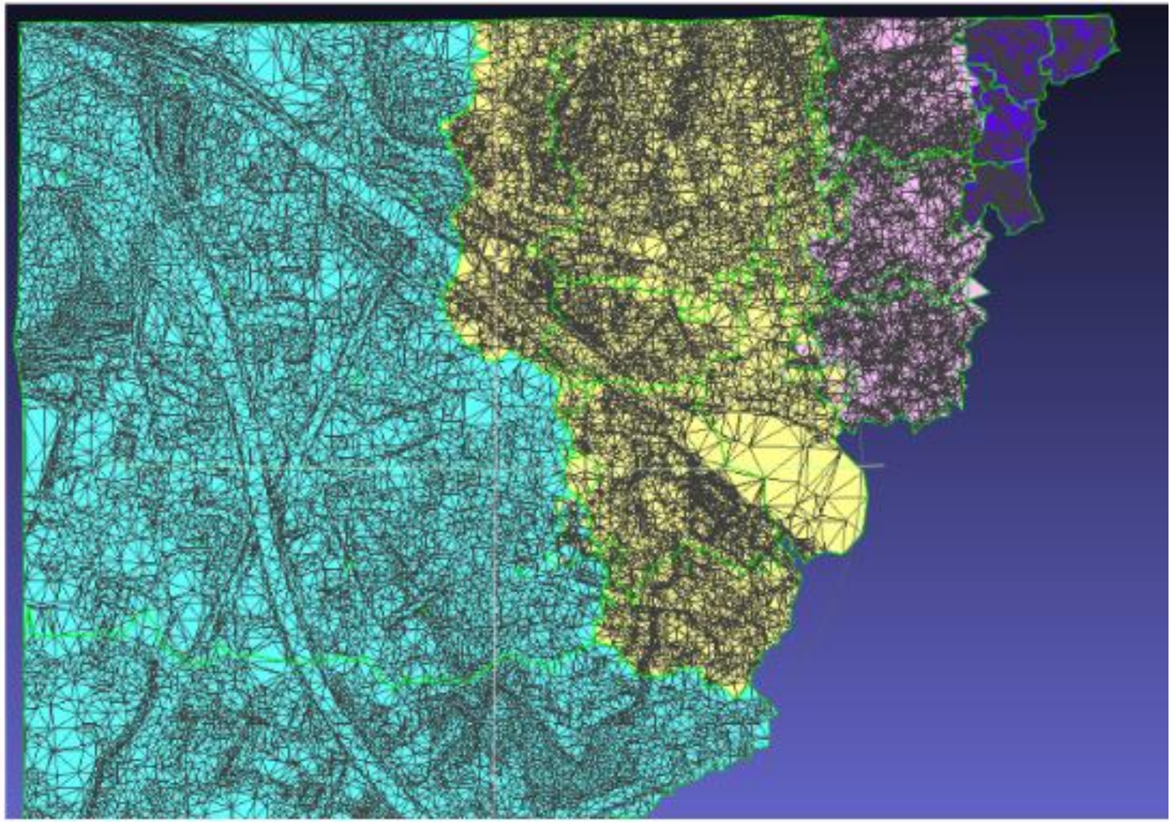


5. Result and Analysis

Construction result

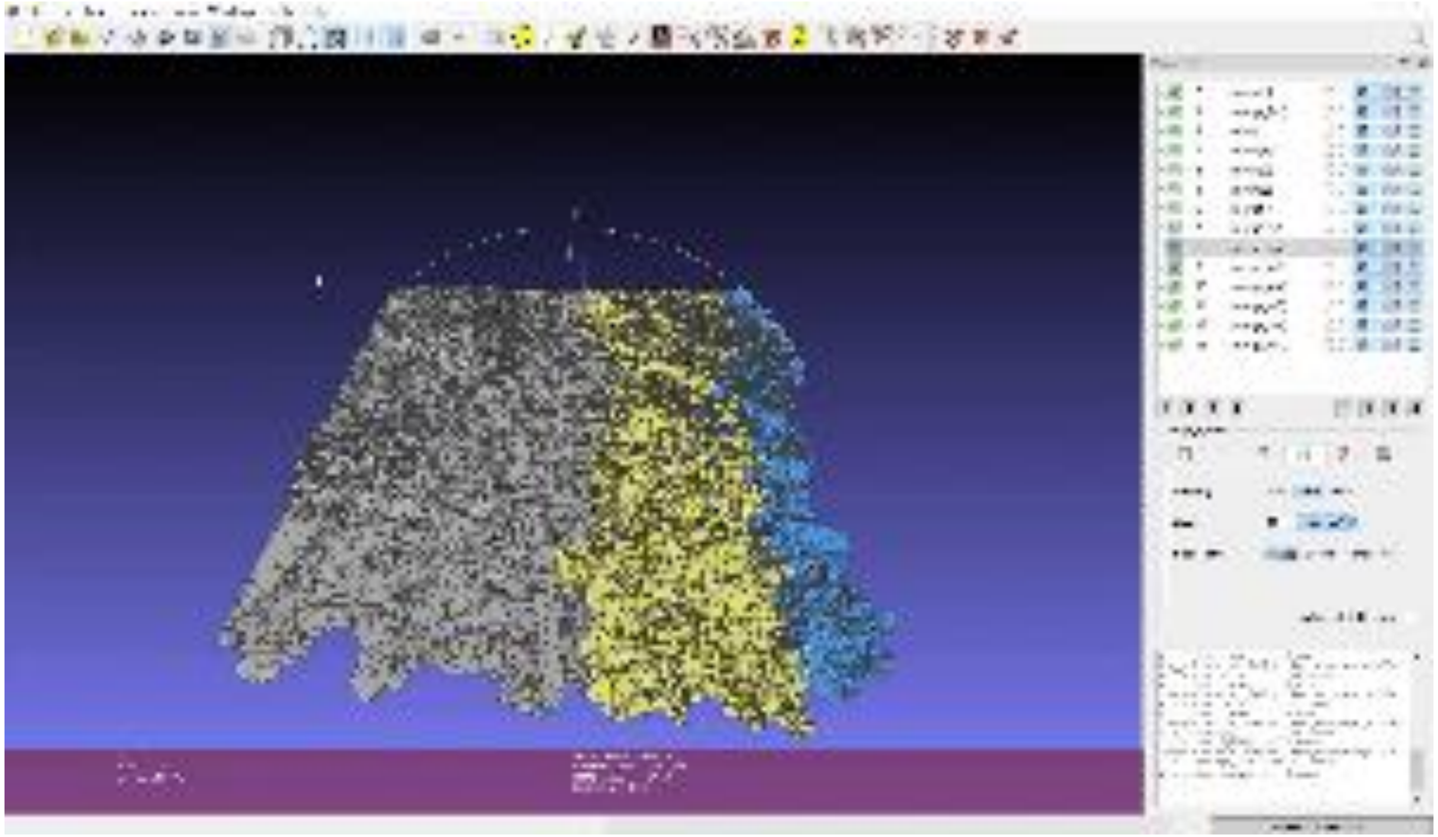
The connection between zoom 2, zoom 1, and zoom 0 tiles is visualized. The result shows that the tiles can be joined seamlessly together in visualization.

2 tile in zoom 0 (Tile 0/0/0, Tile 0/0/1), 2 tiles in zoom 1 (Tile 1/0/2, Tile 1/1/2), and 3 tiles in zoom 2 (Tile 2/0/6 - 2/2/6), 4 tiles in zoom 3 (Tile 3/0/13 - Tile 3/2/13, Tile 3/0/14) are joined togethe



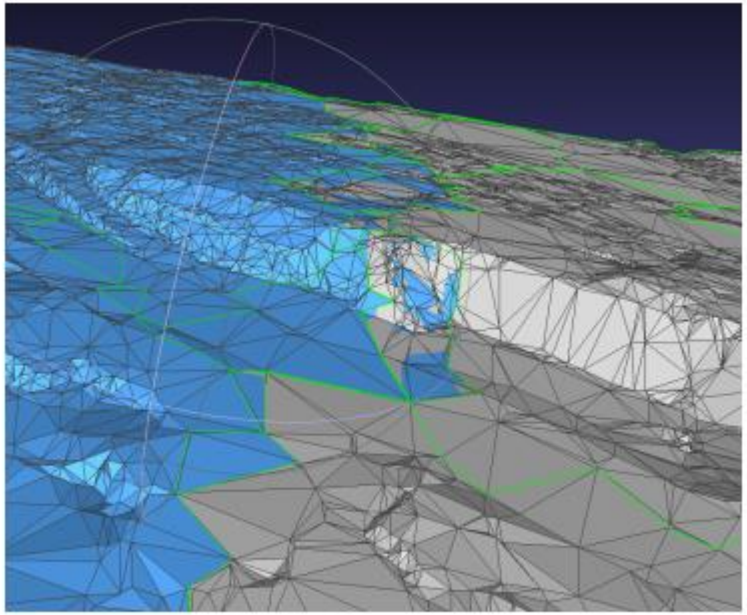
5. Result and Analysis

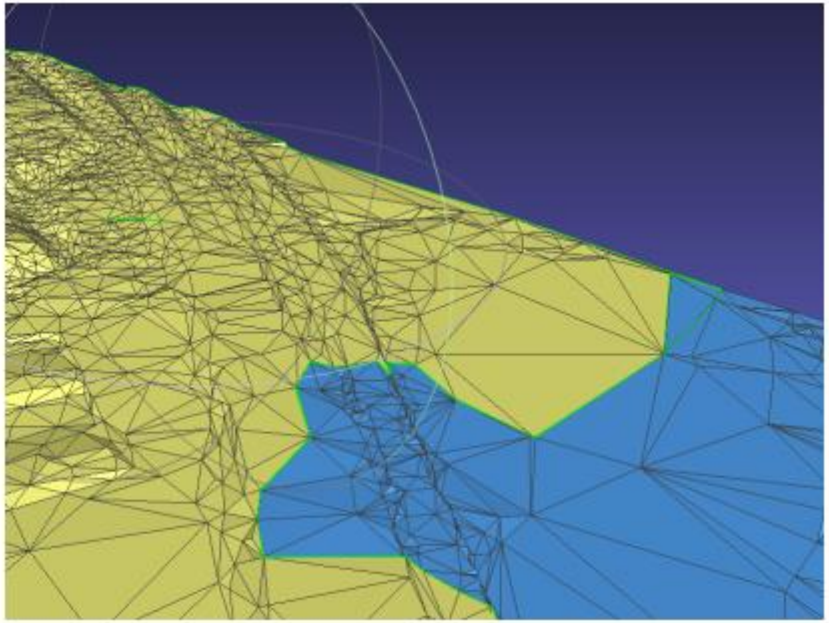
Construction result



5. Result and Analysis

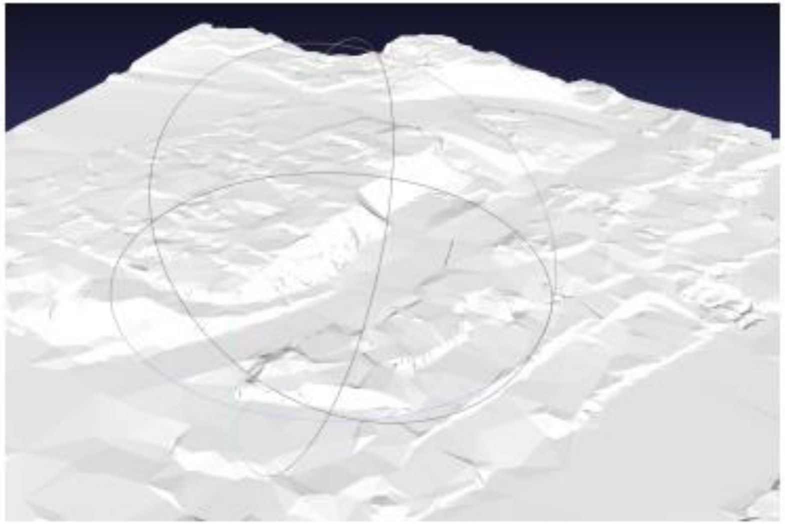
Construction result

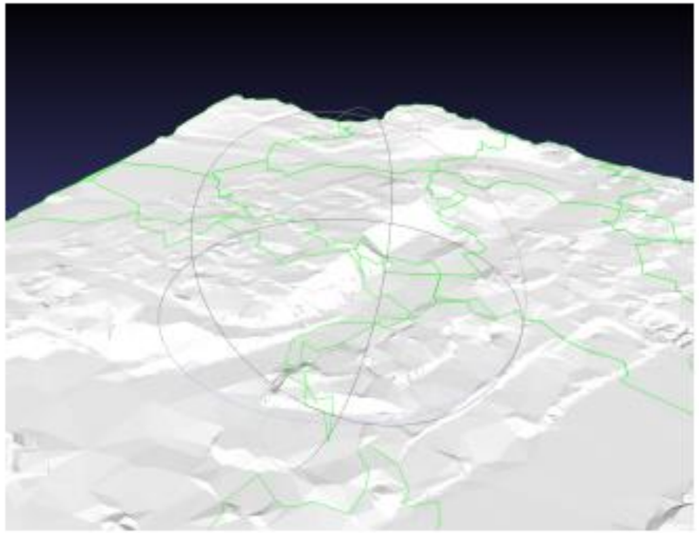

(a) Boundaries connection between zoom 2 and zoom 1


(b) Boundaries connection between zoom 0 and zoom 1

5. Result and Analysis

Construction result


(a) Seamless connection in overlap area

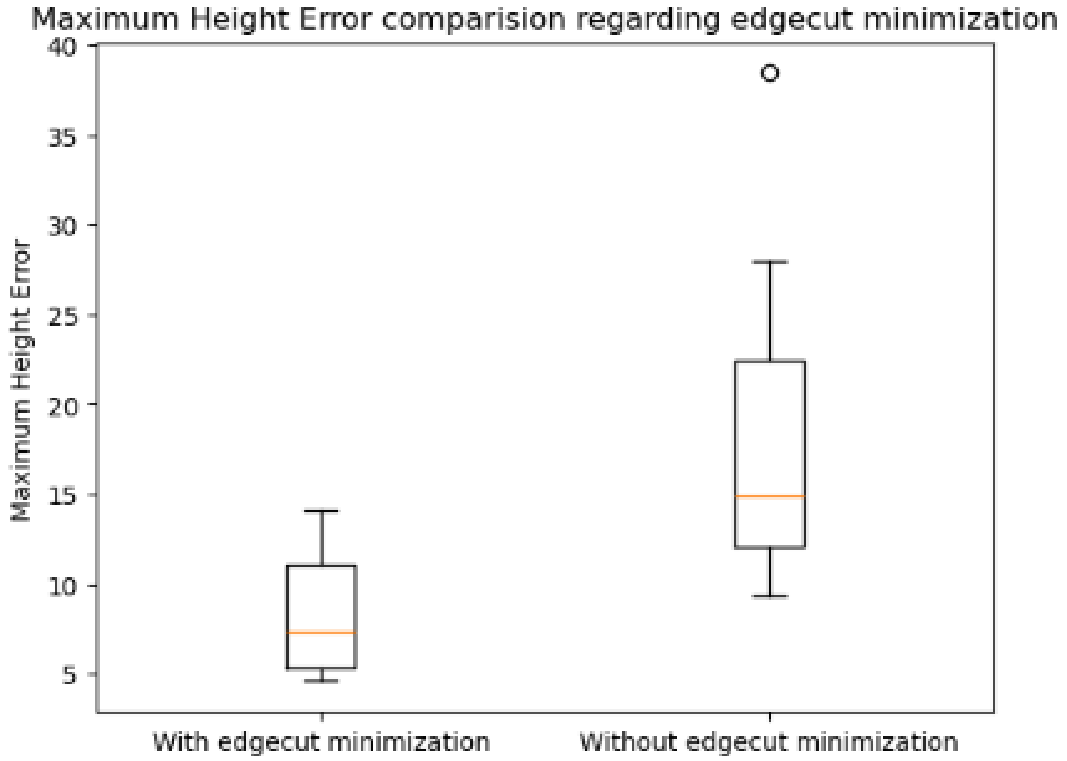

(b) Boundaries of the overlap areas

5. Result and Analysis — Triangle clustering

Simplification effciency – Edgecut minimization

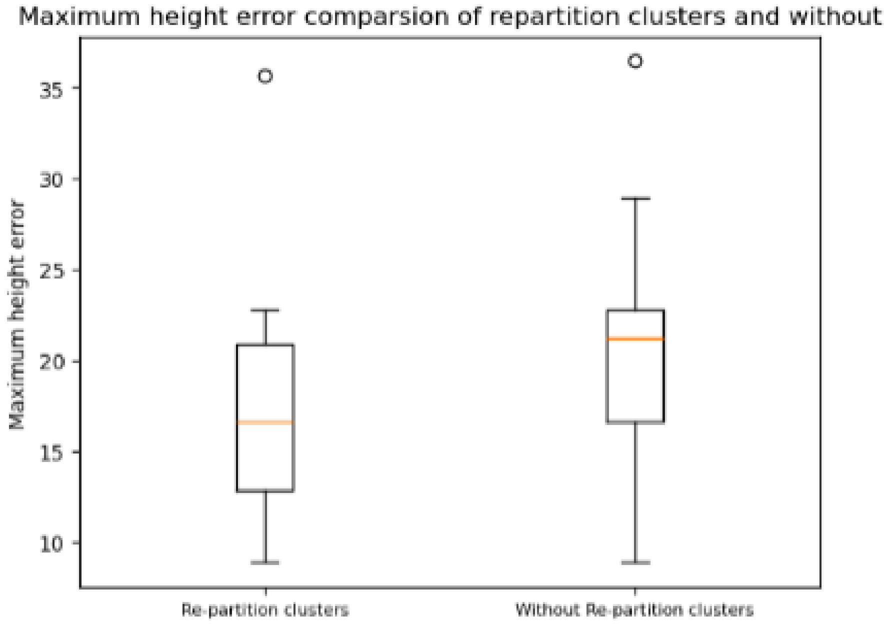
Edge Cut Minimization:

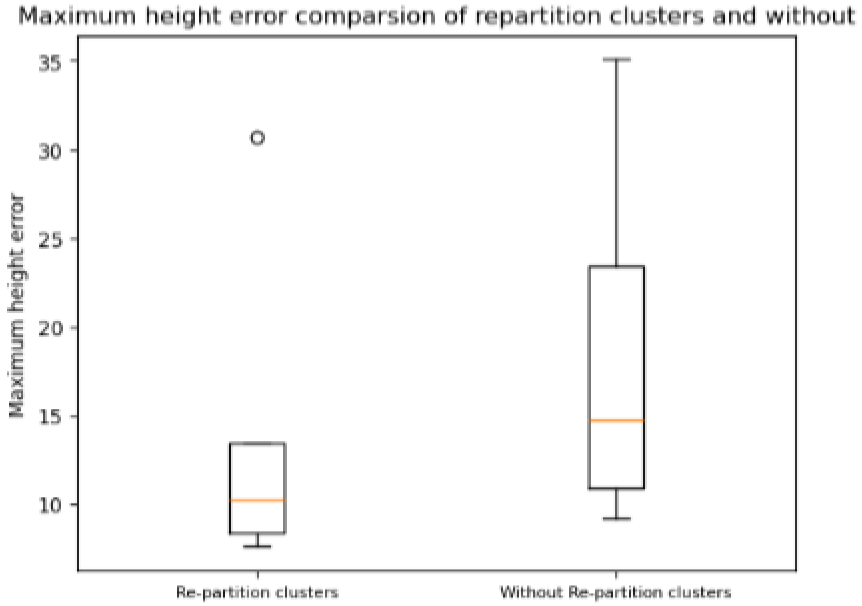
Have better simplificaition result.



5. Result and Analysis – Triangle clustering analysis

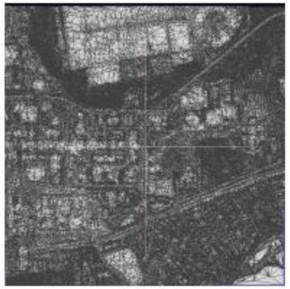
Simplification effciency – Repartition

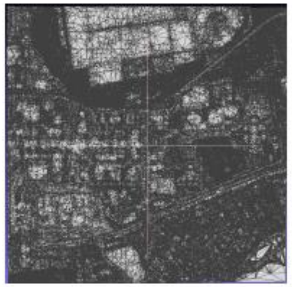


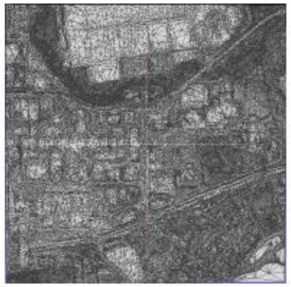


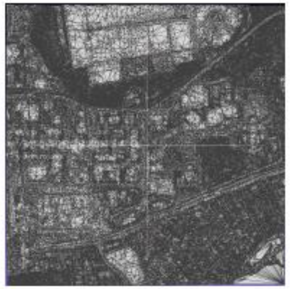
5. Result and Analysis – Triangle clustering analysis

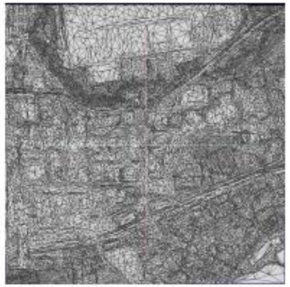
Complicated edges improvement

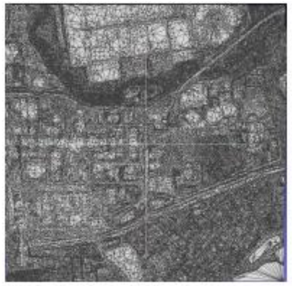

(a). Unchanged locked boundary simplification 1

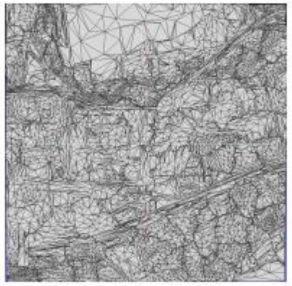

(b). Changed locked boundary simplification 1

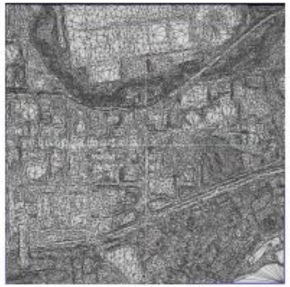

(c). Unchanged locked boundary simplification 2

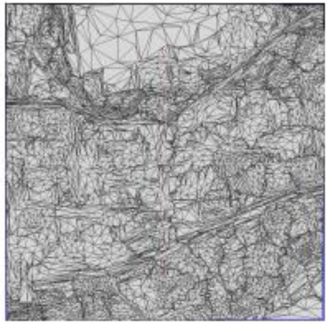

(d). Changed locked boundary simplification 2

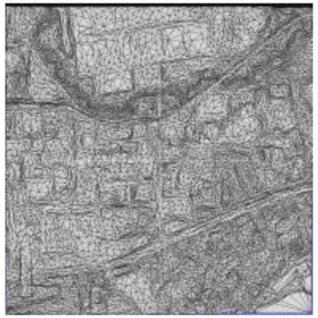

(e) Unchanged locked boundary simplification 3


(f) Changed locked boundary simplification 3


(g) Unchanged locked boundary simplification 4

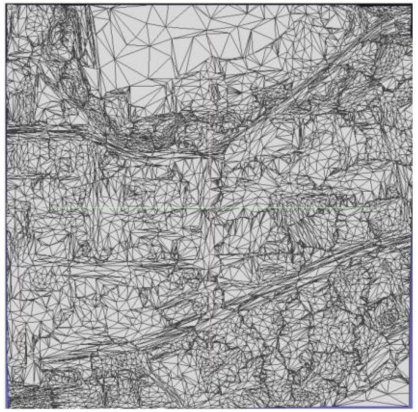
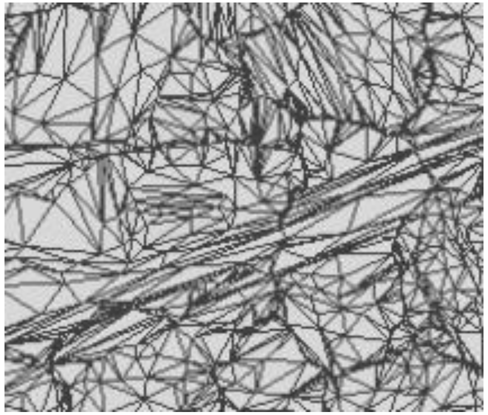

(h) Changed locked boundary simplification 4


(i) Unchanged locked boundary simplification 5

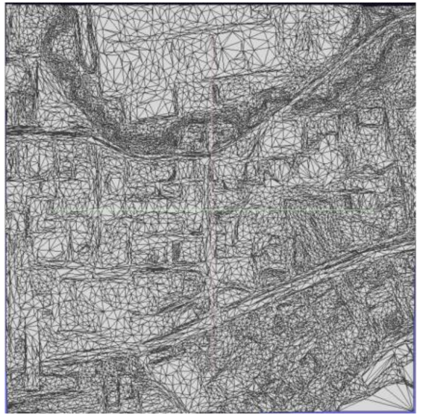
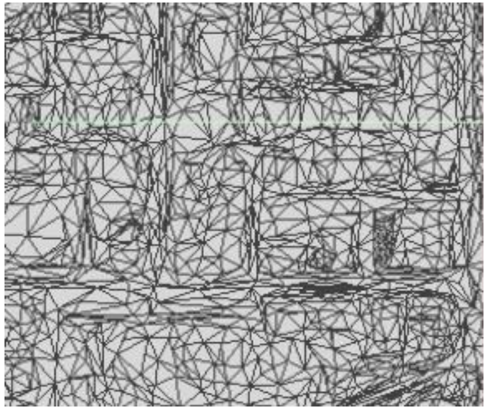

(j) Changed locked boundary simplification 5

5. Result and Analysis – Triangle clustering analysis

Complicated edges improvement

Without triangle clustering

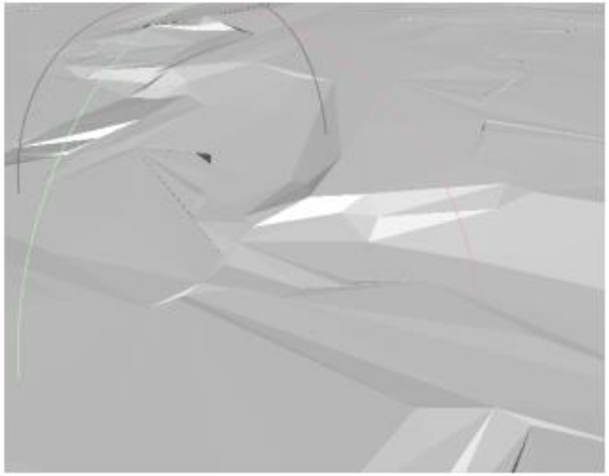



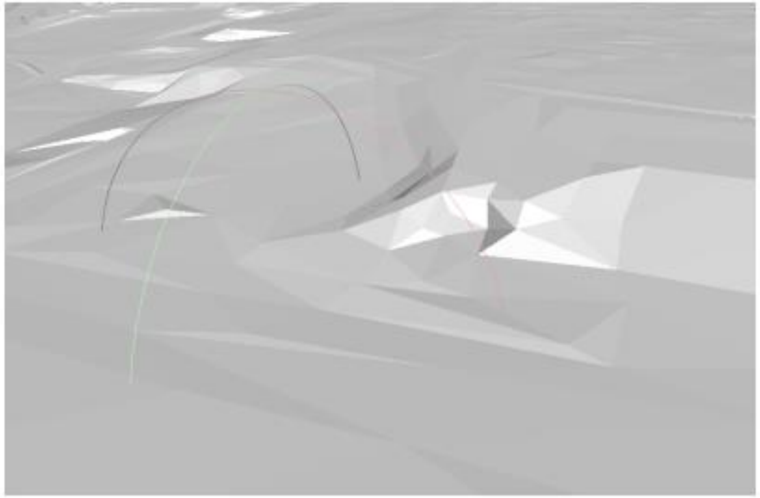
With triangle clustering

5. Result and Analysis – Triangle clustering analysis

Complicated edges improvement

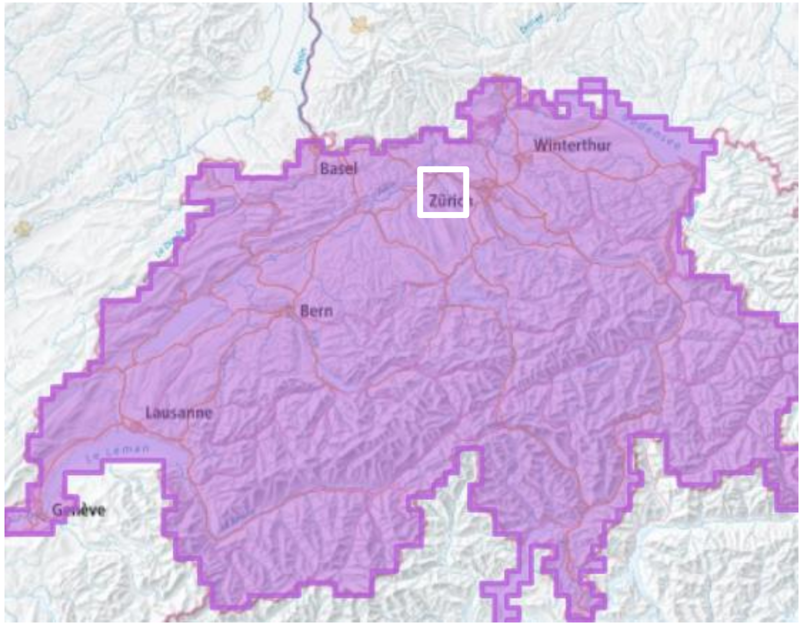
The triangle clustering simplification tend to have a smoother connection with the surrounding meshes compared to the complicated boundaries area.

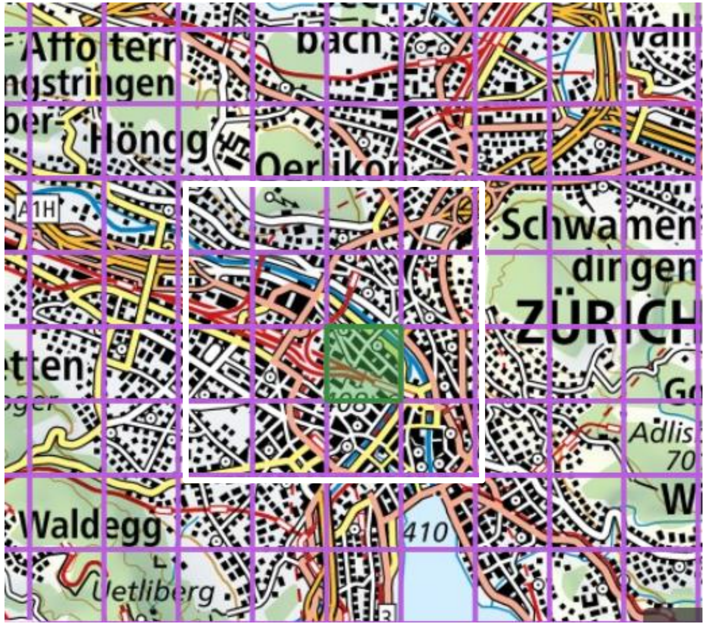

(a) Complicated boundaries artifacts


(b) Simplified boundaries artifacts

5. Result and Analysis – Triangle clustering analysis

More experience data:





Max: 583.123 meters Min: 395.948 meter

As to Zurich, the 16 DEM tiles are combined due to time limitation, the vertices number is 1322370, and the faces number is 2643550

6. Conclusion

This thesis proposes a solution to build multi-LoD large-scale terrain for web map navigation. To deliver large-scale 3D terrain in web mapping, multi-LoD tiled terrain models are constructed due to the data size.

- Simplification method:
  - Edge collapse, preserved elements conserved in constraint map

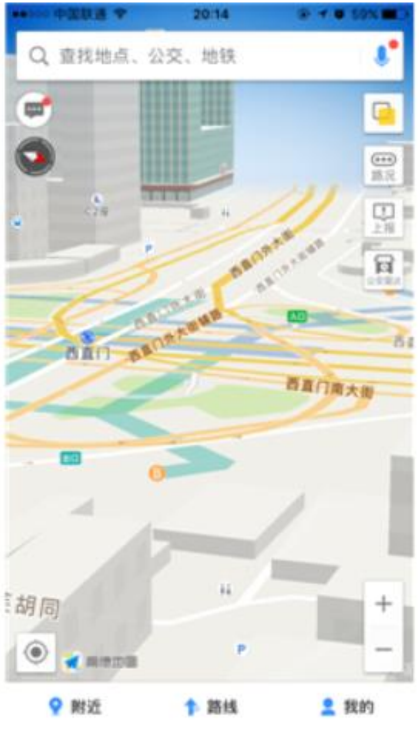
- Partition solution:
  - Edge cut minimization, can improve simplification efficiency

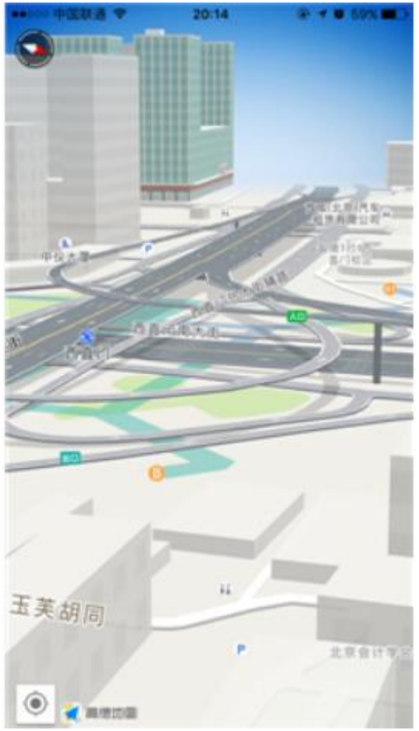
- Triangle clustering method:
  - Improve simplification efficiency and reduce visualization artifacts

- Web tiles assigning:
  - Double zoom tiles, duplicate clusters, and boundaries merge—are proposed and tested. Boundaries merge can cost 37% redundancy.

6. Conclusion – Future Work

- Involves real life roads, building footprints data in the future.
- Consider elements that are not fully attached to the terrain, e.g. flyovers, tunnels, bridges.





6. Conclusion – Future Work

- Make more test samples:
  - Make multiple zoom levels to test the connection and data redundancy.
  - Test triangle clustering effects on simplification after several simplification iteration (only 2 in the thesis and 5 on a single TIN data)
  - Experiment on various terrain conditions, China, USA, etc…

- Quantify visualization improvement of triangle clustering :
  - rasterized and calculate error per pixel.

Thank you for your listening!

Questions?