

UFR

de **mathématique**
et d'**informatique**



Université de Strasbourg

exa-MA WP1 - Vegetation

Pierre-Antoine SENGER

August 18, 2024

Introduction

This project is part of a series conducted within the **exa-MA** project, which is a segment of the **Numpex** project section.

- Exa-MA WP1 - Vegetation
- Exa-MA WP1 - Terrain
- Exa-MA WP1 - Urban Building LOD-1
- Exa-MA WP1 - Urban Building LOD-2 and Kinetic
- Exa-MA WP1 - Performance and Scalability

Introduction

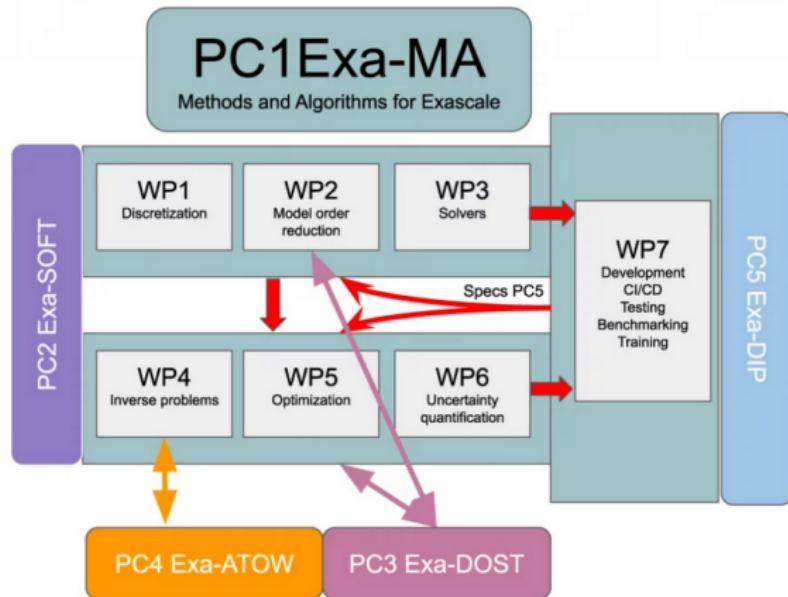


Figure: Exa-MA project overview [1]

Context

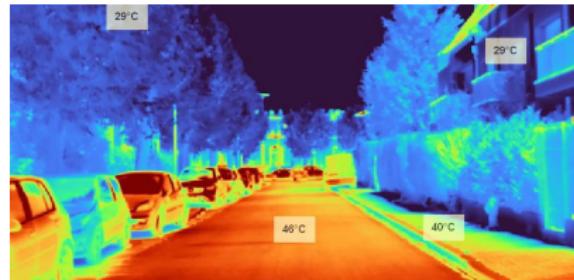


Figure: Thermal image of a street depicting heat distribution [2]

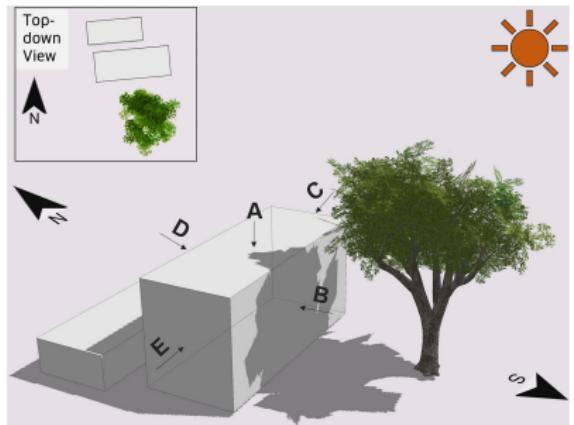


Figure: Tree providing shade to a building [3]

Context: Primiray focus

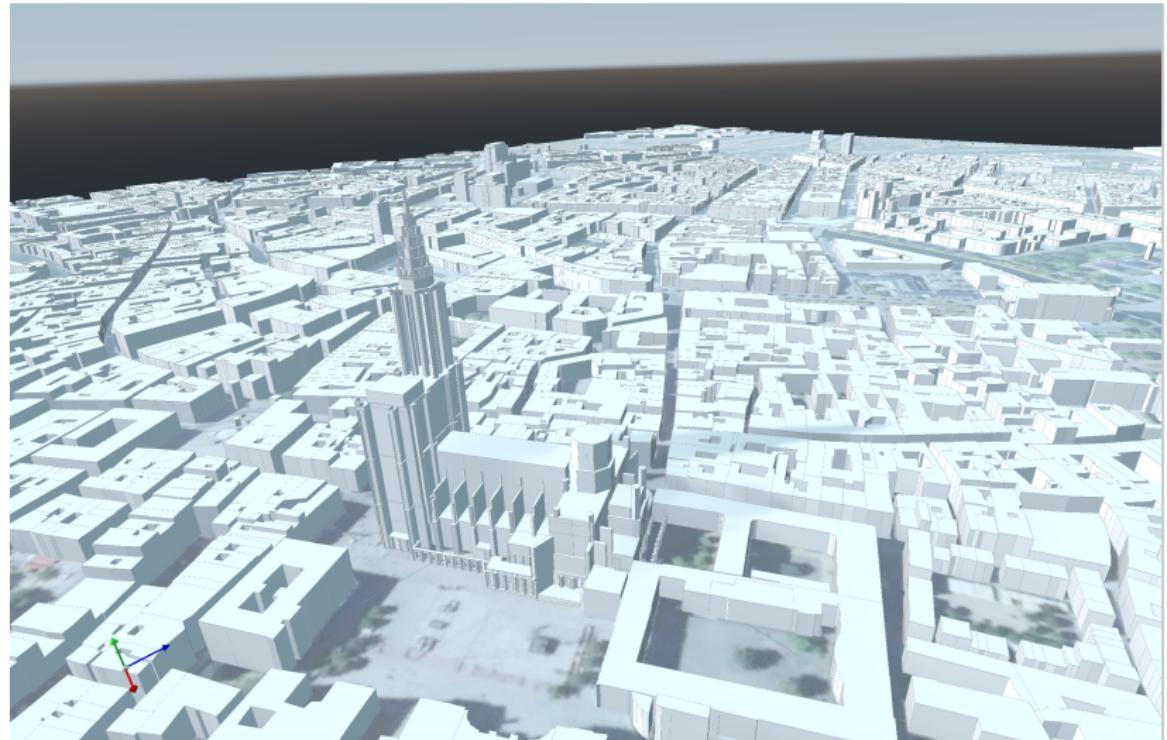


Figure: 3D Model of Strasbourg, France

Context: Adaptability

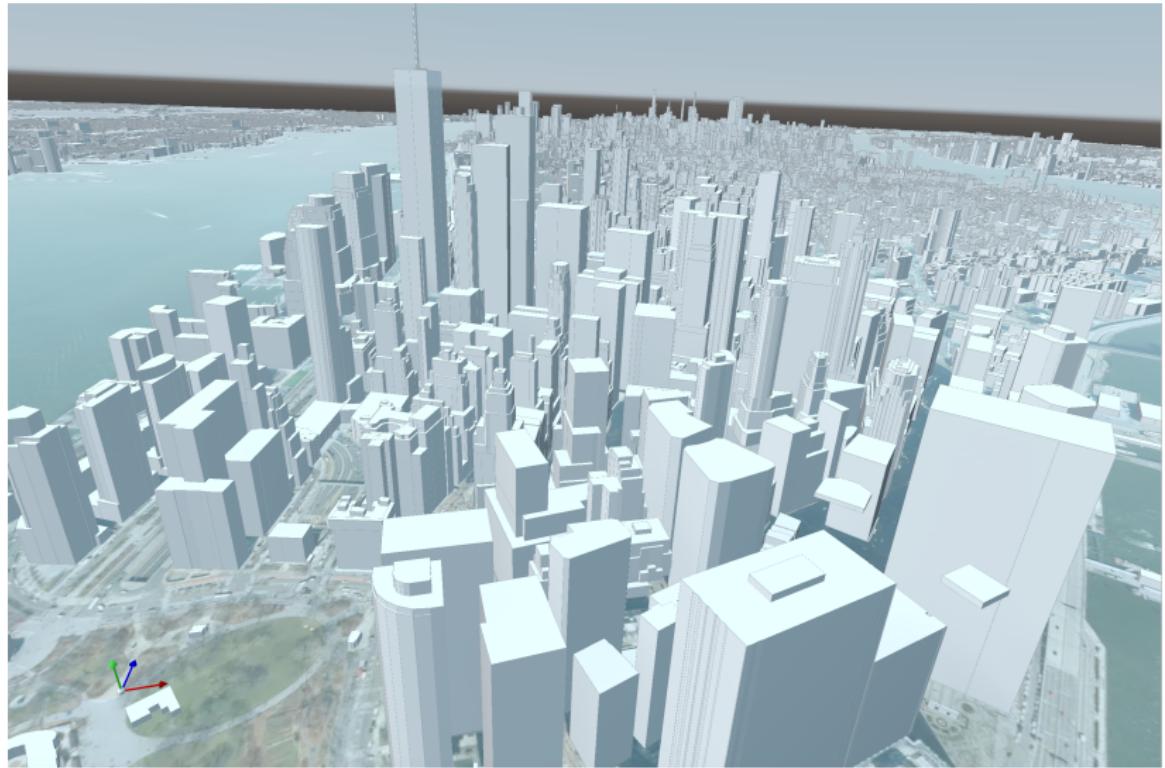


Figure: 3D Model of Manhattan, New York [4]

Objectives

- Extracting tree data from OpenStreetMap
- Generating 3D tree models
- Integrating tree models in the terrain mesh
- Optimizing computational efficiency

Methodology Steps

- Data Acquisition
- Generating Tree Library
- Scaling Trees
- Placing Trees
- Merging Meshes
- Parallelization

Data Acquisition



Figure: OpenStreetMap Logo



Figure: Curl Logo

Data Acquisition: Query Class

```
1 void perform_query(std::string bbox, bool verbose) {
2     std::string query =
3         "[out:json]; (node(\" + bbox + \")["natural\"]="tree
4         \"]); out;";
5     cpr::Response r = cpr::Post(
6         cpr::Url{"http://overpass-api.de/api/interpreter"},
7         cpr::Body{query},
8         cpr::Header{{"Content-Type", "application/x-www-form
9         -urlencoded"}},
10        cpr::Timeout{10000} // Set a timeout of 10 seconds
11    );
12 }
```

Data acquisition: .json output

```
1  {
2      "type": "node",
3      "id": 10162018740,
4      "lat": 48.5850910,
5      "lon": 7.7502624,
6      "tags": {
7          "circumference": "1.47655",
8          "diameter_crown": "5",
9          "genus": "Platanus",
10         "height": "6",
11         "leaf_cycle": "deciduous",
12         "leaf_type": "broadleaved",
13         "natural": "tree",
14         "ref": "16401",
15         "source": "data.strasbourg.eu - patrimoine_arbore",
16         "source:date": "2022-01-02",
17         "species": "Platanus acerifolia x",
18         "species:wikidata": "Q24853030"
19     }
20 }
21 }
```

Generating Tree Library

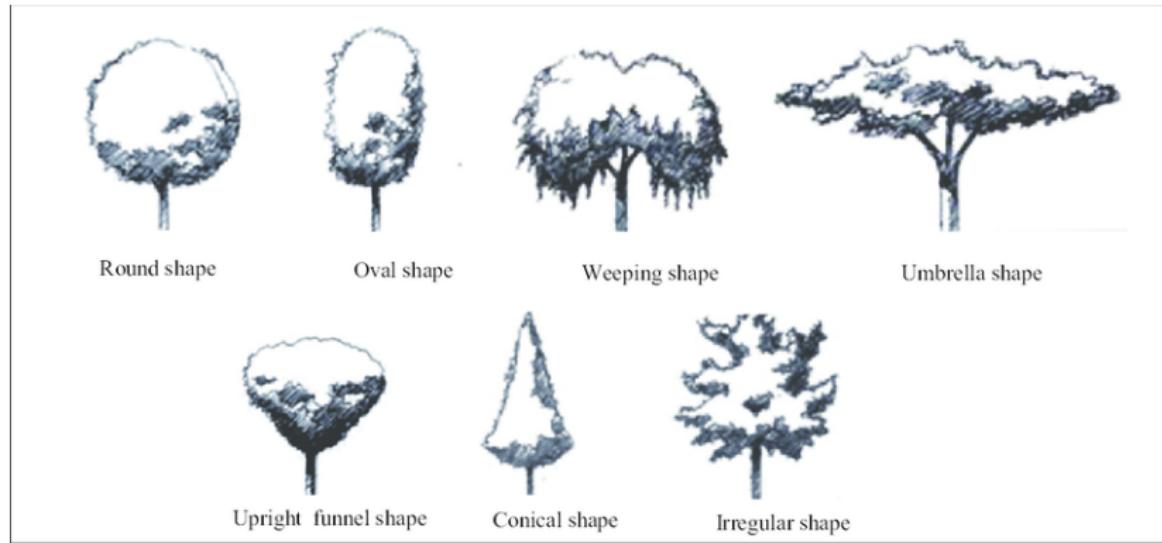


Figure: Different tree shapes

Tree Modeling: lod 0

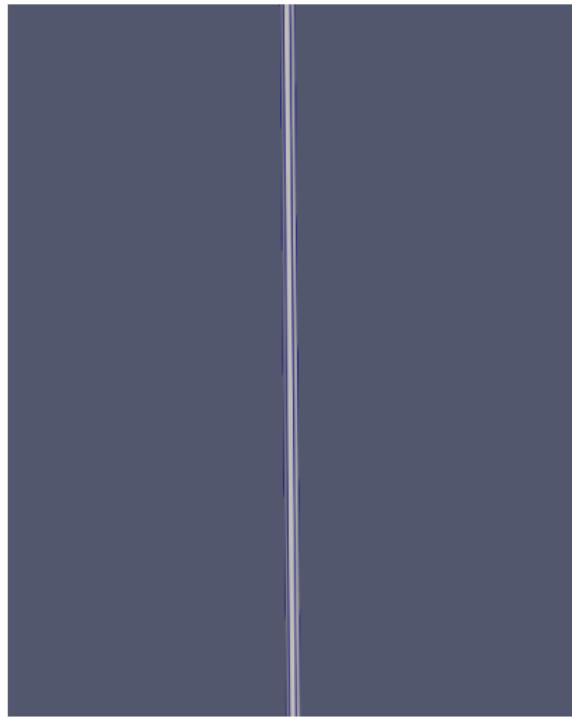


Figure: Tree Trunk model

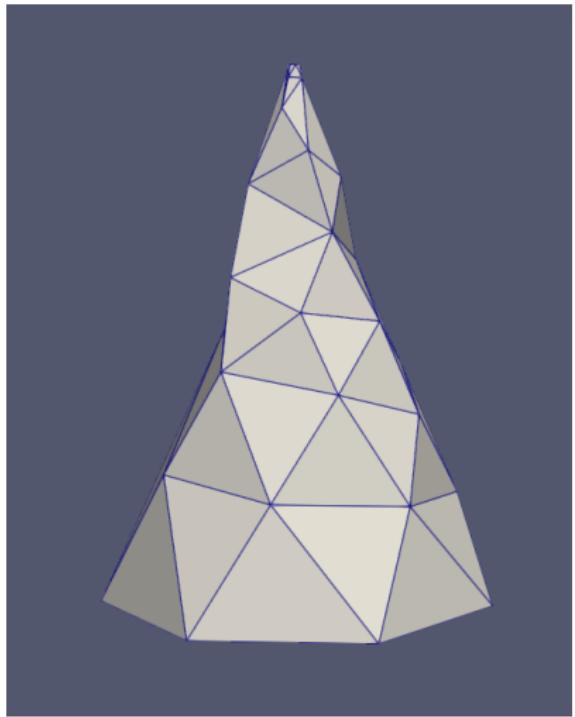


Figure: Cone shaped Tree model

Tree Modeling: lod 0

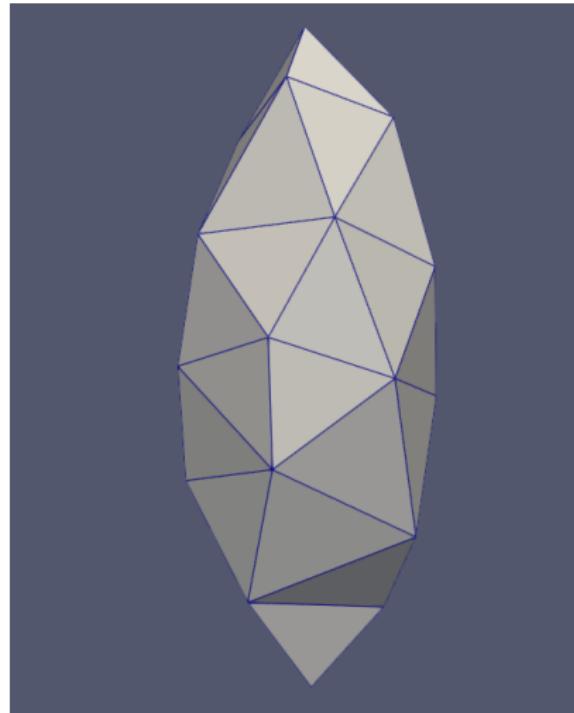


Figure: Oval shaped Tree model

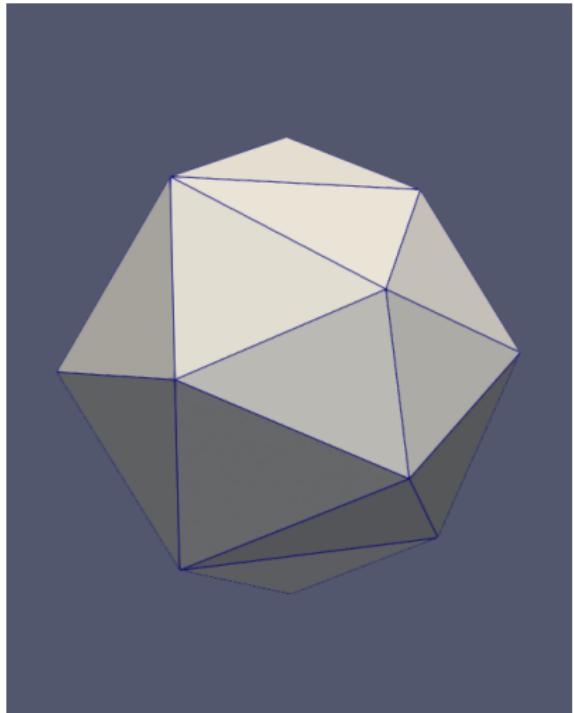


Figure: Round shaped Tree model

Tree Modeling: lod 1,2 and 3

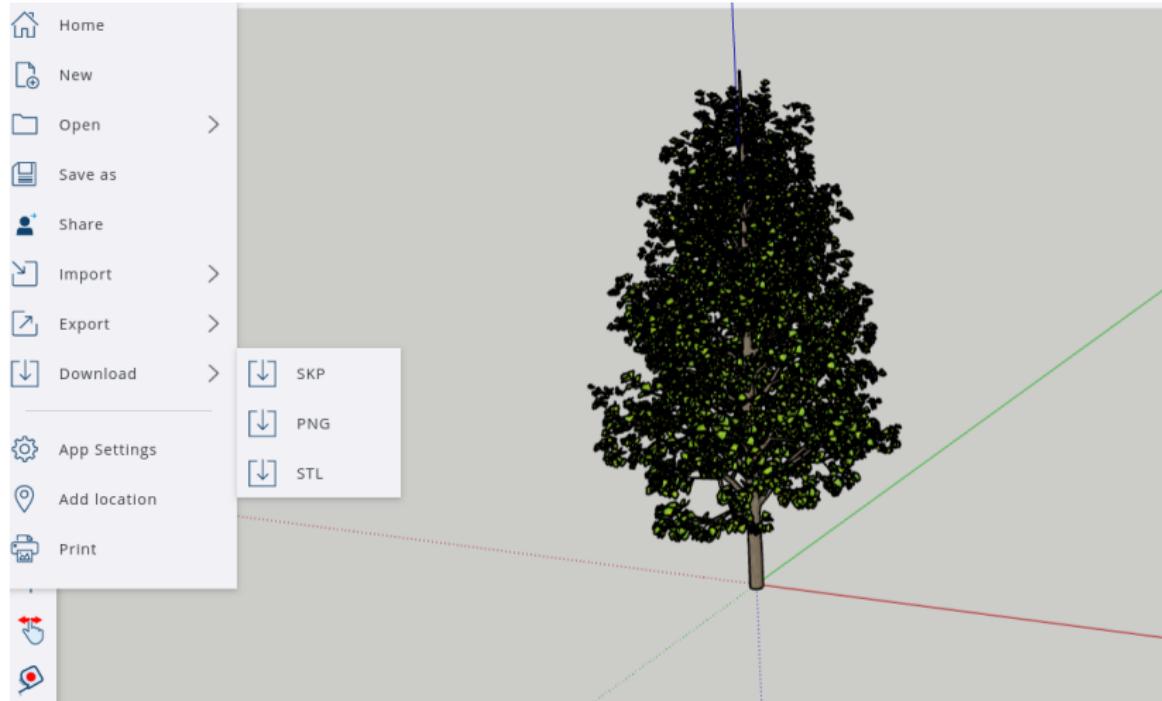


Figure: 3D model of a Ginkgo tree on Sketchup

Tree preprocessing

Remove trunk/branches, normalize and center



Figure: A preprocessed
conifer tree



Figure: A preprocessed
Ginkgo tree

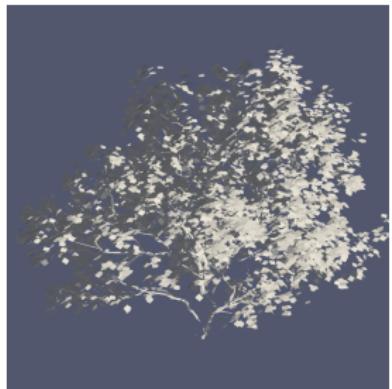
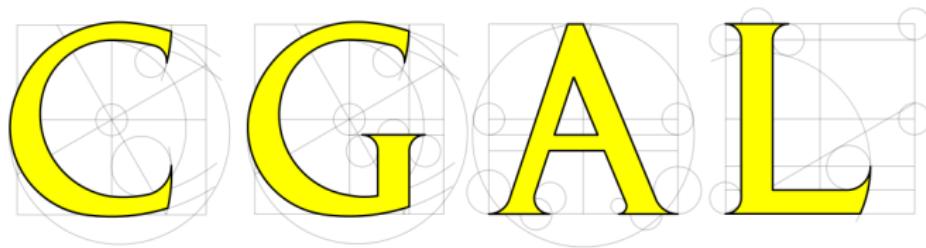


Figure: A preprocessed
Quercus tree

Software and libraries: CGAL



Open source software library for **computational
geometry algorithms**

Tree modeling: Alpha Wrapping



Figure: Different LOD of the Alpha Wrapping of a bike[5]

Tree modeling: Alpha Wrapping

Input:

- 3D model with possible defects

Output:

- Water-tight mesh
- No self-intersections
- Strictly enclosing the input
- Well shaped triangles

Tree modeling: Alpha Wrapping



Figure: Alpha Wrapping in 2D with Offset and different Alpha parameters

Tree modeling: Alpha Wrapping

video link

Tree Modeling: lod 1,2 and 3

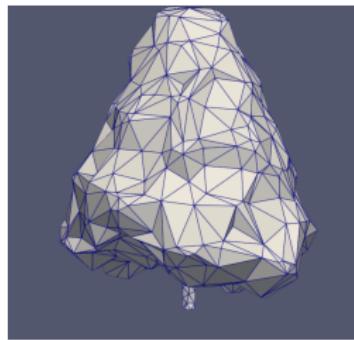


Figure: A wrapped conifer tree for LOD 1

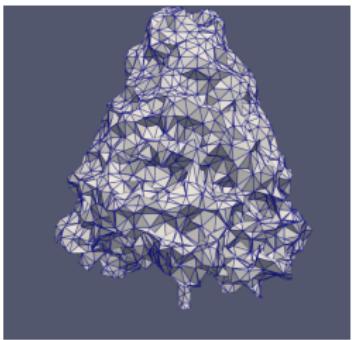


Figure: A wrapped conifer tree for LOD 2

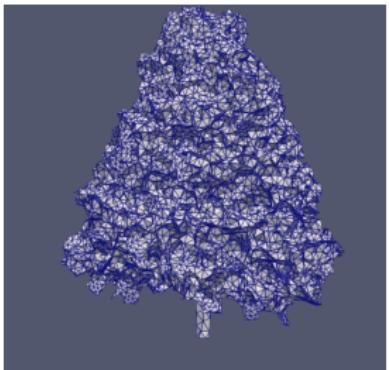


Figure: A wrapped conifer tree for LOD 3

Tree Modeling: lod 1,2 and 3

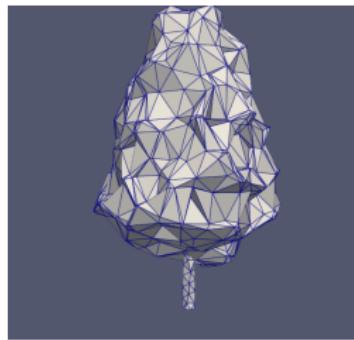


Figure: A wrapped Ginkgo tree for LOD 1

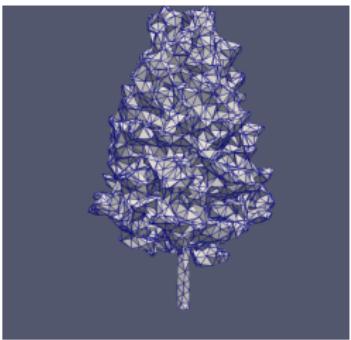


Figure: A wrapped Ginkgo tree for LOD 2

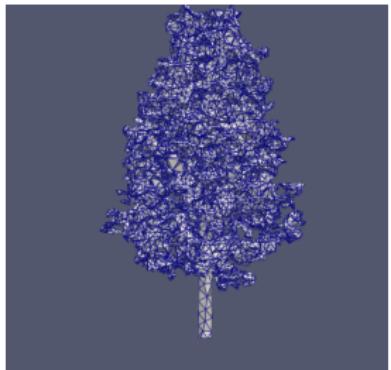


Figure: A wrapped Ginkgo tree for LOD 3

Tree Modeling: lod 1,2 and 3

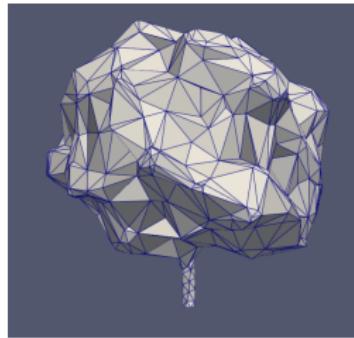


Figure: A wrapped Quercus tree for LOD 1

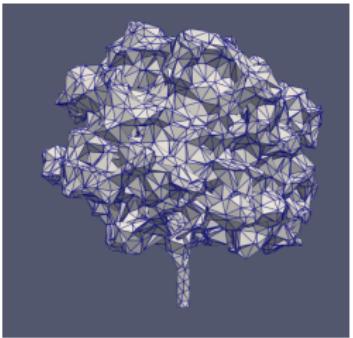


Figure: A wrapped Quercus tree for LOD 2

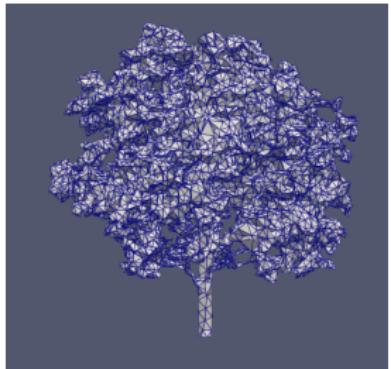


Figure: A wrapped Quercus tree for LOD 3

Tree Modelings

Used Alpha Values for Each LOD

Tree	LOD 0	LOD 1	LOD 2	LOD 3
Alpha	Nan	20	50	100

Number of Faces for Each LOD

Tree	LOD 0	LOD 1	LOD 2	LOD 3
Trunk	28	28	28	28
Cone	72	894	6038	34260
Oval	52	1260	9254	44942
Round	30	1198	10152	45164

Tree Modeling: tagging leaves

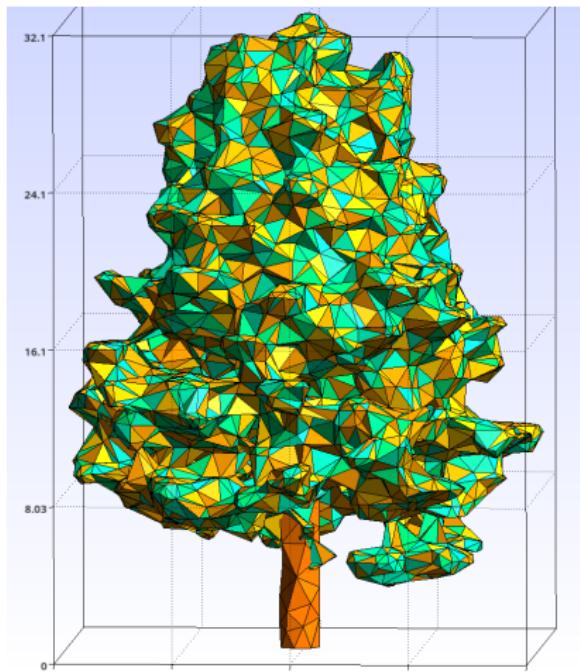


Figure: A tree with 4 different markers on the leaves

Tree Modeling: tree's scaling

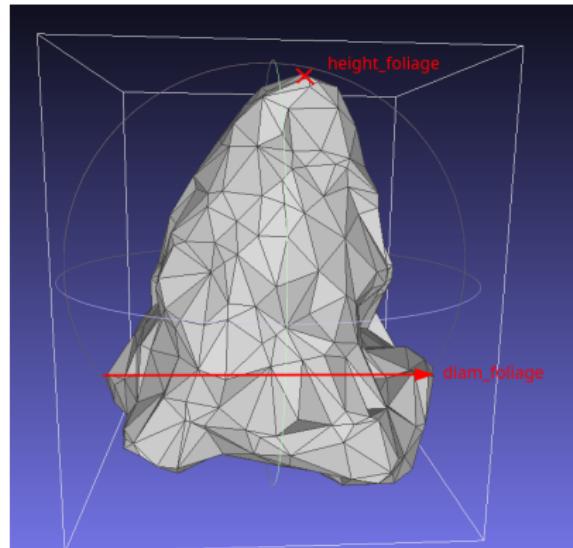


Figure: Foliage mesh

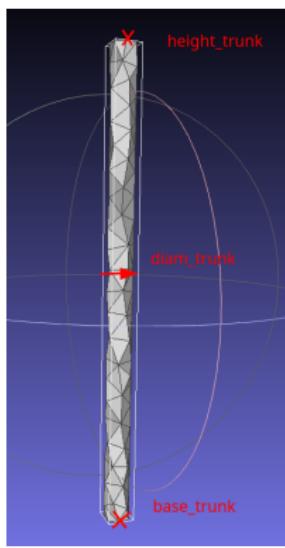


Figure: Trunk mesh

Tree Modeling: Mercator's projection

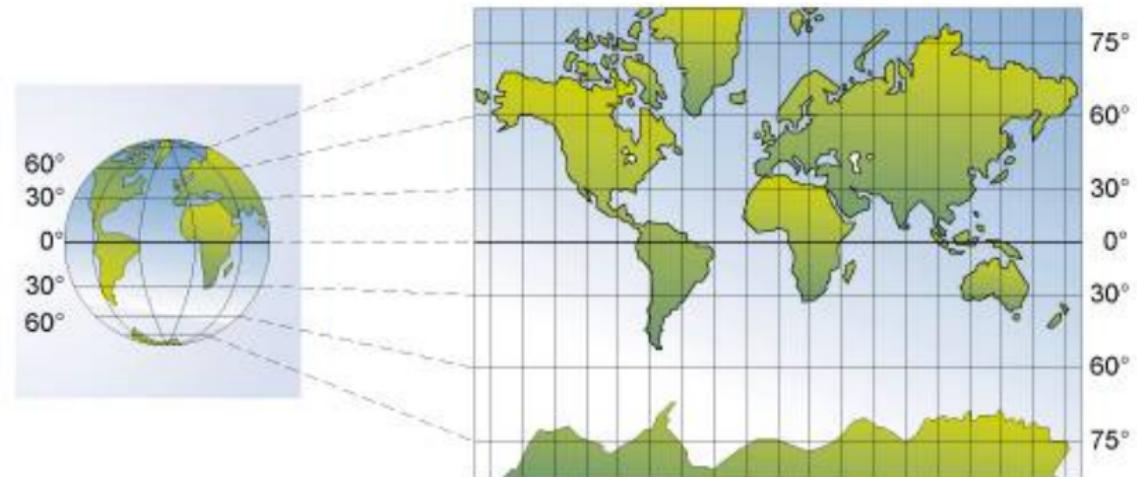


Figure: Mercator's projection[6]

Tree Modeling: Mercator's projection

$$A(\text{latitude, longitude}) = A(\phi, \lambda),$$

projection \Rightarrow
$$\begin{cases} x = \lambda - \lambda_0 \\ y = \ln(\tan(\frac{\pi}{4} + \frac{\phi}{2})) \end{cases}$$
 (1)

, where λ_0 is the center of the map

Tree Modeling: Mercator's projection

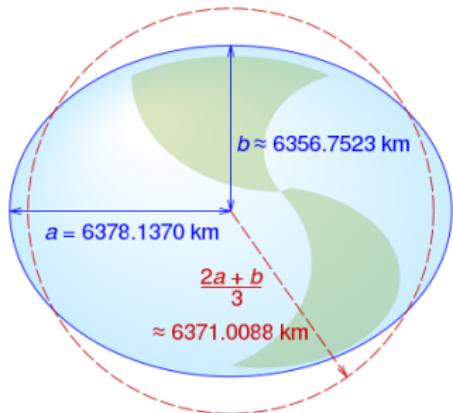


Figure: Earth as an ellipsoid[7]

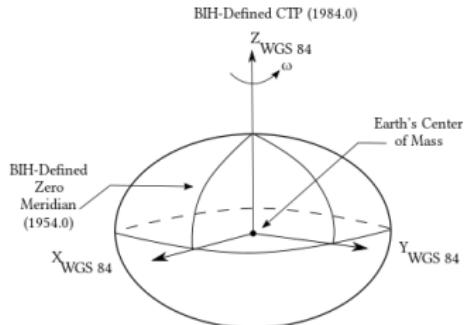


Figure 1.1 WGS 84 Reference Frame

Figure: WGS 84 reference frame[7]

WGS84toCartesian.hpp \implies **GPS to Cartesian**

Tree Modeling: foliage and trunk placement

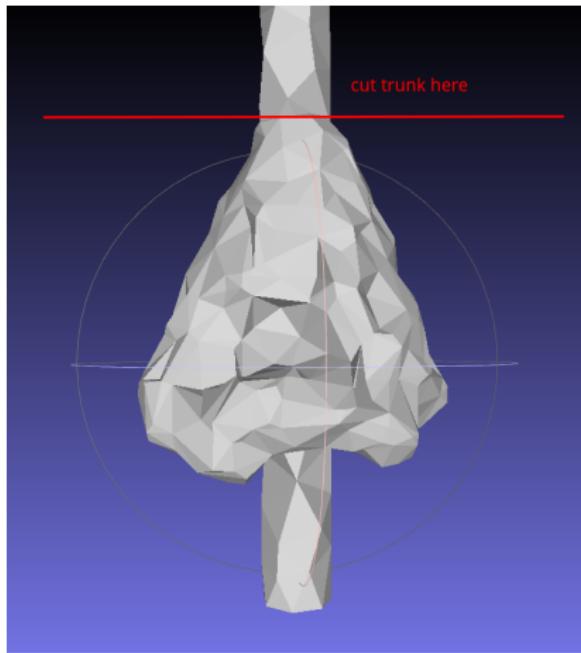


Figure: .3D model of a tree with a trunk that is too long

Tree Modeling: tree's placement

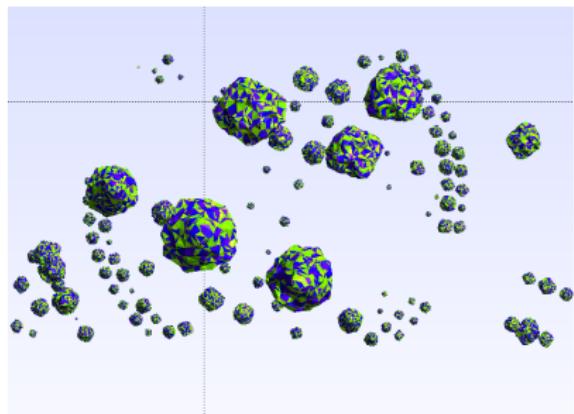


Figure: Republic square with LOD 1 trees

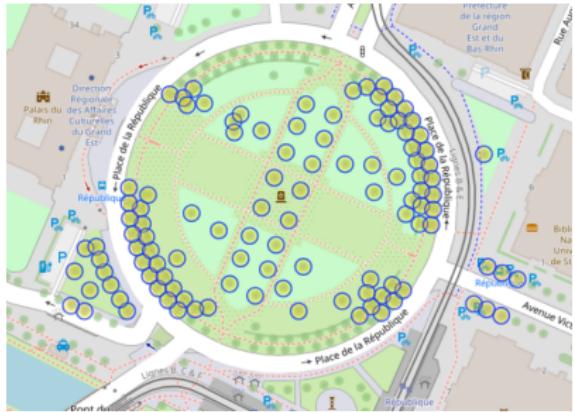


Figure: Republic square trees from Overpass turbo[8]

Tree Modeling: tree's placement

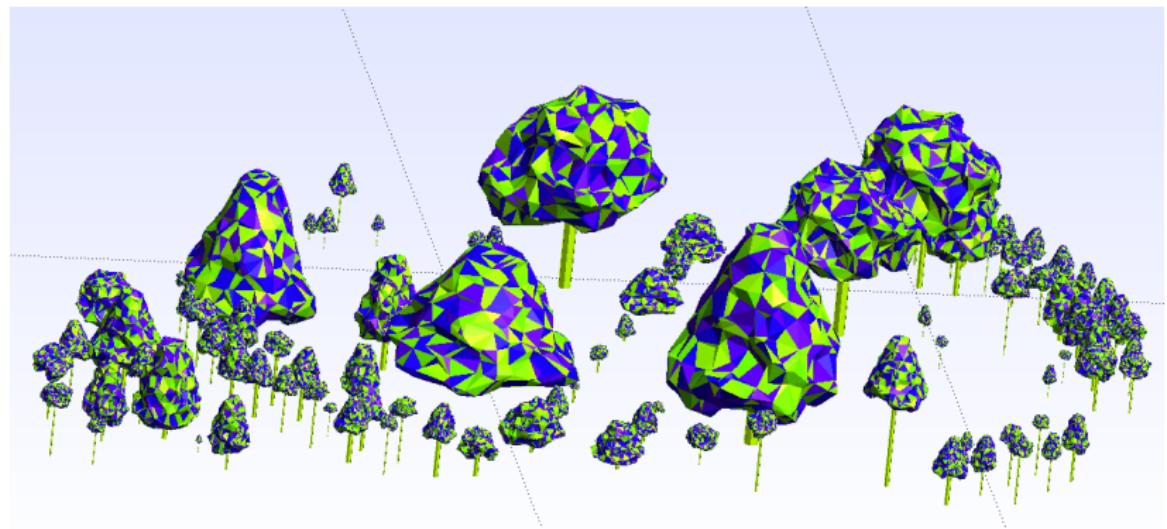


Figure: Republic square with LOD 1 trees

Tree Modeling: tree's altitude

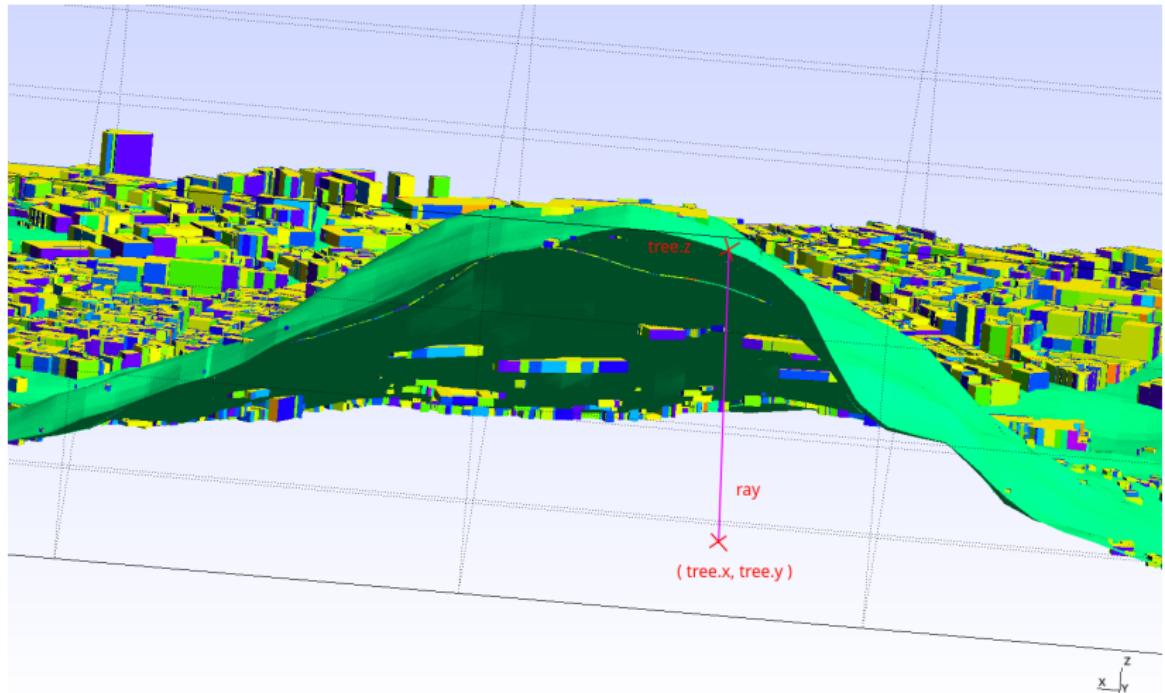


Figure: Raycasting of a tree in Grenoble, France

Raycasting: BVH

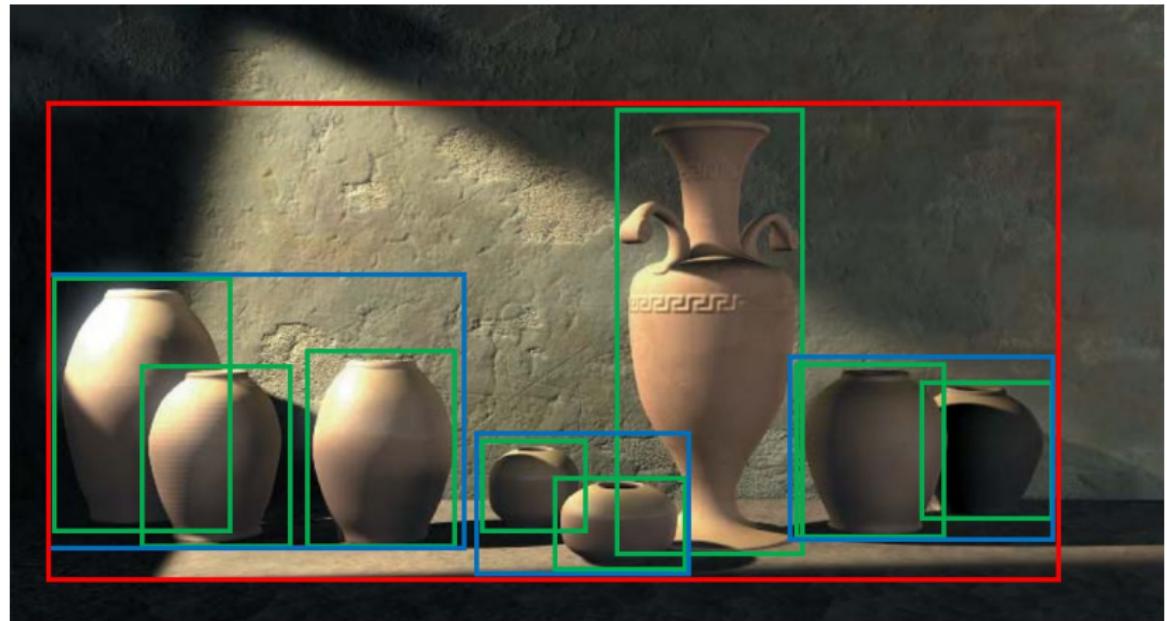


Figure: Image based on A 3-Dimensional Representation for Fast Rendering of Complex Scenes, by Rubin and Whitted. [9]

Tree Modeling: mesh merging



Figure: A plant going through a wall

Tree Modeling: mesh merging

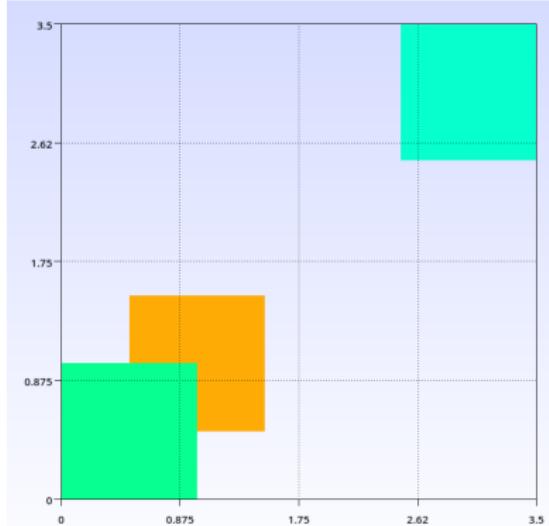


Figure: Autorefine before

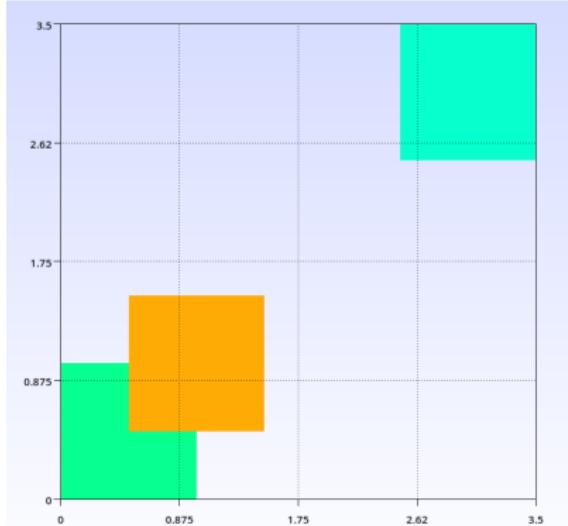


Figure: Autorefine after

Tree Modeling: thread parallelization

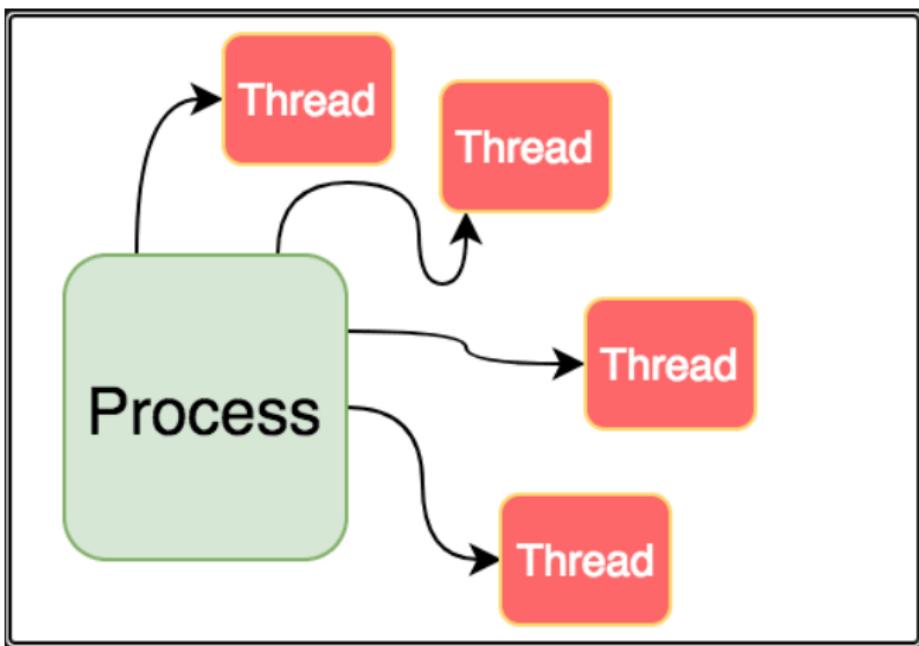


Figure: Parallelization of the tree generation process

Tree Modeling: thread parallelization autorefine

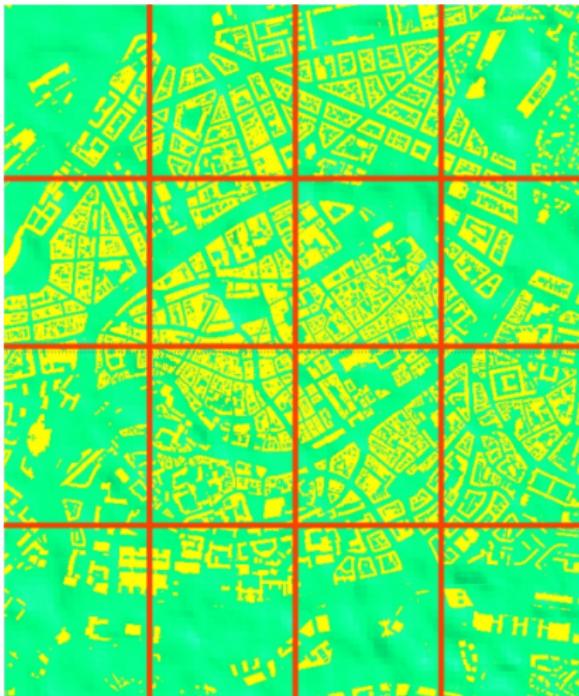


Figure: Parallelization of the mesh refinement process

Results: Model Integration

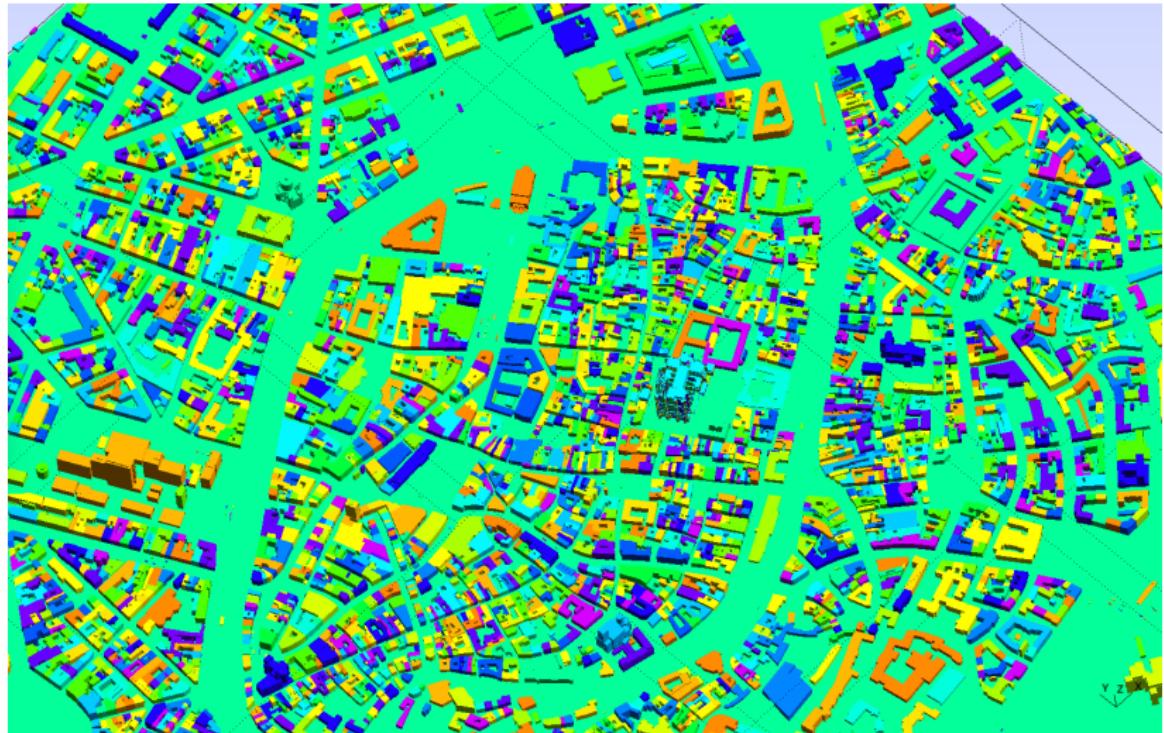


Figure: 3D Model of Strasbourg city center, buildings only

Results: Model Integration

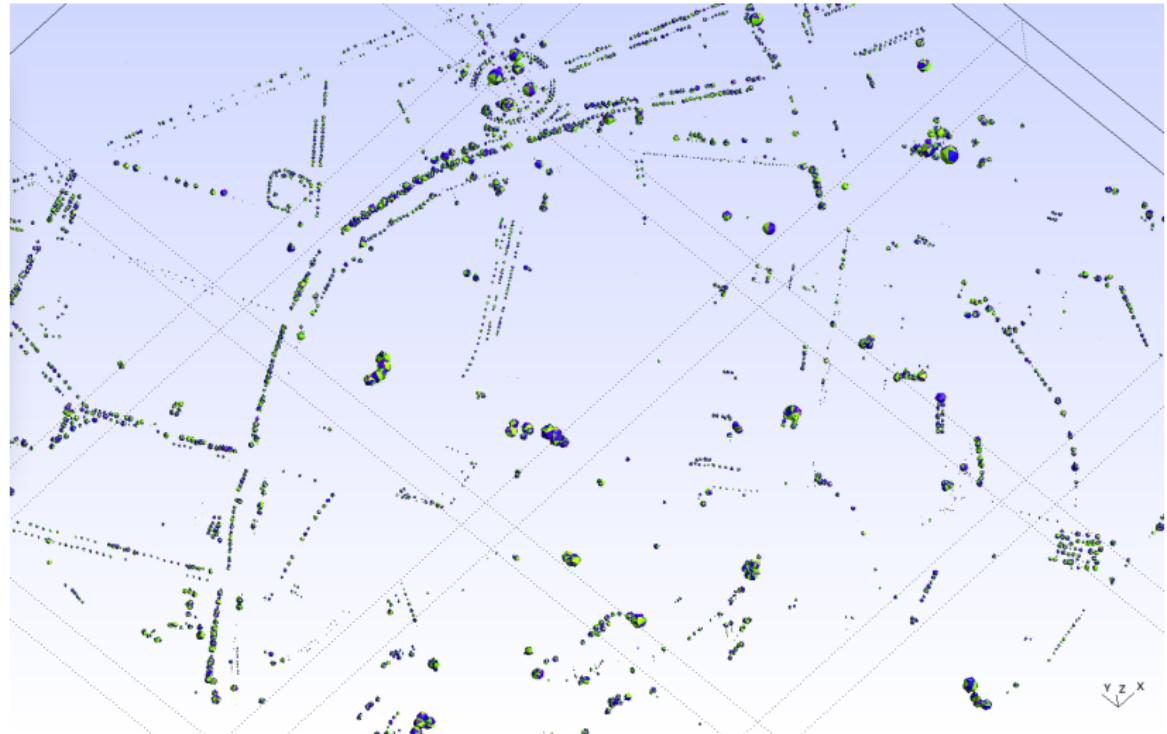


Figure: 3D Model of Strasbourg city center, trees only

Results: Model Integration



Figure: 3D Model of Strasbourg city center, buildings and trees

Results: Model Integration

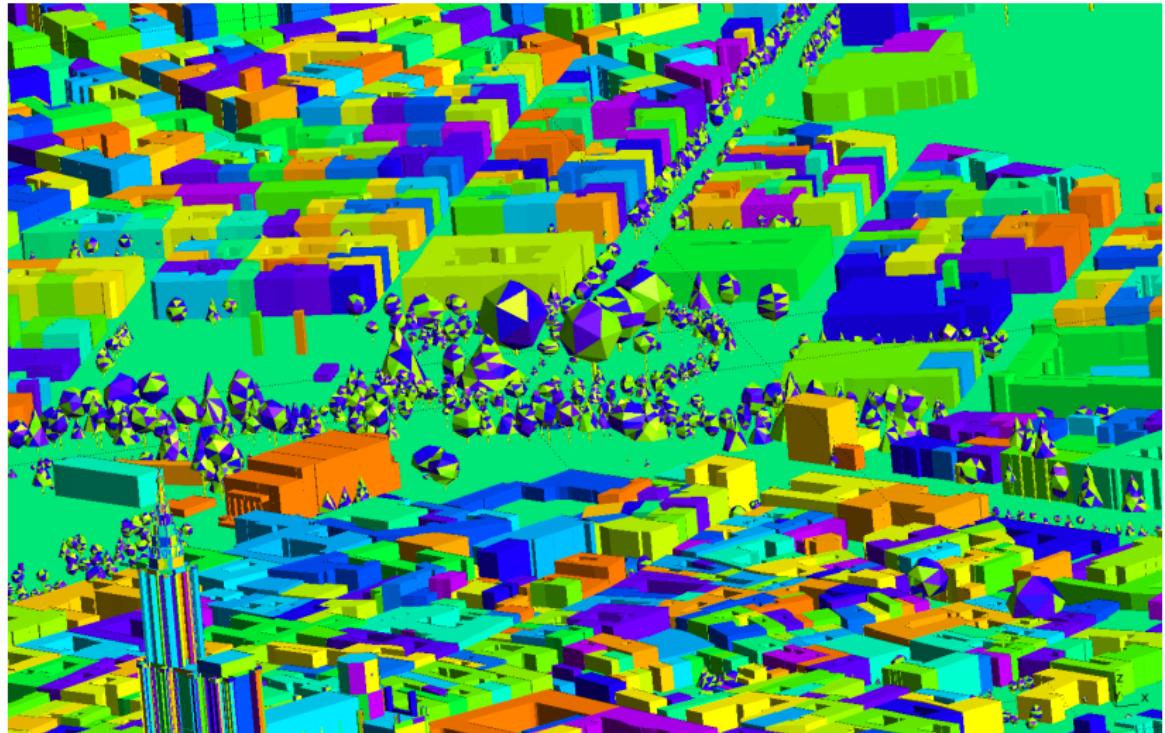


Figure: 3D Model of Strasbourg city center, buildings and vegetation, with a focus on Republic Square, LOD 0

Results: Model Integration

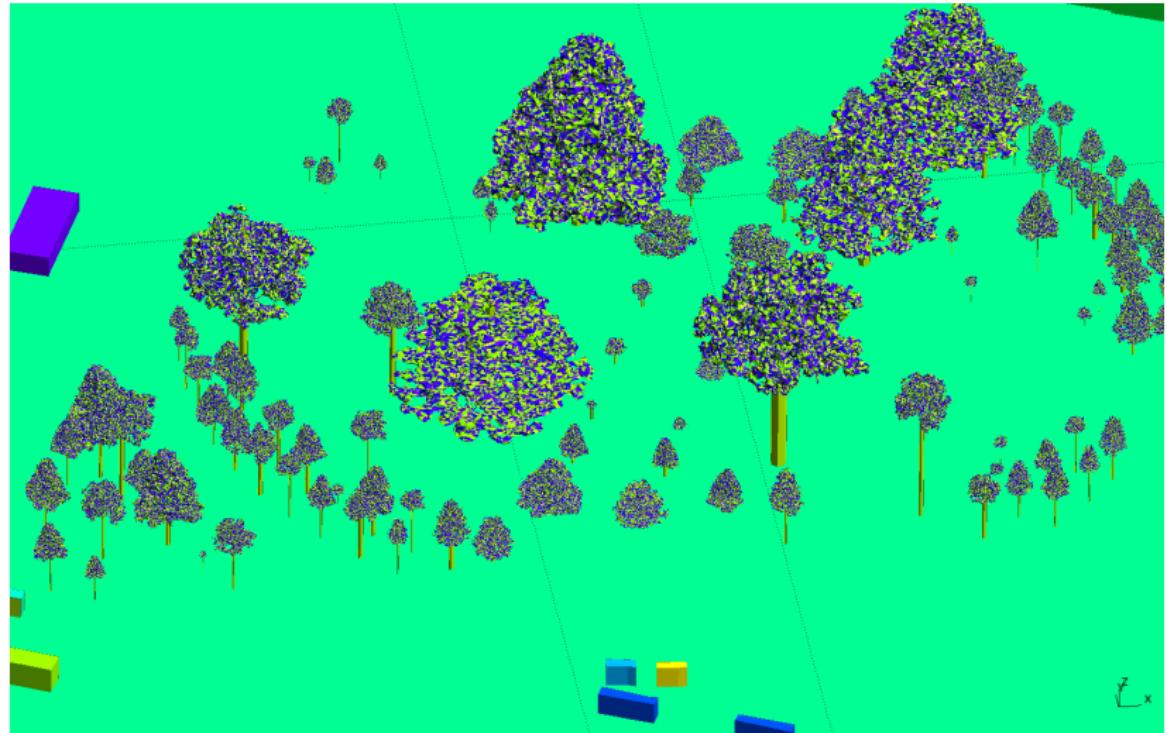


Figure: 3D Model of Strasbourg city center, buildings and vegetation, with a focus on Republic Square, LOD 3

Results: Performance



Figure: Bounding Box 1: 153.7 m^2 , 12 trees

Results: Performance

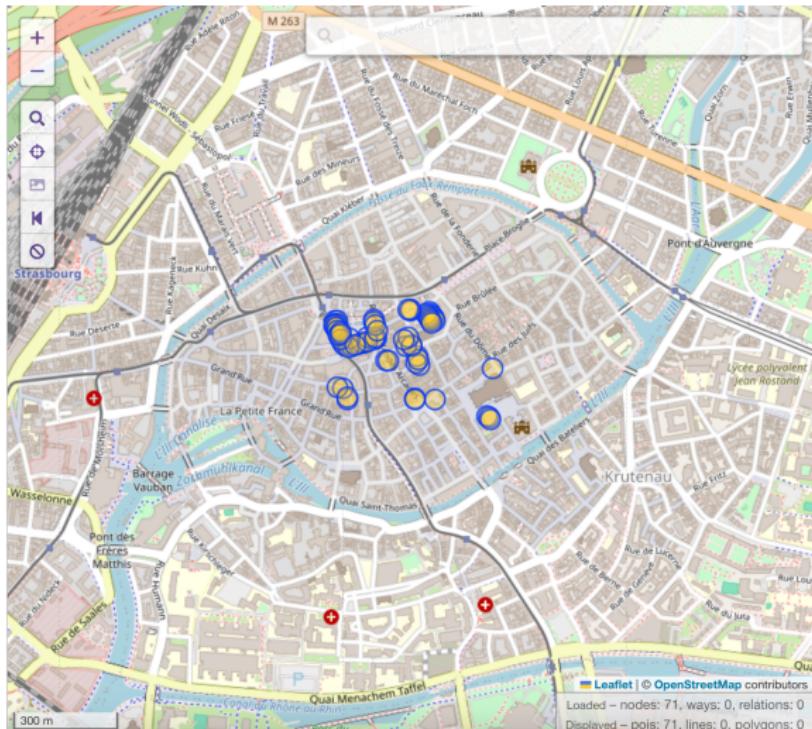


Figure: Bounding Box 2: 384.0 m², 71 trees

Results: Performance

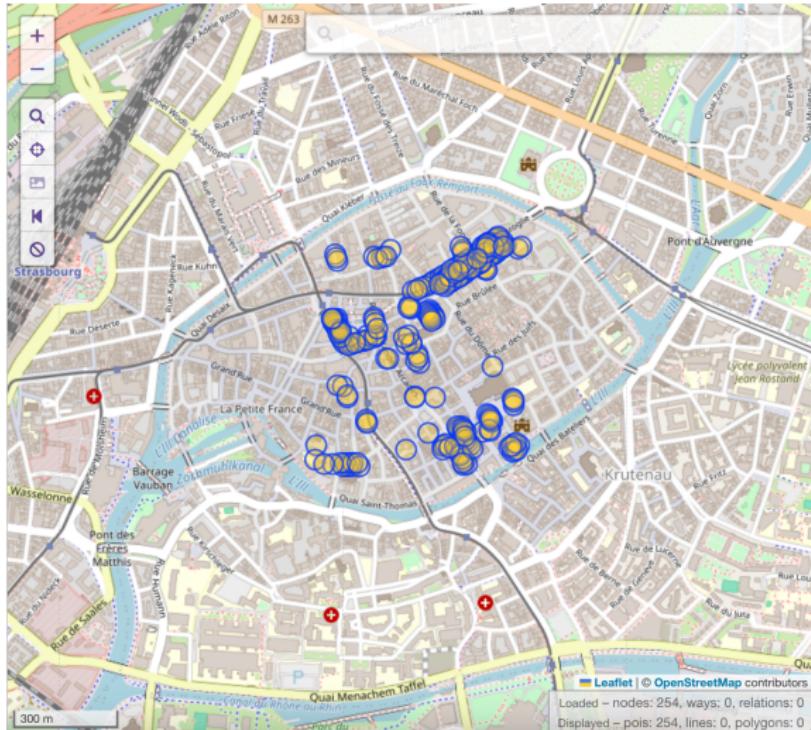


Figure: Bounding Box 3: 626.1 m², 254 trees

Results: Performance

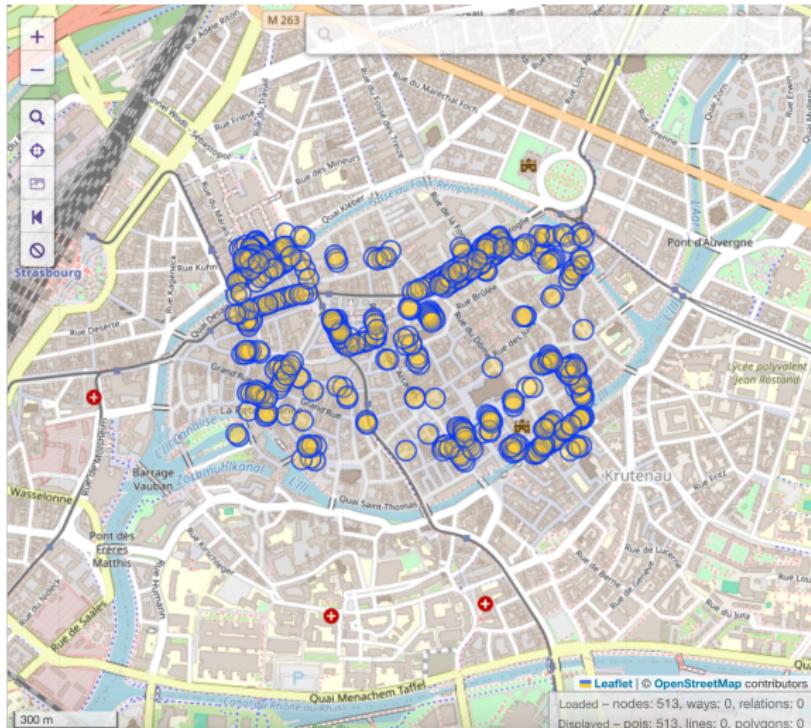


Figure: Bounding Box 4: 808.4 m², 513 trees

Results: Performance

Without LOD 3

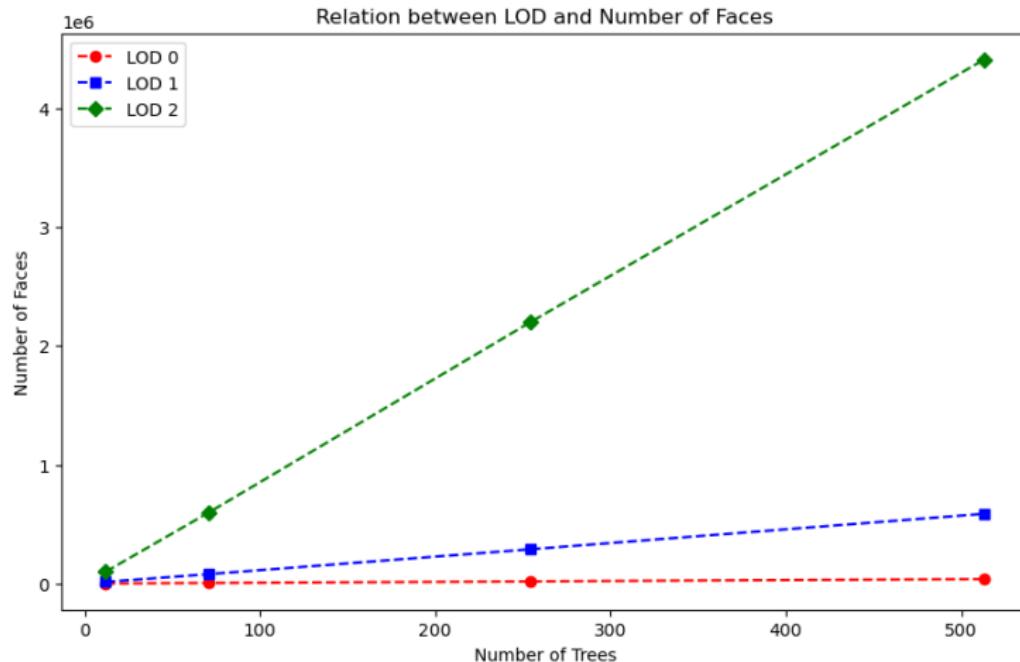


Figure: Relationship between the number of faces and the number of trees

Results: Performance

With LOD 3

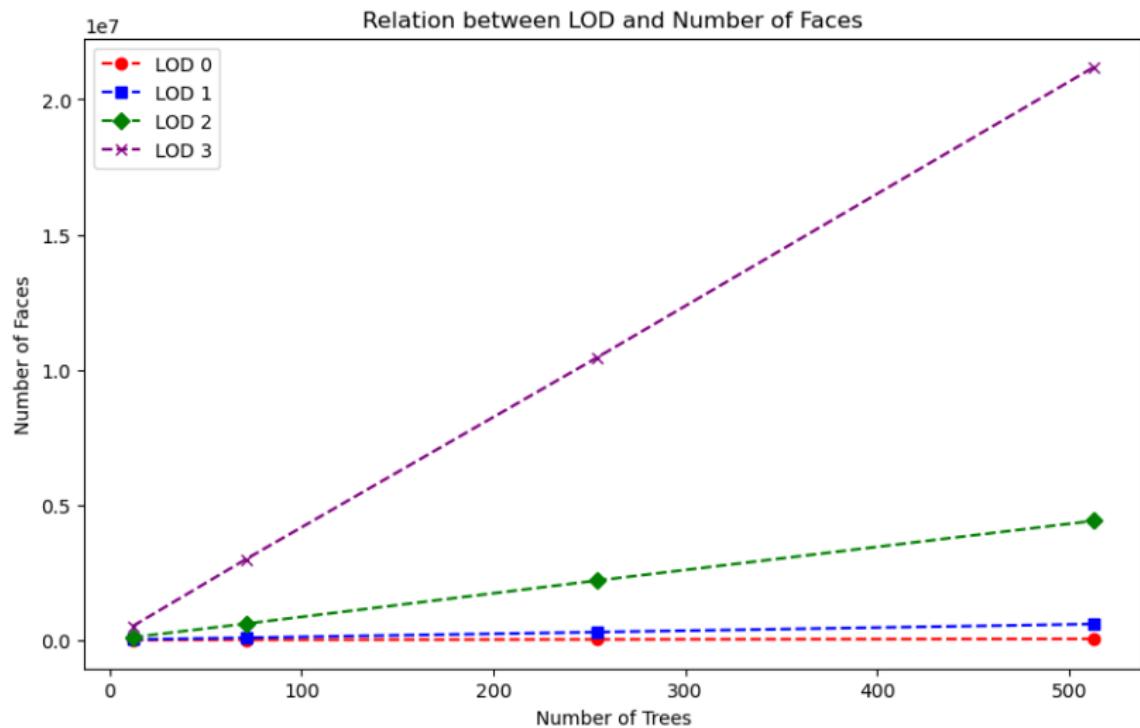


Figure: Relationship between the number of faces and the number of trees

Results: Performance

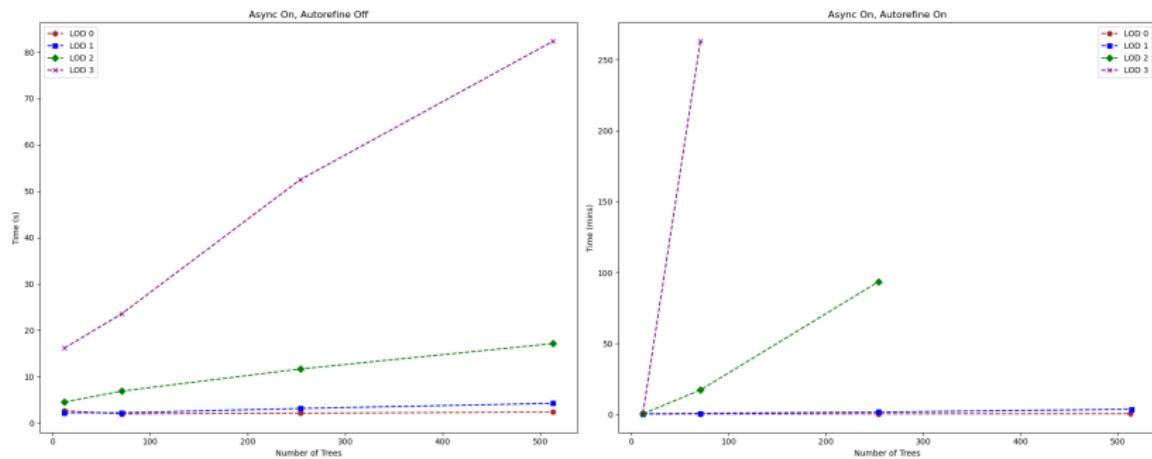


Figure: Comparison of the execution time with and without the autorefine method

Results: Performance

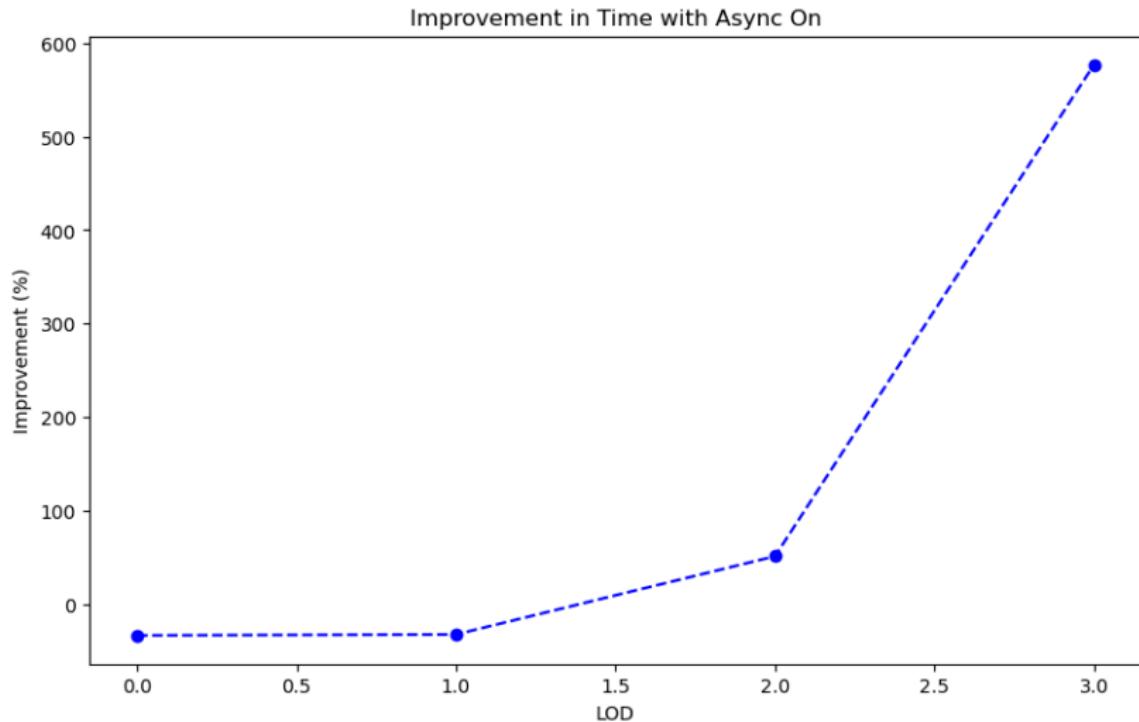


Figure: Thread parallelization improvement

Prospects

- Fixing the tree's altitude

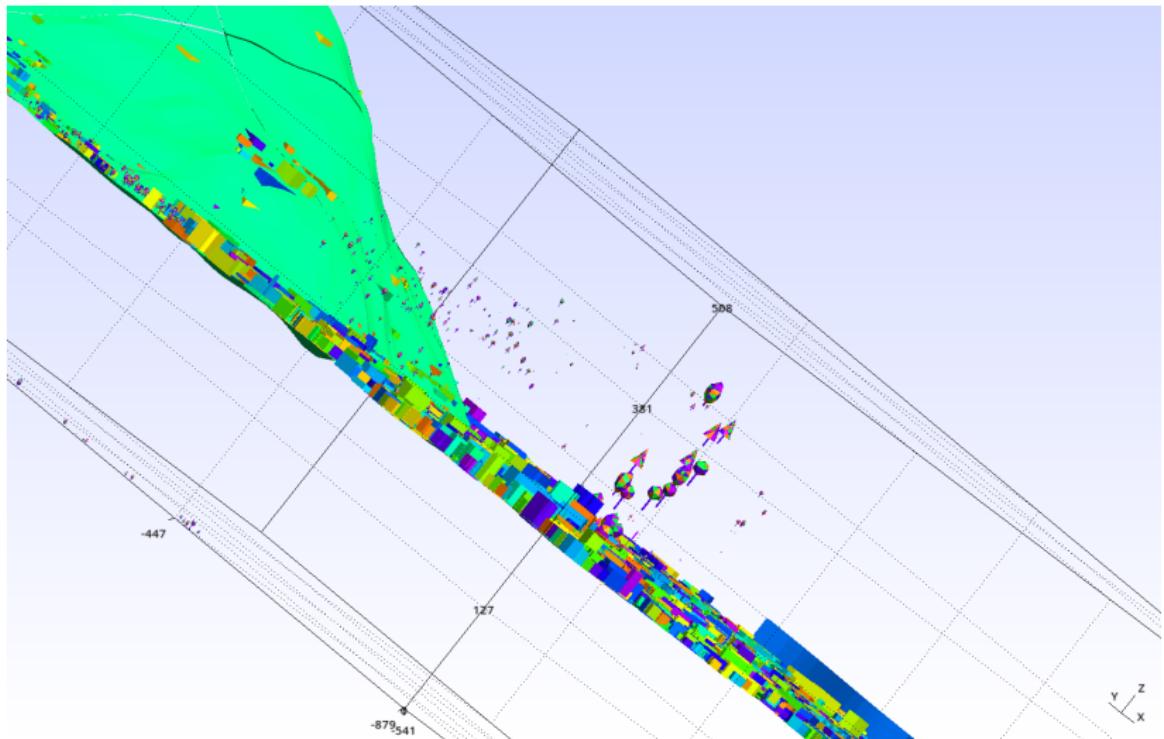


Figure: Grenoble elevation fail

Prospects

- Parallelize the computation of the ‘autorefine’ method

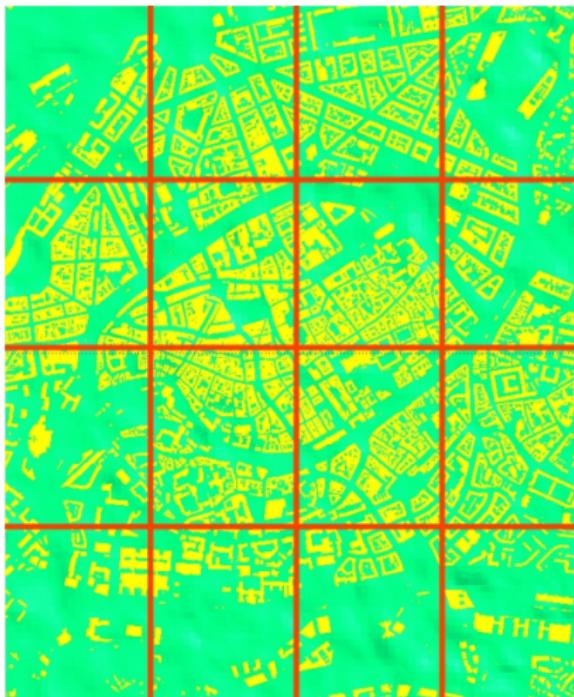


Figure: Terrain mesh splitted into smaller parts, the red lines represent the boundaries of the parts and should not be parallelized

Prospects

- Improve the handling of missing tree data

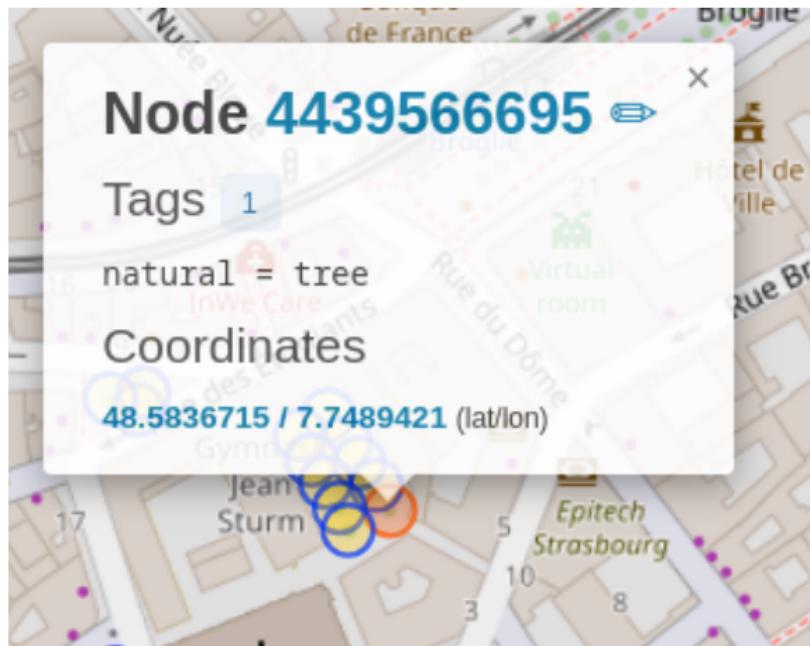


Figure: An Overpass Turbo Node with missing metadata

Prospects

- Shading calculation

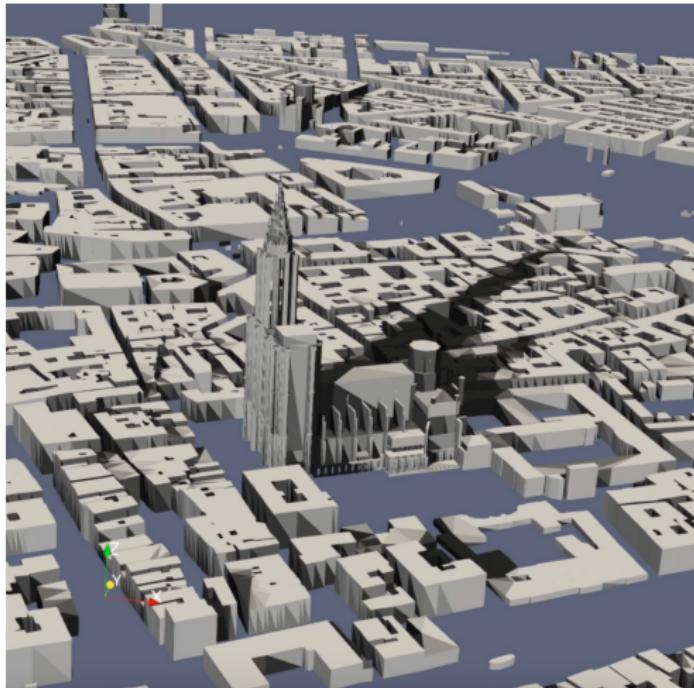


Figure: The shadow cast by the Cathedral of Strasbourg

Conclusion

- **Project Goal:** Enhanced urban modeling by integrating 3D tree models into urban environments.
- **Key Achievements:**
 - Extracted tree data from OpenStreetMap.
 - Generated 3D tree models using CGAL and Gmsh.
 - Integrated models into existing terrain meshes.
 - Utilized different Levels of Detail (LOD) for flexible modeling.
- **Technical Highlights:**
 - Robust data acquisition and model generation techniques.
 - Successful scalability and performance benchmarks.
 - Addressed missing data with default tree height values.
- **Future Directions:**
 - Incorporate tree elevations, parallelization, and advanced shading for improved modeling.

The end

Thank you for your attention!



Any questions?

Contact Information:

pierre.antoine.senger@gmail.com

github.com/PA-Senger

 Numpex.

Exa-MA.

<https://numpex.org/>

exama-methods-and-algorithms-for-exascale/, 2024.

 P. Verchere.

Thermal image of a street in the city, 2023.

 Yujin Park, Jean-Michel Guldmann, Desheng Liu.

Impacts of tree and building shades on the urban heat island:
Combining remote sensing, 3D digital city and spatial regression
approaches.

[https://www.sciencedirect.com/science/article/pii/
S0198971521000624](https://www.sciencedirect.com/science/article/pii/S0198971521000624), 2021.

 Christophe Prud'homme.

New York City mesh, 2023.

 Pierre Alliez, David Cohen-Steiner, Michael Hemmer, Cédric
Portaneri, Mael Rouxel-Labbé.

CGAL 5.6.1 - 3D Alpha Wrapping, 2024.

 Bibm@th.

Mercator projection, 2024.



Wikipedia.

Mercator projection.

Wikipedia, 2024.



Roland Olbricht, Martin Raifer.

Overpass turbo.

https://wiki.openstreetmap.org/wiki/Overpass_turbo,
2024.



Jacco Bikker.

How to build a BVH.

[https://jacco.ompf2.com/2022/04/13/
how-to-build-a-bvh-part-1-basics/](https://jacco.ompf2.com/2022/04/13/how-to-build-a-bvh-part-1-basics/), 2022.



Yannick Verdie, Florent Lafarge, Pierre Alliez.

LOD Generation for Urban Scenes.

ACM Transactions on Graphics, 34(3):15, 2015.

hal-01113078.



Yannick Verdie, Florent Lafarge.

Detecting parametric objects in large scenes by Monte Carlo sampling.

International Journal of Computer Vision, 106(1):57–75, 2014.

 O. Stava, S. Pirk, J. Kratt, B. Chen, R. Mečh, O. Deussen, B. Benes.

Inverse Procedural Modeling of Trees.

Preprint, 2014.

Adobe Systems Inc., USA; University of Konstanz, Germany;
Shenzhen Institute of Advanced Technology, China; Purdue
University, USA.

 Shenglan Du, Roderik Lindenbergh, Hugo Ledoux, Jantien Stoter,
Liangliang Nan.

AdTree: Accurate, Detailed, and Automatic Modelling of
Laser-Scanned Trees.

MDPI, 2019.

 Bedrich Benes.

Computational vegetation.

<https://cs.purdue.edu/homes/bbenes/vegetation/>.

 CGAL Development Team.

CGAL User and Reference Manual.

<https://doc.cgal.org/latest/Manual/index.html>.

 [Feel++ Consortium.](#)

Feel++.

<https://docs.feelpp.org/home/index.html>.

 [curl.](#)

<https://curl.se/>.

 [MeshLab Developers.](#)

MeshLab.

<https://www.meshlab.net/>.

 [OpenStreetMap Contributors.](#)

Overpass API.

https://wiki.openstreetmap.org/wiki/Overpass_API.

 <https://www.openstreetmap.org/help>.

 [Tania Landes.](#)

D'où vient le pouvoir rafraîchissant des arbres en ville ?

<https://theconversation.com/>

dou-vient-le-pouvoir-rafraichissant-des-arbres-en-ville-1999
2023.

 [Conseil départemental de la Somme.](#)

Aerial thermal view, 2023.



INSA Strasbourg.

Climatologie urbaine : suivi des arbres en ville à Strasbourg.

<https://www.youtube.com/watch?v=IJoj7Knm-uA>, 2023.



Wikipedia.

OFF (file format) - Wikipedia.
2024.



Wikipedia.

STL (file format) - Wikipedia.

[https://en.wikipedia.org/wiki/STL_\(file_format\)](https://en.wikipedia.org/wiki/STL_(file_format)), 2023.



CGAL Development Team.

CGAL 5.6.1 - 2D and 3D Linear Geometry Kernel , 2024.



Wikipedia.

World Geodetic System.

[https:](https://)

[//en.wikipedia.org/wiki/World_Geodetic_System#WGS_84](https://en.wikipedia.org/wiki/World_Geodetic_System#WGS_84),
2024.



Christian Berger.

WGS84toCartesian.

<https://github.com/chrberger/WGS84toCartesian?tab=readme-ov-file>, 2021.

 [Wikipedia](#).

K-nearest neighbors algorithm.

https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm,

2024.

 [Wikipedia](#).

SketchUp.

<https://en.wikipedia.org/wiki/SketchUp>, 2024.

 [Wikipedia](#).

JSON.

<https://en.wikipedia.org/wiki/JSON>, 2024.

 [CGAL Development Team](#).

CGAL 5.6.1 - 3D Alpha Shapes.

https://doc.cgal.org/latest/Polygon_mesh_processing/group__PMP__corefinement__grp.html, 2024.

 [HiDALGO2](#).

Hidalgo2 ubm.

 HiDALGO2.

HiDALGO2.

<https://www.hidalgo2.eu>, 2024.

 European Commission.

European Green Deal.

https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/european-green-deal_en, 2024.

 Wikipedia.

Delaunay triangulation.

Wikipedia, 2024.

 Dimitri van Heesch.

Doxxygen.

<https://www.doxxygen.nl/index.html>, 2024.

 HiDALGO2.

About HiDALGO2.

<https://www.hidalgo2.eu/about>, 2024.

 Cemosis.

Cemosis.

<https://www.cemosis.fr>, 2024.



IRMA.

IRMA.

<https://irma.math.unistra.fr>, 2024.



INRIA.

INRIA.

<https://www.inria.fr>, 2024.



Pierre Alliez.

Pierre Alliez.

<https://team.inria.fr/titane/pierre-alliez>, 2024.



Vincent Chabannes.

Vincent Chabannes.

<https://www.researchgate.net/profile/Vincent-Chabannes>,
2024.



Kitware.

ParaView.

<https://www.paraview.org>, 2024.

 CGAL Development Team.

CGAL.

<https://github.com/CGAL/cgal>, 2024.

 Roland Olbricht.

Overpass QL.

[https:](https://wiki.openstreetmap.org/wiki/Overpass_API/Overpass_QL)

[//wiki.openstreetmap.org/wiki/Overpass_API/Overpass_QL](https://wiki.openstreetmap.org/wiki/Overpass_API/Overpass_QL),
2024.