Report, Kinetic project

Dorian Geraldes Pereira, Axel Demuth

March 2024

Contents

1 Objectives

To be able to solve equations on a mesh, we need it to be watertight.

The objectives of the project are:

- Repair mesh to make them watertight
  - watertight building model
  - watertight urban model

In our project we will have to use files in IFC format containing building meshes that are not watertight in an algorithm repairing geometric error in a kinetic data structures. We will have to find a way to keep the label on the surfaces in the algorithm or relabel every surfaces at the end. we can see here an example of what we will need to do:
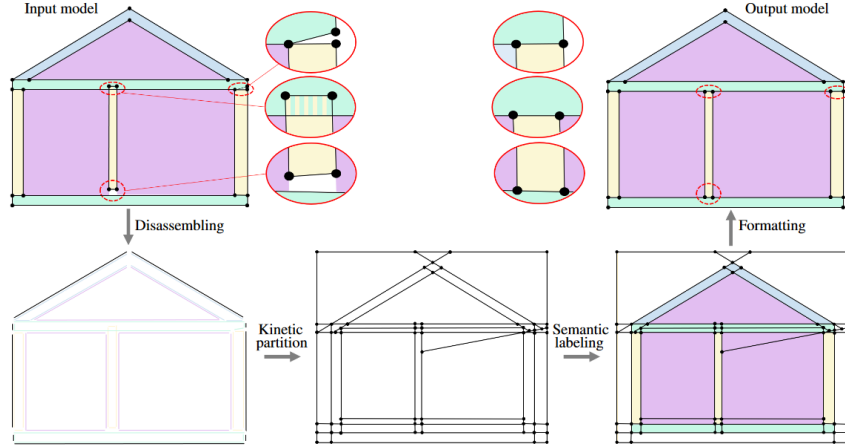

2 Tools

2.1 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.

2.2 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:
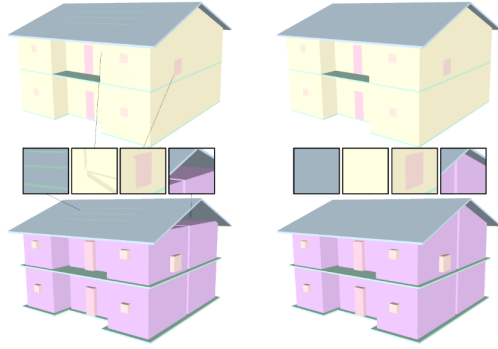
To utilize the algorithm, we employ one from CGAL. Given a set of parameters and a file containing a point cloud along with the associated normals to the points, the algorithm is applied. We were fortunate to have a meeting with Florent Lafarge, one of the creators of the algorithm, who explained to us which parameters are crucial for analysis and how each one can significantly influence the results. Two parameters stand out as particularly important:

- 'dist' indicates the distance between two points required to consider them for plane construction.

- 'pmin' represents the number of points used to construct a plane.

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.

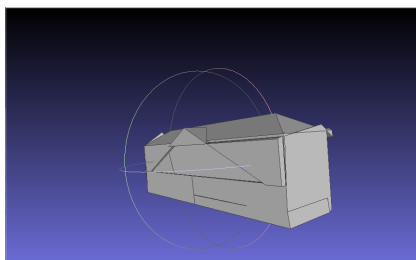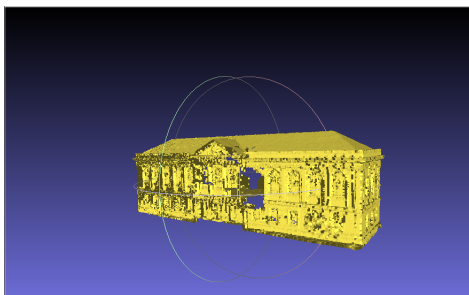The original point cloud represents this building:
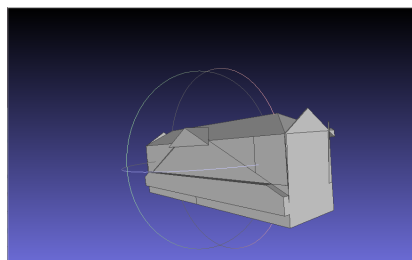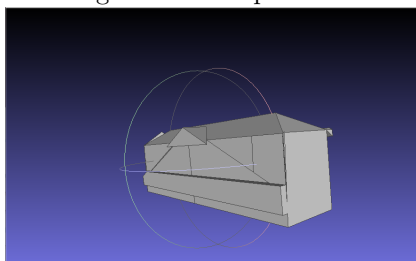

Figure 1: dist1_pmin220
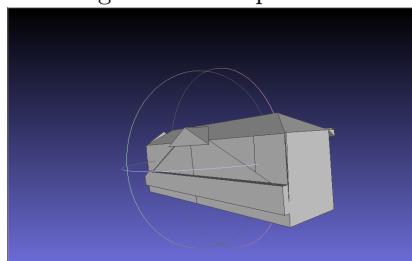

Figure 2: dist1_pmin250

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
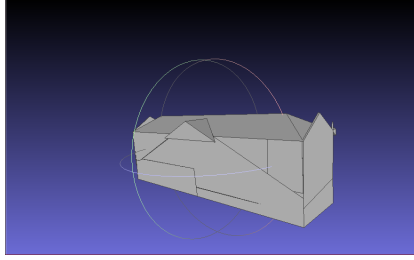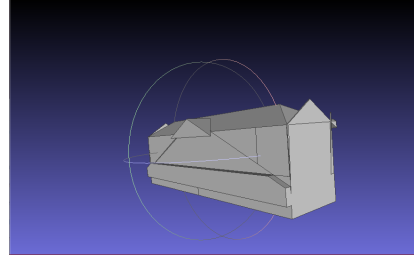
Figure 5: dist15_pmin250.png
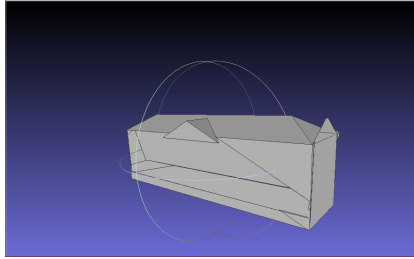

Figure 6: dist1_pmin250
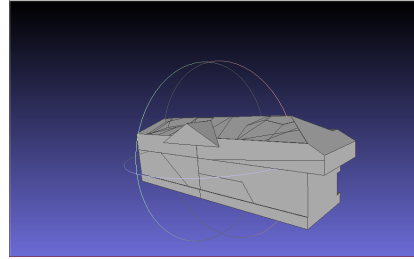

Figure 7: dist03_pmin250
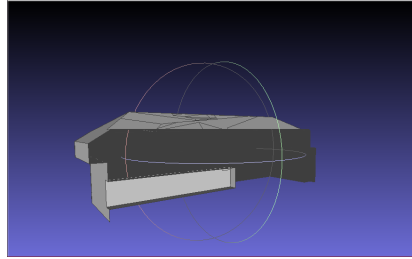

Figure 8: dist001_pmin250

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.

2.3   Roadmap

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