Here we propose a 3D simplification method aimed at reducing storage, transmission and model rendering costs. In contrast with state-of-the-art algorithms, our proposal simplifies the 3D model during the building stage, simplifying the process pipeline and reducing the computational complexity related to the 3D modeling process. Practical results discussed in this chapter show that, while highly reducing the complexity of the 3D model, the method has a minimal impact on the representation accuracy.

5.1 Introduction

Scene reconstruction algorithms approximate the shape of the scene using 3D features such as vertices or lines. These features can be seen as discrete measurements of a continuous model representing the scene. Clearly, the higher the number of the 3D features, the higher the accuracy of the scene structure estimation.

When navigation and mapping algorithms have to deal with large areas, however, the amount of data may prove overwhelming. This is especially the case when Kalman Filter or Global Alignment algorithms are used, in which the complexity of the problem grows with the square of the number of scene features.

The solution to this problem is reducing the number of extracted scene features. The difficulty stands in selecting the 3D features in a way to minimize the impact on the precision of the resulting scene model.

The problem of reducing the complexity of 3D models while maintaining the model precision has been studied by the computer graphics community where it is known as mesh simplification. The state of the art in mesh simplification includes a wide range of alternatives. In [95, 147], the authors divide the 3D volume into a user-specified grid. Then, the model is simplified by removing all vertices within a grid cell, maintaining only the most representative vertices. Schroeder et al. [157] use a multiple pass simplification, based
on a user specified user error threshold. Eck et al. \cite{30} use a wavelet-based approach to create a fast multi-resolution representation of the original surface. A similar multi-resolution approach is employed by Progressive Meshes (PM) \cite{70,71}, a widely used method in real-time 3D rendering. Other authors have proposed the use of color and texture information in addition to the shape in the simplification criteria \cite{54,57,181}, minimizing visual aliasing due to model simplification.

From the point of view of the model simplification strategy, two general approaches should be underlined:

Local Simplifications Algorithms. Define a mesh operation that only affects a small set of its elements and produces a new mesh with fewer elements. This operation is associated with a cost function measuring the error introduced by the local simplification operation, which allows applying first the operations introducing less error. The process repeats the simplification operation until the user requirements are fulfilled, which normally consist in a maximum error at the cost function, or a desired number of faces. In \cite{157}, Schroeder et al. proposes a method that iteratively deletes vertices while tessellating the resulting holes. This method does not change the topology and it is applicable to non-manifold surfaces (though will not simplify near non-manifold vertices). In contrast, edge contraction approaches converts edges into single vertices. Such methods can change the topology of the model (e.g., contract edges repeatedly around a hole may eventually close it), and is applicable to non-manifold meshes. This method was first proposed in \cite{72}, with many of its variants proposing different alternatives for defining the vertex-to-edge association cost function. The quadric error metrics approach of \cite{53}, for example, offers a very good trade-off between geometric accuracy and computational cost. Here, the error at a vertex is described using a $4 \times 4$ matrix $Q$ that represents the sum of squared distances from a vertex to the planes defined by the neighboring triangles as $v^TQv$. In order to update the error metric of the vertex resulting after edge collapse, the quadrics $Q$ of the original vertices are directly summed. This simple updating operation makes the method achieve a nice computational cost. Other authors propose methods that define the error associated with the simplification in terms of pixels. In \cite{90}, the simplification is guided by minimizing the deviation from the rendered screen space mesh, that is, minimizing the visual artifacts.

Global Simplifications Algorithms. While less significant than the local strategies, there are some approaches that define the simplification problem in a global manner. Vertex clustering methods discretize the space into a regular voxel grid. Vertices from the original mesh falling into the same voxel are clustered to a single one, using some heuristics. An example from this category is \cite{148}. Despite easy to implement, vertex clustering can severely distort the topology of the mesh. On the other hand, using shape approximation methods, the surface simplification problem