Report, Kinetic project

Dorian Geraldes Pereira, Axel Demuth

March 2024

Contents

1 Project Objective

Within this project scope, the primary goal is to implement an efficient and accurate conversion process for Industry Foundation Classes (IFC) files representing buildings or cities into meshes compatible with the Kinetic algorithm. Subsequently, the Kinetic algorithm will be applied to these meshes to produce watertight models, facilitating the execution of finite element calculations.

The specific steps to be undertaken are as follows:

1. **Mesh Conversion:** From the IFC files we will have a conversion in the stl or msh format,we will need to convert the STL or MSH meshes into one of the formats accepted by the Kinetic algorithm, such as .ply, .xyz, .las, .off.

2. **Application of the Kinetic Algorithm:** Applicate the Kinetic algorithm on the the converted meshes to produce meshes optimized for finite element calculations.

3. **Recovery of Material Labels:** Ensure the preservation of information regarding materials present in the initial IFC-format mesh and correctly associate them with elements of the converted mesh.

4. **Utilization on City Modeling:** Extend the application of the Kinetic algorithm to entire city models.

2 Current Project Challenges

Currently, the project faces several technical challenges:

1. **Mesh Conversion:** Find a solutions to convert meshes from STL or MSH files into one of the formats accepted by the Kinetic algorithm.

2. **Parameter Optimization:** Identify and adjust appropriate parameters to avoid segmentation faults and achieve satisfactory results when applying the Kinetic algorithm.

3. **Version Differences:** Understand the distinctions between versions of the Kinetic algorithm,developed by CGAL and INRIA, to select the right parameters to get the best result

By overcoming these challenges, the project aims to provide a comprehensive and efficient solution for analyzing urban structures using the Kinetic algorithm to facilitate finite element calculations.

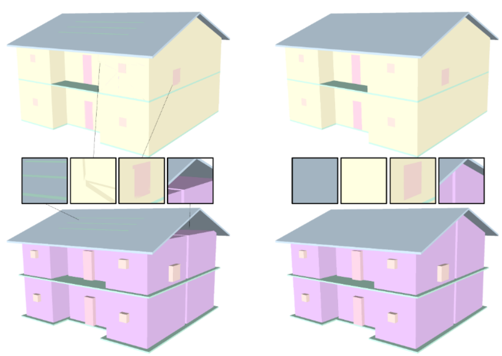
3 Tools

3.1 CGAL

CGAL is a comprehensive package for geometry algorithms, providing various data structures and algorithms for working on polygons, surfaces, mesh generation, and more. It offers a wide range of functionalities for geometric processing and analysis in various fields such as computer graphics, computational geometry, and geometric modeling.

3.2 Kinetic

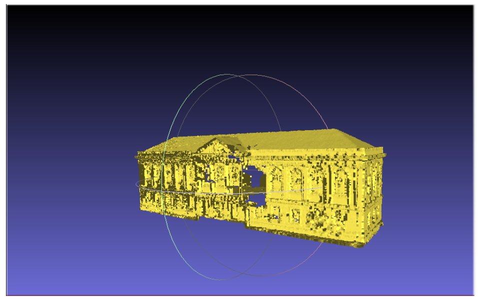
Kinetic algorithms is a package from CGAL that allows working on meshes with some holes in them. When applied to the mesh, the Kinetic algorithms will 'extend' some surfaces to fill the mesh and make it watertight. Here's what the algorithm is capable of:



o utilize the algorithm, we employ one from CGAL. Given a set of parameters and a file containing a point cloud along with the associated normals to the points, the algorithm is applied. We were fortunate to have a meeting with Florent Lafarge, one of the creators of the algorithm, who explained to us which parameters are crucial for analysis and how each one can significantly influence the results. Two parameters stand out as particularly important: 'dist' and 'pmin.' 'Pmin' represents the number of points used to construct a plane, while 'dist' indicates the distance between two points required to consider them for plane construction.

In order to understand how these parameters affect the outcome, we will examine the same point cloud while varying the value of 'pmin' first and then varying 'dist.' The algorithm produces .off files as output, which can be visualized using software such as MeshLab.

The original point cloud represents this building:



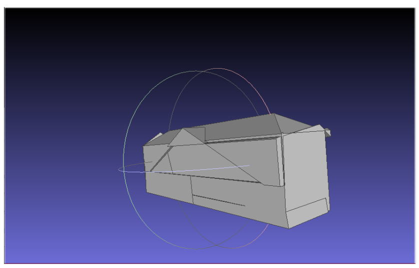


Figure 1: dist1_pmin220

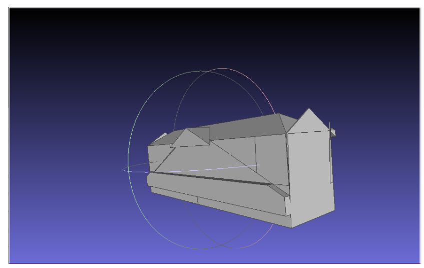


Figure 2: dist1_pmin250

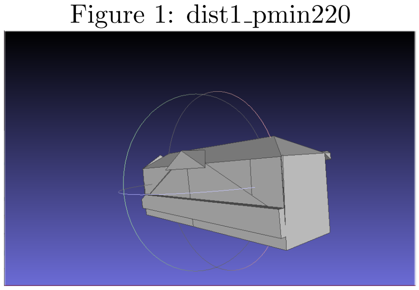


Figure 3: dist1_pmin280

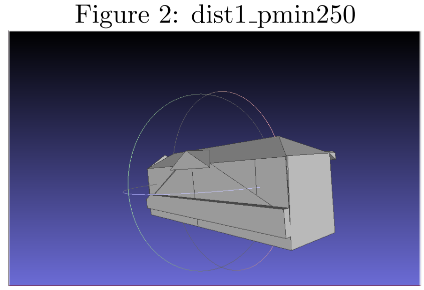


Figure 4: dist1_pmin300

First, it's important to understand that setting 'pmin' too low can lead to errors. Subsequently,if 'pmin' is too small, we may not capture enough structural detail, whereas setting 'pmin' too high may cause planes to overlap, resulting in loss of information

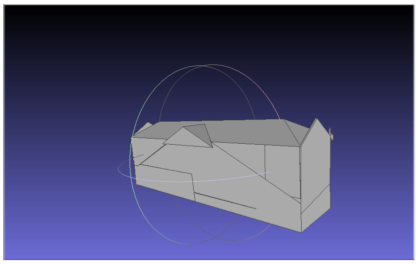


Figure 5: dist15_pmin250.png

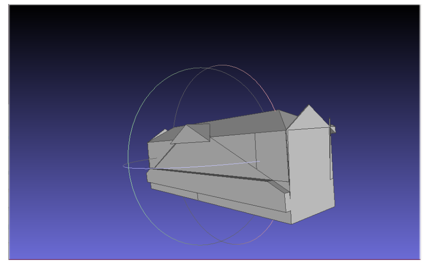


Figure 6: dist1_pmin250

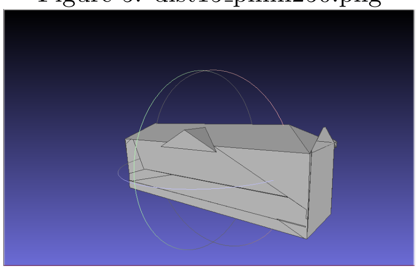


Figure 7: dist03_pmin250

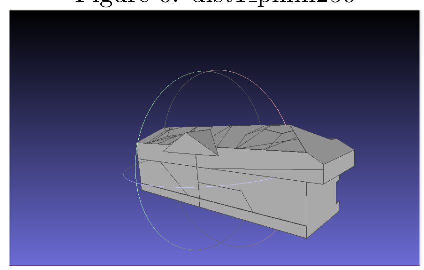


Figure 8: dist001_pmin250

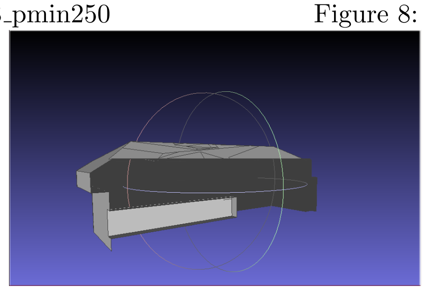


Figure 9: dist001_pmin250

When considering the 'dist' parameter, if it is set too high, we risk losing significant structural details. In the opposite, if the distance is too small, surfaces may be divided into too many planes, potentially resulting in an incomplete coverage of the entire mesh, as observed at the rear of the building with a 'dist' of 0.001.

One of the challenges in studying this algorithm is determining the appropriate 'qmin' and 'dist' to achieve the desired number of space partitions. Setting it too high may enhance precision but extend execution time, and in some cases, incorrect values can lead to program crashes.

This algorithm could be a crucial tool in our project, particularly when dealing with massive meshes for large buildings, hospitals, etc. When creating large meshes, issues can arise, and errors within the mesh can be detrimental during simulations,potentially leading to false results. This algorithm allows us to rectify such mesh problems.

One limitation of the algorithm is its input requirement, as it currently only works with scatter plots. Consequently, we are unable to utilize the IFC format, resulting in the loss of valuable information regarding the types of structures present. To address this limitation, we plan to implement a process involving the conversion of IFC to scatter plot with associated data, followed by conversion to formats such as .plt, .ply, .xyz, which are supported by Kinetic CGAL, ultimately resulting in a mesh with associated data.

4 Analysis of Results

Analysis of the final meshes reveals significant differences between those produced by the CGAL algorithm and those generated by the INRIA algorithm. The final mesh obtained with CGAL demonstrates satisfactory watertightness but is characterized by noticeable roughness. Conversely, the final mesh generated by the INRIA algorithm is remarkably smoother, offering a more uniform surface.

4.1 Visualization of Results

For better visual understanding, three images are provided below:

- Initial point cloud
- Result of the mesh with the CGAL algorithm
- Result of the mesh with the INRIA algorithm

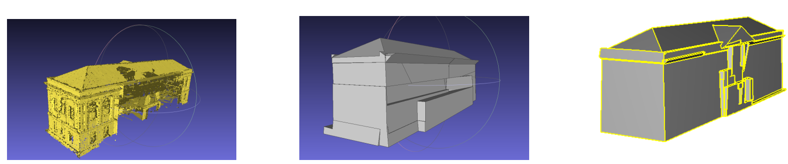


Figure 10: Visualization of results with a building

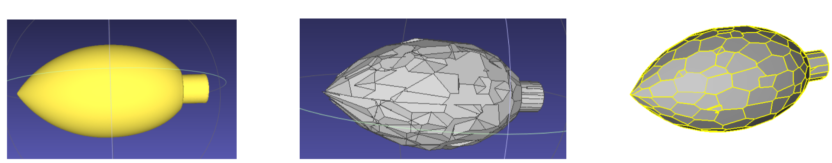


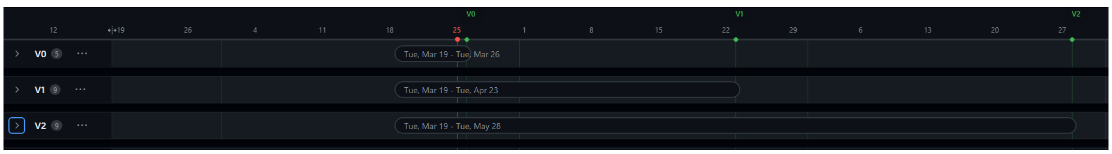
Figure 11: Visualization of results with a flame

Future Perspectives

While the results obtained with CGAL and INRIA library data provide valuable insights, it is essential to extend our analysis to data from files in IFC format. We plan to integrate this data into our upcoming experiments to evaluate the performance of the algorithms on real models of buildings and cities.

5 Roadmap

We intend to work on this project in the coming months and will continuously update our progress as outlined in the following roadmap.



6 Reference

References

[1] Jean-Philippe Bauchet and Florent Lafarge. Kinetic Shape Reconstruction. *ACM Transactions on Graphics*, 2020.

[2] The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 5.6.1 edition, 2024.

[3] Mulin Yu, Florent Lafarge, Sven Oesau, and Bruno Hilaire. Repairing geometric errors in 3D urban models with kinetic data structures. *ISPRS Journal of Photogrammetry and Remote Sensing*, 192, October 2022.