Report, Kinetic project

Dorian Geraldes Pereira, Axel Demuth

March 2024

Contents

1 Project context **2**

2 Project Objective **2**

3 Current Project Challenges **2**

4 Tools **3**
4.1 CGAL 3
4.2 IFC project 3
4.3 Kinetic 3

5 Implementation **8**

6 Analysis of Results **8**
6.1 Visualization of Results 8

7 Roadmap **9**

8 Reference **10**

1 Project context

Our project is part of a bigger project with the goal of taking IFC file representing building to create 3D models that will be used for energy simulations with hihgh performance calculations. In this project they need to fill the missing data in the IFC file to be able to run simulations of energy consumption,losses or even conform estimation .

2 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.

3 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.

4 Tools

4.1 CGAL

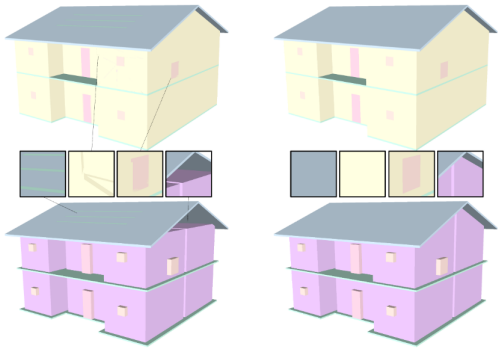
CGAL is a comprehensive package for geometry algorithms, providing various data structures and algorithms for working on polygons, surfaces, mesh generation, and more. It offers a wide range of functionalities for geometric processing and analysis in various fields such as computer graphics, computational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

4.2 IFC project

The initiave was create by buildingSMART(BIM),which goal are to support portability between software in construction sector.IFC is a file format to replace a framented information systeme to a normalise one.So every actor in the sector can use IFC compatible software to open file and modelise what's within and don't need to waste time and calcul time to traduct from a format to one they support. IFC format is a base object oriented code, with a lot of class associated to any type of needs, like construction site, tools, materials. The probleme we have is it's hard to read IFC object to use them in Kinetic program. They are bunch of software already traducting IFC to the format we want, but we loose lot of information. We will need to work on the object we receive from the traduction to make Kinetic algorithms to read the files.

4.3 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:



We can get a good description about How it's work in general with the report from INRIA [3]

- It's use geometric primitives to create planes because of it's relevance for human made environnements
- There are multiple methode to do plan fifting in a 3D modele as Neural Network archictectures or Energy bases Models The algorithm use the second one.
- To calculate the quality of a primitive configuration x with and an energy U of the format
  $U(x) = w_f U_f(x) + w_s U_c(x) + w_c U_c(x)$
  where all U function are for fidelity,simplicity and completeness of the function and all w are positive watertight
- Them it use geometric operation like merging,splitting,transfer,insertion and exclusion on differents planes and close primitive.

Here we have a speudo code about how the exploration of the set of primitives works

Algorithm 1 Pseudo-code of the exploration mechanism

```
Initialize the primitive configuration x
repeat
    Initialize the priority queue Q
    while top operation i of Q decreases energy U do
        Update x by operation i
        Update Q
    end while
    Update x by the global transfer operator
until no update modifies x any more
```

The priorrity queue is the set of primitives which we apply the geometric application on.

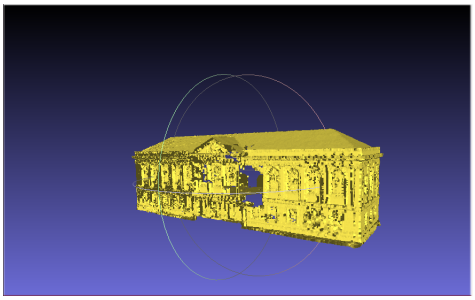
To utilize the algorithm, we employ one from CGAL. Given a set of parameters and a file containing a point cloud along with the associated normals to the points,

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:



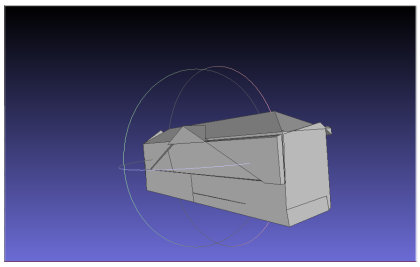


Figure 1: dist1_pmin220

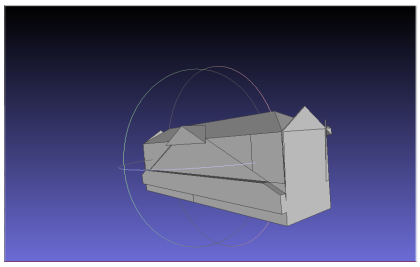


Figure 2: dist1_pmin250

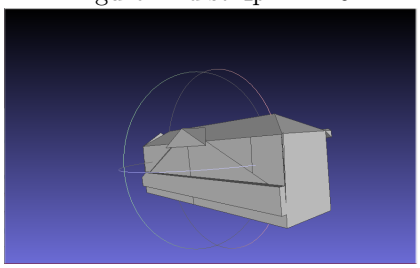


Figure 3: dist1_pmin280

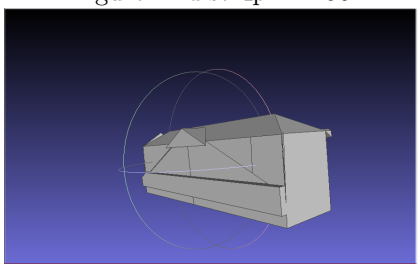


Figure 4: dist1_pmin300

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

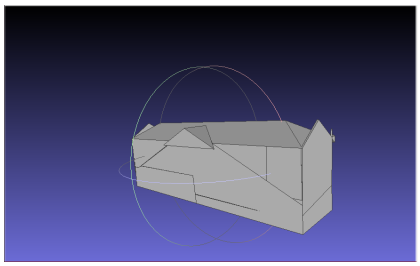


Figure 5: dist15_pmin250.png

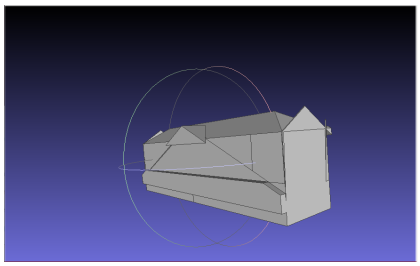


Figure 6: dist1_pmin250

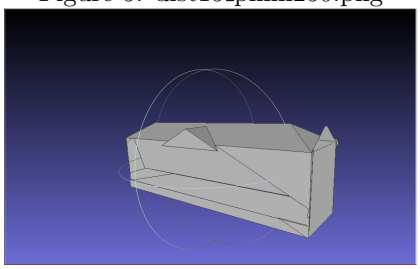


Figure 7: dist03_pmin250

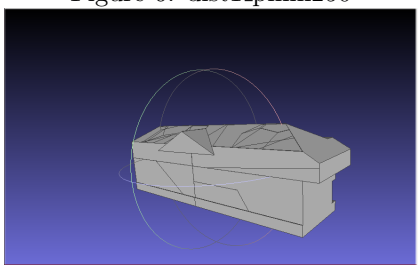


Figure 8: dist001_pmin250

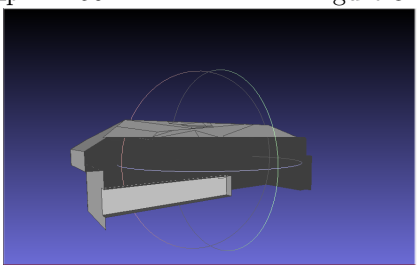


Figure 9: dist001_pmin250

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.

5 Implementation

We are using the most trouve mot example of the cgal kinetic algorithm as base for our main. In this main we can change every parameters of the algorithm.
To use the algorithm you need to give him a file from supported format such as .ply , .off, .xyz, .las . If the parameters dont respeced the critera we described in the previous section the algorithm can end up in segmentation fault. The result of the algorithm will be in an .off format in "resultat" folder.
We can present an example of the command to execute the program :
" ./build/default/bin/kinetic -data data/flame.ply -dist 0.3 -minp 50 -regangle 5 "

6 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.

6.1 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

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.

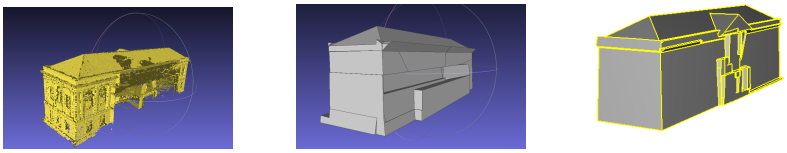


Figure 10: Visualization of results with a building

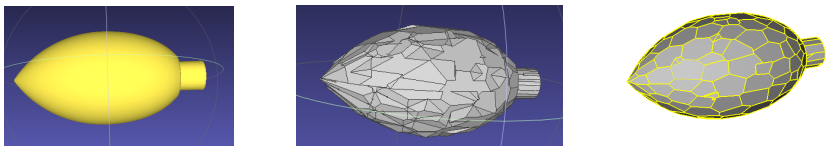


Figure 11: Visualization of results with a flame

We plan to integrate this data into our upcoming experiments to evaluate the performance of the algorithms on real models of buildings and cities.

7 Roadmap

As of now, we have work to do in order to utilize the Kinetic algorithm in our environment and understand how to apply it correctly. In the following, we will need to work on the following points:

- Convert PLT or MSH meshes into a format compatible with the Kinetic algorithm.
- If possible, be able to retrieve the labels on the mesh generated by the Kinetic algorithm.
- Quality check of the obtained mesh from the kinetic algorithm (same volume as the first mesh)
- Apply physical theories to the base mesh and the output mesh to analyze the differences between them.
- Investigate the feasibility of applying the algorithm on a larger scale, such as a city or a large building, instead of a basic one.

for the V1 we modified and completed some issues :

- The Issues Create watertight mesh with kinetic we completed what we wanted to do with this issue but now we want to go further so we modified it to create watertight mesh from files coming to IFC format for the V2
- The issue Convert IFC to .ply format was also changed to convert MSH and STL to supported cgal format for the V2

- The issue setup programming environment for CGAL with github action and submodul,the creation of the main program,the study of the kinetic program were completed

8 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 and Florent Lafarge. Finding Good Configurations of Planar Primitives in Unorganized Point Clouds. In *CVPR 2022 - IEEE Conference on Computer Vision and Pattern Recognition*, La Nouvelle-Orléans, United States, June 2022.

[4] 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.