Straightening and simplifying a multi-view stereo mesh of a city

Msc Geomatics Thesis project plan

Yuxuan Kang

January 8, 2017

1 Introduction

3D city models are getting popular with the developments of both hardware and software. There are more and more means of collecting, storing, processing and using data. Instead of 2D maps, 3D models give more scaled, detailed, advanced representation of the city. Together with texture information, 3D models also provide more realistic visualization (Biljecki et al. 2015, Brenner, Haala, and Fritsch 2001).

There are varieties of representations for 3D modelling such as b-reps, voxel, SCG (Zlatanova, Abdul Rahman, and Shi 2004, Lee and Zlatanova 2008) and mesh is one of the most popular for representing 3D city model. A mesh is the collection of vertices, edges and faces. The faces often consist of trianges thus called triangle mesh. The quality of the mesh has high influence on the representation of city model. Low quality meshes (e.g Meshes with noises, inaccurate positions of vertices etc.) will weaken the effect of realistic visualization, the simplicity of storing and accuracy of the objects. As a consequence, 3D models lose some advantages and sometimes they will be even worse than 2D representations due to more complex data structure, more data storage etc. So it is essential to ensure the quality of the mesh.

A multi-view stero mesh is generated by image matching methods, which are widely used in photogrammetry and computer graphics (Gruen 1985). By matching two overlapping images together with the camera infomation, the coordinates of each pixel on the images can be acquired. This method is productive and efficient however the quality of the data can easily decrease if the orientation of the camera is bad or there are obstacles between the camera and the objects etc. Because these errors propagate from data collection to the final meshes, the generated mesh may have unsmooth surfaces with spikes. Figure 1 shows a low quality mesh for a part of Amsterdam.

A high quality city model should describe objects in a realistic way, especially artificial objects like buildings and facilities. Instead of rugged surfaces shown in Figure 1, the edges and surfaces should ideally be straight and regular. Thus it is necessary to carry out a method to straighten the meshes. A similar research of straightening the mesh was carried out by Jonsson, Figure 2 shows the result of the research (Jonsson 2016). This Msc project will focus on the same goal but use different method described in Section 4.

Moreover, high density meshes can describe a model in a detailed way, it also requires large space of data storage. Figure 3 shows the difference between meshes with different densities. It is clear that high-density mesh models the object more smoothly and detailedly. With the decrease of the density, the model begins to lose details.

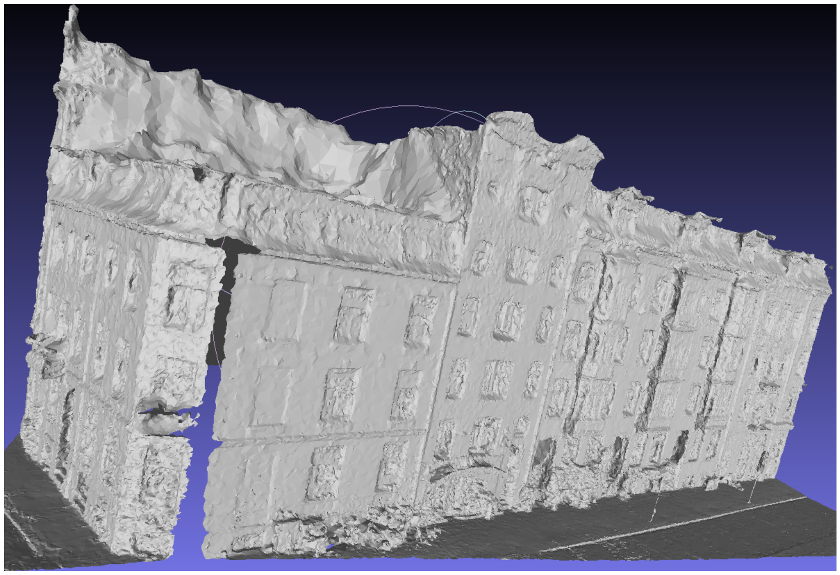
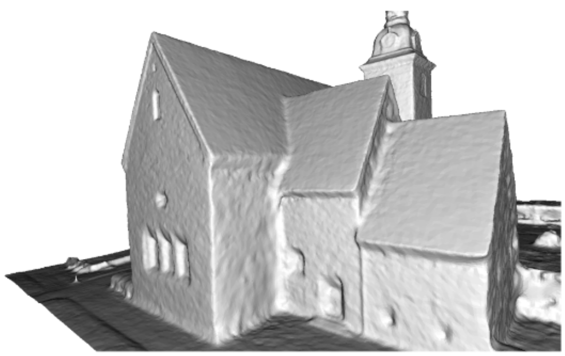
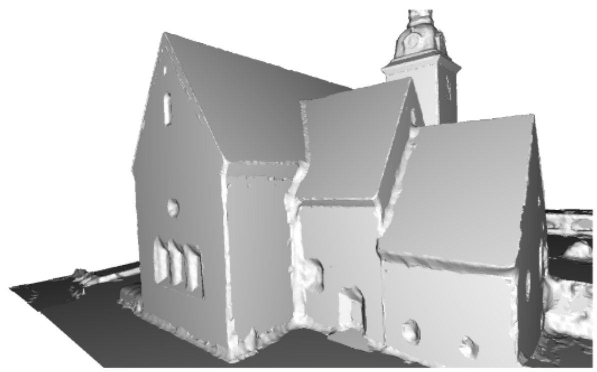


Figure 1: Mesh with low quality



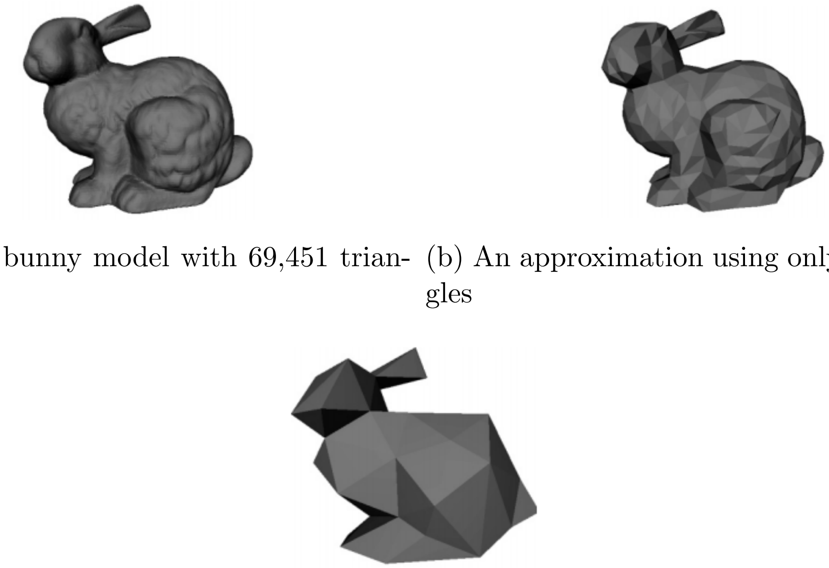
(a) Original Vreta Church mesh



(b) Vreta Church mesh after flattening

Figure 2: Mesh straightening (Jonsson 2016)

For a city model, the artificial objects are often regularly shaped with straight lines and planes, it is unnecessary to use high density mesh to model relatively flat surfaces like walls. In order to save storage space and speed up display, mesh can be simplified without losing much details by keeping key features of the mesh. As shown in Figure 4, the mesh can be simplified while keeping some features of the buildings like windows. By this means, the data redundancy will decrease while the

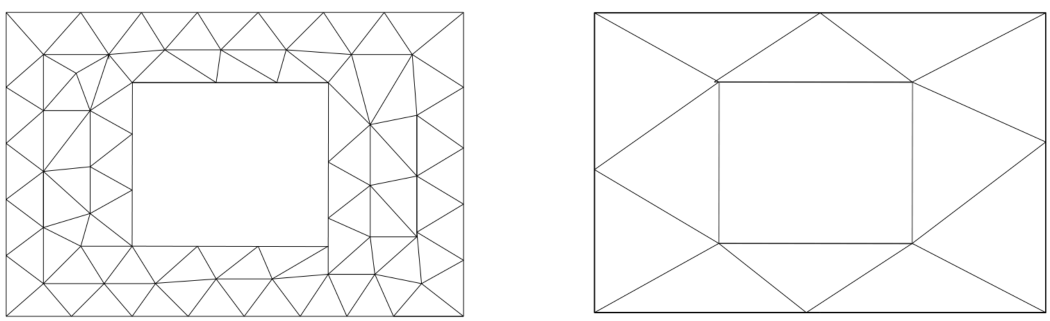


(a) Original bunny model with 69,451 triangles

(b) An approximation using only 1,000 triangles

(c) An approximation using only 100 triangles

Figure 3: Mesh simplification (Garland and Heckbert 1997)



(a) High density mesh with window feature

(b) Low density mesh with window feature

Figure 4: Mesh simplification while keeping details

shape of the object will retain.

In general, this Msc project will mainly focus on these two aspects: 1. mesh straightening 2. simplification of the mesh. It is expected that the quality of the city model will be improved by solving these topics.

2 Related work

A similar research of straightening meshes was carried out in 2016 by Jonsson. It aimed at the same goal as this Msc project. His method mainly consists of two parts: 1. plane detection 2. plane correction (Jonsson 2016). The general workflow is as follows:

1. Vertices labelling. With principal curvature values for each vertex on a mesh, a label describing a local surface property to each vertex is assigned.

2. Region growing. Based on the properties of each vertex, neighboring vertices with the same label can be clustered by region growing.

3. Plane fitting. Planes can be fitted to the clusters.

4. Plane refinement. Plane refinement involves many operations, i.e plane growing, merging adjacent and similar, dividing planes into clear substructures.

After these steps, some refined planes are obtained. The vertices which belong to one plane will be projected to the corresponding plane, thus the mesh will be flattened. The figure 2 shows the result of this method.

In the research of indoor scanning with Tango tablet, RANSAC algorithm is used for extracting planar patches from dense meshes generated by Tango (Diakité and Zlatanova 2016). The result of the plane fitting provides an idea that RANSAC is capable of detecting planes from a mesh. Figure 5 shows the result of plane detection on indoor scanned mesh. This method can be extended so that the extracted planes can be used for mesh straightening. The principles of RANSAC algorithm and the method of combining plane extraction and mesh straightening will be discussed later in section 4.

In order to get good quality city model, fitted planes should be under some rules so that the quality of the fitted plane can be guaranteed. Kada and Wichmann did some researches about using different kinds of rules to constrain the planes. Instead of directly using the fitted planes, they refine these planes based on the predefined rules so that the model will be more accurate and consistent. They proposed a feature-driven 3D building modelling method (Kada and Wichmann 2013). There are normally two kinds of models: model driven and data driven. Model driven will define building models in the library while data driven on the contrary, will form surface models without shape restriction and add adjacency information. Instead, Kada and Wichmann used feature driven which is between model driven and data driven. It defines low-level feature with building knowledge and uses one half-space

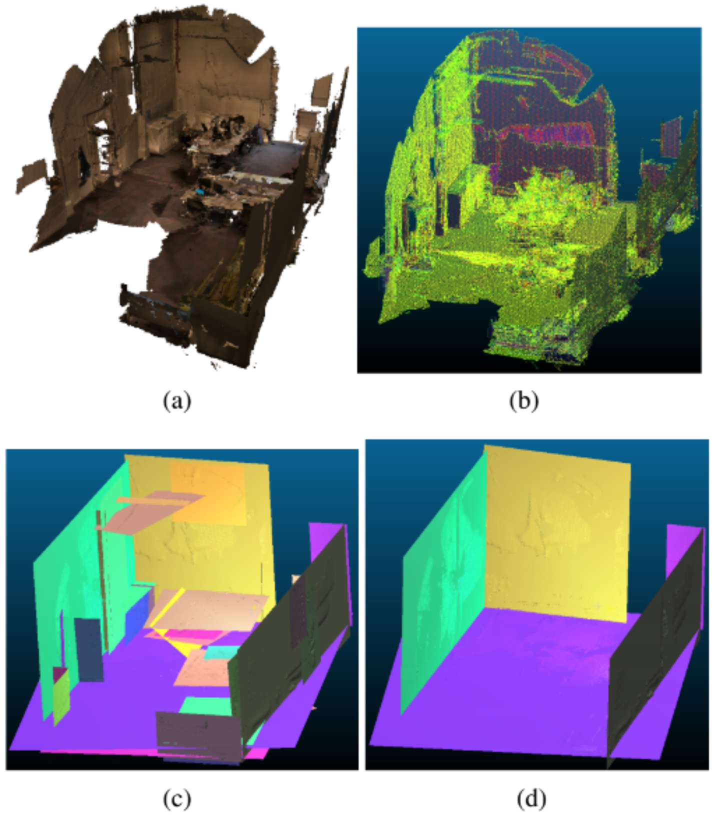


Figure 5: Plane extraction (Diakité and Zlatanova 2016)

which is represented by a parametric plane for every segment. The main methodology can be concluded as:

1. Sub-surface segmentation, enlarge segmentation with virtual points which fit the segment but are located below real surface points. This sub-surface segmentation will make the segment more complete, which is helpful for intersections and setting up topology relations such as adjacencies.

2. iterative rule-based feature recognition

3. 3D building model construction, using half-space modeling.

For the second step rule-based feature recognition, there are several kinds of rules applied:

- Geometric criteria: including orientation of segments, height differences in the proximity of features, lengths of possible segment intersection lines and distances between points, lines and segments.
- Topological criteria : 2D and 3D adjacency and topology relations of segments.
- Locational criteria: location of segments, i.e if segments of a dormer are above roof segments.
- Semantic meaning of features which are already recognized.

The feature recognition is a interative process and the thresholds can change adaptively. If no more features can be detected without relaxing thresholds, the segments will be split. A segment needs to be split, if it is part of the surface of a convex building component, but also has surface points that are outside this component.

Based on the research of feature-driven 3D building modelling, Wichmann and Kada enhanced the method by introducing half-space adjustment(Wichmann and Kada 2014). The adjustment settles after iterative feature recognition before 3D model reconstruction. The adjustment can be divided into two categories:

- Local half-space adjustment. Local adjustment only focuses on a building component (i.e facade and roof) and not on other half spaces. The aim of this adjustment is to refine the model of a single building. Roughly speaking, there are three steps: slope adjustment, orientation adjustment and position adjustment.
- Global half-space adjustment. The aim of the global half-space adjustment is to adjust the half-spaces of building components so that the half-spaces will be more regularized in a global scope.

By applying several new regularization rules, the accuracy and the shape of building models will be improved.

3 Research objectives

3.1 Research questions

My research has several stages, corresponding to different problems in this project. The main research question is:

Can plane fitting methods yield better result than exisiting approach for mesh straightening and simplification?

The main aim of this research is to explore the capacity of plane fitting method in improving the quality of the meshes. Moreover, exploring whether the improved mesh can be simplified is also part of the topic. In the first stage, the main research question will be explored. Based on the result from the first stage, the simplification will be carried out on the improved mesh.

In order to achieve these goals, there are some sub-questions that need to be explored as well:

- What constraints can be used for fitting planes and how can plane constraints be used for straightening meshes?
- How can some other information for example texture, geometry and topology information be involved to improve the algorithm?
- How to keep textures to simplified meshes?

3.2 Research scope

The research mainly focuses on artificial objects, especially the facades of the buildings. The feasible solution should output a improved mesh with smooth and straight surfaces (i.e meshes with much less spikes).

4 Methodology

Based on the research questions, there are several stages for this topic. Each data processing is based on previous result. In general, the whole research can be divided into three stages, plane fitting, mesh straightening and mesh simplification. The whole procedures and workflow are shown in the following figure 6.

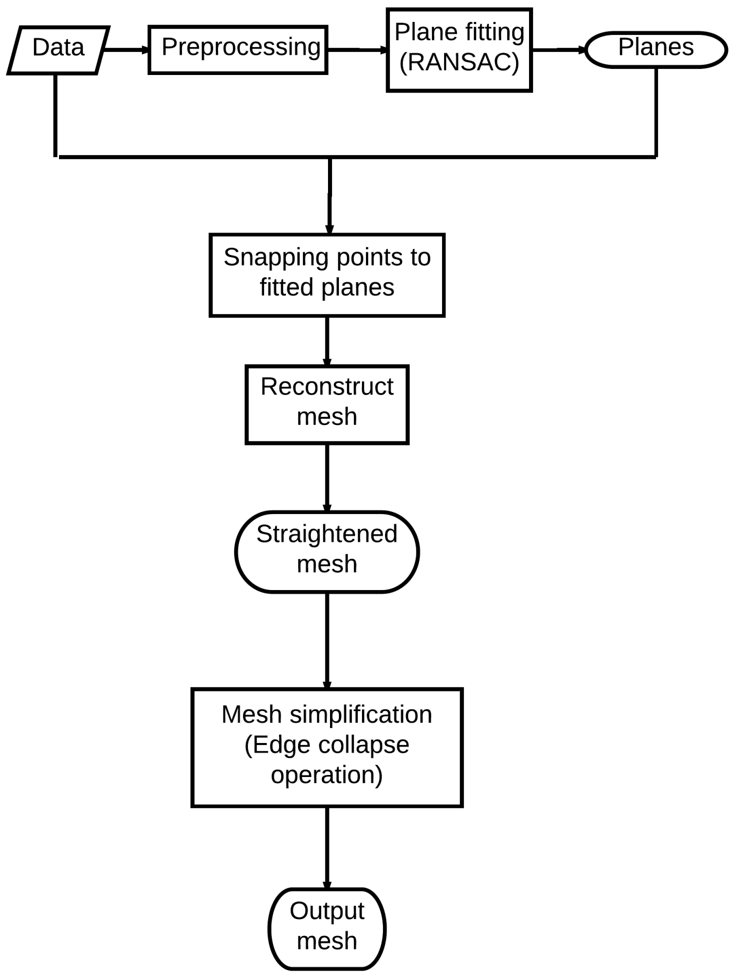


Figure 6: Workflow

Most existing methods put focus on the meshes, trying to find solutions for improving them. These methods tend to work with the vertices considering the topology relations between faces. Instead of working with meshes, this research mainly focuses on dealing with points, which means it will disgard the mesh information at the first stage and fit planes on input points directly.

4.1 Preprocessing

Plane fitting stage includes data preprocessing and fitting planes. In this stage, the original data is preprocessed in order to get normal for each vertex. Considering topology information, each vertex shared by several faces, the normal can be estimated by all the normal of incident faces. The normal of a vetex v_i can be calculated as follows:

$$v_i(x, y, z) = \sum_{j=1}^{n} v_j(x_j, y_j, z_j) \quad j \in \{ \text{ all in incident faces of vertex } i \text{ } \}$$

$$|v| = 1$$

Instead of using some normal estimation method based on points like Principle component analysis, this face-based normal estimation is much faster since a mesh already contains some underlying information of normal, it is much easier to translate normal from faces to each point.

4.2 Plane fitting

In this project, RANSAC algorithm is used for plane fitting. RANSAC (Random sample consensus) algorithm is a method which is able to distinguish "inliers" and "outliers" of the data. Inliers are the data whose distribution can be described by some mathematical model while outliers are the data that do not fit in the model. RANSAC iteratively detects outliers from the data and only estimate parameters of the model based on inliers. Figure 9 demonstrates the principle of RANSAC. The blue points are inliers while the red points are outliers. Compared to least square aujustment which will consider all points into the model, RANSAC has more advantages when there are many noise and outliers in the dataset.

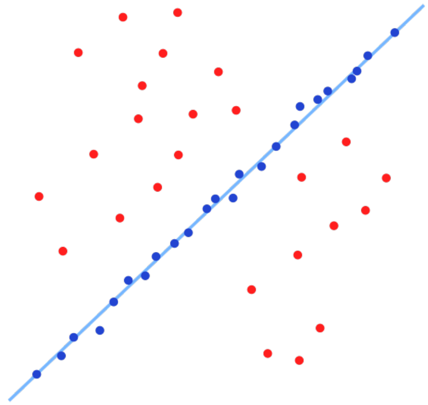
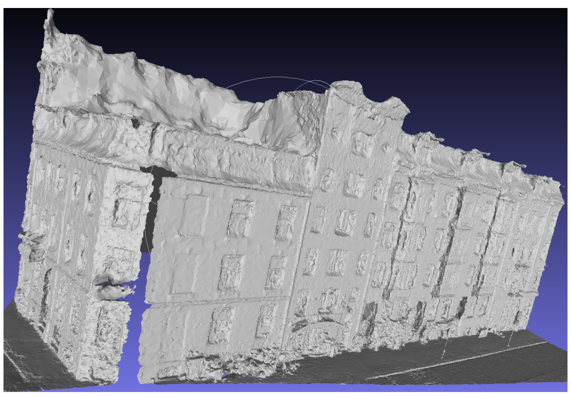
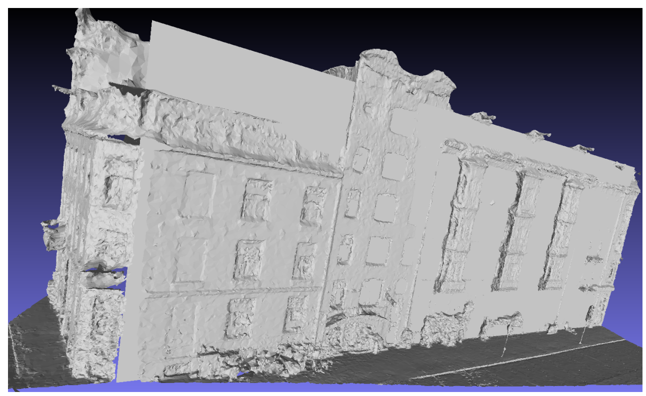


Figure 7: RANSAC algorithm



(a) Original mesh



(b) Mesh with a plane fitted by RANSAC

Figure 8: Preliminary result

In the case of this research, the original data are the city models. RANSAC can automaticlly detect points on facade as inliers because these points can best model planes. The rest points that do not contribute to plane fitting will be regarded as noise or outliers. However, the performance of RANSAC is highly dependent on the parameters such as the minimal number of points used for fitting a plane and the angle between the normal of the vertex and the normal of the fitted plane etc, thus the automatic choice of the parameter is also a challenge. Figure 8 shows the preliminary result of plane fitting. The plane is infinite but is visualized with finite rectangle. The preliminary result shows the potential capacity of plane fitting in

Figure 9: Texture information of the mesh

straightening the mesh. With the proper parameters, fitted plane can match the city model well.

4.3 Mesh straightening

After this procedure, mesh and plane models can be combined. The points can be snapped to plane model obtained from RANSAC algorithm. After this step, new mesh should be constructed. The mesh can be reconstructed based on the indices of original mesh. By snapping points to the planes, each point as inlier for plane fitting will be corrected to the plane model. Besides, some additional information like texture might be used in this procedure, i.e hypothetically window points are distinguished from wall points based on texture information, they will be set to different classes. Then it is unnecessary to snap window points to the planes, since it is observed that window points have some offsets from the walls. Some details can be kept by this means. Even window points can be dealt separately later in order to get a building model with better shaped windows.

4.4 Mesh simplification

After the mesh being straightened, a simplification of a mesh can be explored. In this stage, the research will focus on verifying whether mesh can be simplified without losing much quality in this case. Cleaner data will reduce the complexity of the data, the space for data storage and speed up the data display. There exist some methods for simplying the mesh. Mainly by lowering the density of the vertices. Figure 10 shows some topological operations including vertex removal, edge-collapse and

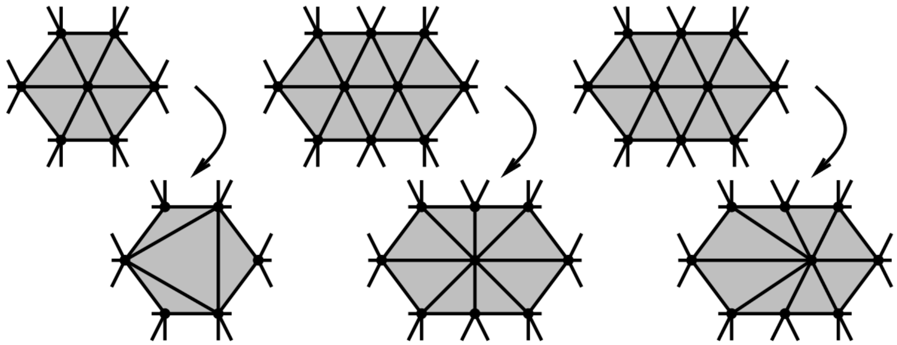


Figure 10: Vertex-removal, Edge-collapse, and Half-Edge-Collapse (Kobbelt, Campagna, and Seidel 1998)

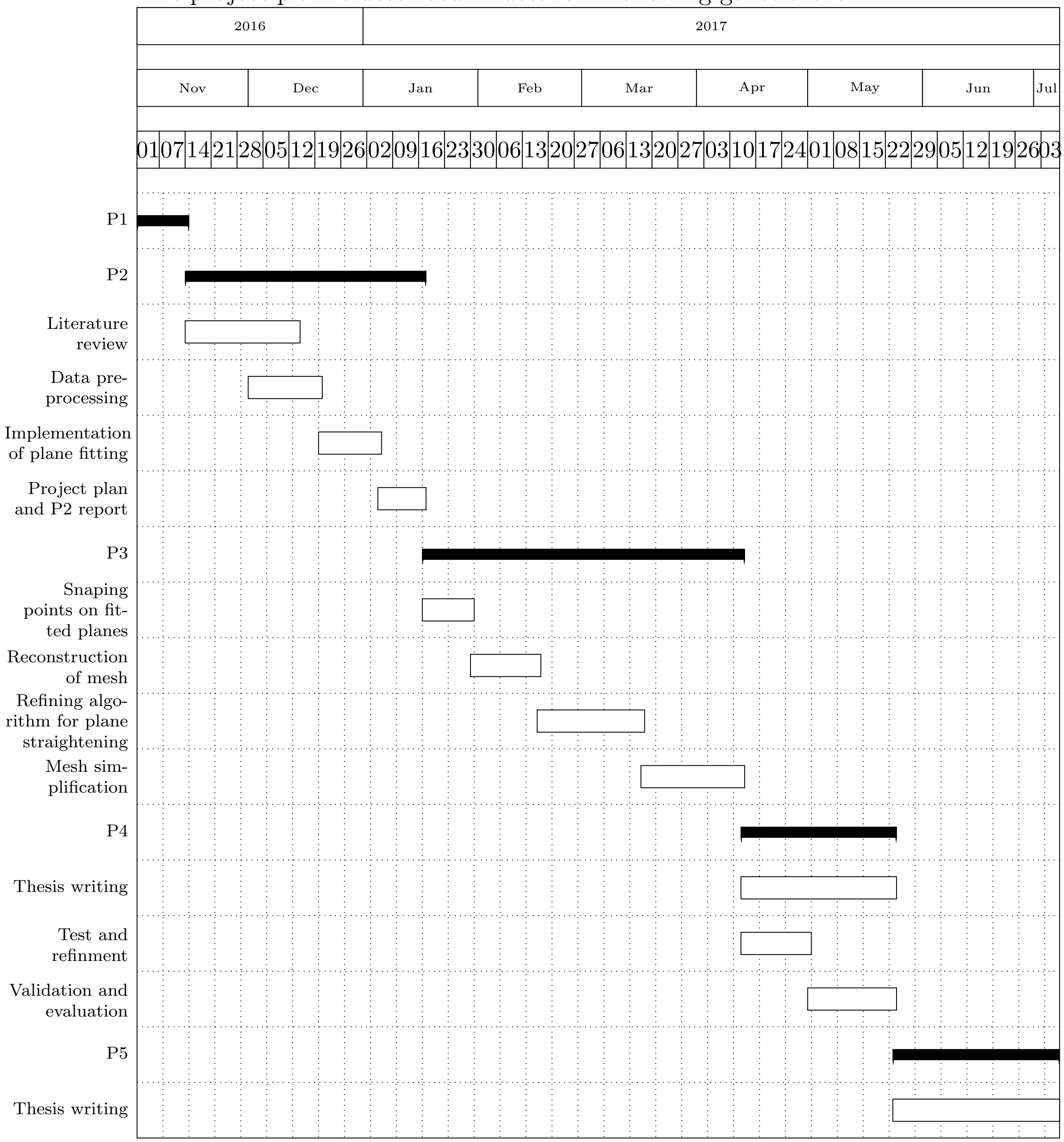
half-edge-collapse operation (Kobbelt, Campagna, and Seidel 1998). Edge-collapse replaces the edge and its two end vertices with a single vertex while half-edge-collapse pulls one of the two end vertices to another, removing edge and leaving the left vertex in place (Cacciola 2015). The simplification of this project will be based on half-edge-collapse operation, calculating the cost for each simplification step. By restricting the cost of each simplification operation, the basic shape of the mesh will be reserved.

4.5 Evaluation

After all the data processing, the quality of the output mesh needs to be evaluated. Since the mesh should be impoved in visulization, the output mesh will be evaluated visually. Ideally, the surfaces of the output mesh should be straight and smooth.

5 Schedule

The project plan is described in details in following gantt chart:



6 Tools and data

6.1 Data description

The data is the city model of Amsterdam and it is collected by CycloMedia Technology. The files are in COLLADA format, and they are converted into OBJ format because OBJ format is widely supported by software and it is a easy format for data read and write.

OBJ file contains information vertices coordinates, texture reference, normal, face indices. Vertices coordinates are structured as "x y z" coordinates. Texture is structured in uv coordinates, it refers to the coordinates of the corresponding images. Normal also has "x y z" vector coordinates and the order of the normal is the same as the vertices so that each normal is related to its corresponding vertex. Faces are organized by the format of vertex coordinate index, texture coordinate index, and normal index.

6.2 Tools

Meshlab is a software which support many 3D data format. In this research, meshlab is used for visualizing meshes and planes. CGAL(The computational Geometry Algorithms Library) is a powerful library which provides various geomatric computation algorithms. In this project, CGAL is used for developing. C++ is the development tool since CGAL is implemented in C++.

6.3 Deliverables

The deliverables of the project are a thesis describing the whole methods and procedures in solving the problem, a C++ project which can be reused to processing similar data (at least data from the same data source), and processed meshes.

References

Biljecki, Filip et al. (2015). "Applications of 3D city models: state of the art review". In: *ISPRS International Journal of Geo-Information* 4.4, pp. 2842–2889.

Brenner, C, N Haala, and D Fritsch (2001). "Towards fully automated 3D city model generation". In: *Automatic Extraction of Man-Made Objects from Aerial and Space Images (III)*, pp. 47–57.

Cacciola, Fernando (2015). "Triangulated surface mesh simplification". In: *CGAL User and Reference Manual. CGAL Editorial Board* 4.

Chen, Jiale (2015). "Edge-aware simplification of roof and facade point clouds into a uniformly dense point cloud". PhD thesis. TU Delft, Delft University of Technology.

Cignoni, Paolo, Claudio Montani, and Roberto Scopigno (1998). "A comparison of mesh simplification algorithms". In: *Computers & Graphics* 22.1, pp. 37–54.

Diakité, Abdoulaye A and Sisi Zlatanova (2016). "FIRST EXPERIMENTS WITH THE TANGO TABLET FOR INDOOR SCANNING." In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 67–72.

Garland, Michael and Paul S Heckbert (1997). "Surface simplification using quadric error metrics". In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., pp. 209–216.

Gruen, Armin (1985). "Adaptive least squares correlation: a powerful image matching technique". In: *South African Journal of Photogrammetry, Remote Sensing and Cartography* 14.3, pp. 175–187.

Hoppe, Hugues (1996). "Progressive meshes". In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, pp. 99–108.

Hoppe, Hugues et al. (1993). "Mesh optimization". In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. ACM, pp. 19–26.

Jonsson, Mikael (2016). *Make it Flat: Detection and Correction of Planar Regions in Triangle Meshes*.

Kada, Martin and Andreas Wichmann (2013). "Feature-driven 3D building modeling using planar halfspaces". In: *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, pp. 37–42.

Kobbelt, Leif, Swen Campagna, and Hans-Peter Seidel (1998). "A general framework for mesh decimation". In: *Graphics interface*. Vol. 98, pp. 43–50.

Lee, J and Sisi Zlatanova (2008). "A 3D data model and topological analyses for emergency response in urban areas". In: *Geospatial information technology for emergency response* 143, p. C168.

Pauly, Mark, Markus Gross, and Leif P Kobbelt (2002). "Efficient simplification of point-sampled surfaces". In: *Proceedings of the conference on Visualization'02*. IEEE Computer Society, pp. 163–170.

Salinas, David, Florent Lafarge, and Pierre Alliez (2015). "Structure-Aware Mesh Decimation". In: *Computer Graphics Forum*. Vol. 34. 6. Wiley Online Library, pp. 211–227.

Sander, Pedro V et al. (2001). "Texture mapping progressive meshes". In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, pp. 409–416.

Verdie, Yannick, Florent Lafarge, and Pierre Alliez (2015). *LOD Generation for urban scenes*. Tech. rep. Association for Computing Machinery.

Wichmann, Andreas and Martin Kada (2014). "3D building adjustment using planar half-space regularities". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2.3, p. 189.

Zlatanova, Sisi, Alias Abdul Rahman, and Wenzhong Shi (2004). "Topological models and frameworks for 3D spatial objects". In: *Computers & Geosciences* 30.4, pp. 419–428.