Chapter 1

Introduction

1.1 Over view of the simplification

Polygonal surface models are used in several computer graphics applications for display and simulation purposes. In such a scenario where it is normally expected that, designer moves, adds, or deletes some parts interactively; it is required that the model representation should be efficient or approximation techniques must augment the capabilities of the graphics hardware [93]. These models have large set of polygons; and to get more realism more numbers of polygons are required. But large amount of polygons i.e. large detail will challenge the hardware capability and lead to increase in the processing time and transmission time. A focus is required to address this problem and propose a method to simplify the high detailed polygon model into much simplified model. Simplified model should contain fewer numbers of polygons without compromising the quality of original model.

Polygon models of 3D objects are preferred for rendering techniques. In practice a major stumbling block in polygon models is the complexity and size of the data required to keep the topological information. It should be noted that, polygon models are linearized form of non-linear reality. Thus, the number of tessels required to represent the 3D object with clarity, sharpness and smoothness increases without bounds. This leads to the problem of complexity control of the models [14, 86].

In medical, scientific, entertainment, and computer aided design systems; polygonal models are commonly used for representation. In all these applications and many more, highly detailed and complex models are generally required. But to
achieve acceptable running times and hardware capacity, simpler model should substitute the original detailed model. Recent work on polygonal simplification algorithms has focused on this goal. Different methods have been proposed in this context, like vertex clustering, polyhedral refinement, region merging, and wavelet decomposition, decimation of vertex and contraction of edges [20, 42, 44, 77, 82]. Care should be taken to find the cost of each edge and the least cost edge should be contracted first. Some predominant geometry like boundary features should not be simplified in order to sustain the basic quality of model. An error measuring techniques may be used to estimate the quality of the model at every step of simplifying algorithm [28, 34, 43, 45].

Sometimes it is required to apply some method to judge the quality of the new output with respect to the previous one. Hence the focus of this work is to generate fast and easy method that measures the error in the model while maintaining the quality (less compromise in quality). The error measurement is a measure of differences between two polygonal models. Small error between two models means, they are very much similar to each other. An error measurement is usually required to define as the geometric distance between an original and a simplified model. Some error measurement combine other attributes – colour, normal, and texture values apart from the geometric details to ensure or differentiate the appearance during the manipulation of 3D models [51]. Because of the dependency of human vision on intuition and limitations of human persistence of vision a mathematical model is always preferred to be used as an error metric. Human metric system is not very reliable and fails after certain limit. Thus a mathematical method is required to
evaluate the error between models with the same accuracy and efficiency all the time. A consistent and quantitative method is required to get a consistent and accurate result again and again. Use of a good measuring error will always improve the final quality of the simplified model. In case of displaying an object or a scene with multi-resolution rendering to vary the quality level for faster processing and reduced computational overhead. Checking the error of every level of detail of every object with respect to the viewing angle can help to represent the things better. This approach applies not only to discrete per–object (Level of Detail) LODs, but also to dynamic view-dependent level of detail of an object or environment. In this case the error can be used to reduce the quality of the model with increasing distance of changing viewing angle [21, 50].

Generally simplification algorithms involve iteratively reducing the data complexity of the models. After simplifying the model, error in the model is measured in order to evaluate the simplification. With this in mind, it is very much important to understand whether we are measuring and optimizing the incremental error or the total error. The incremental error is the error between quality of the model being simplified and quality of the model that results from a simplification operation. The total error is the error between the quality of the original model and the quality of the model resulting from a simplification operation. In many cases, the incremental error is more efficient to measure and optimize [17].

Modelling of any object is done for designing purpose, animation purpose, scientific purpose, architectural purpose etc., It is required to concentrate about the desired complexity, quality and realism of the model while modelling is done. But it
has been observed that, simplification of the model while modelling is done in order to meet the practical conditions of the available hardware capabilities. It is very much required the simplify and measure the error to the optimum level of data complexity without exporting the model to the other tools or software. It needs conversion of the model to the compatible format of the tool or software which does the simplification and error measurement. As this is time consuming, it is highly desired to have this simplification and error measurement while modelling itself to optimise the working time.

1.2 Objectives of the work

The objectives of the thesis are as follows:

1. To create a faster and easy simplification that can produce simplification of the 3D original polygonal model from highly detailed one to simplified one by retaining the quality. It is proposed to use edge contractions to the average position for reduction of the complex polygonal data. It is proposed to use area weighted quadric error metric for retaining the quality of the model while simplifying. It is also proposed to simplify polygonal models in sequential edges removal with out concerning to retain the quality of the model for comparison purpose. This is planned to implement through MAYA API programming.

2. To create a Maya plug-in for faster evaluation of the simplified model by considering mean geometric error – average distance travelled by the vertices as a quality parameter.
3. To create faster and easy optimisation of the 3D polygonal complex original models using optimised edge contractions and quadric error metric for retaining the quality of the model. It is also proposed to simplify polygonal models randomly with out trying to retain the quality for comparison purpose. This is planned to implement through creation of MLL (Maya link library) in VC++ and use as a plug-in in Maya for optimising the speed and memory requirement.

4. To create a Maya plug-in for faster evaluation of the simplified model by considering hausdroff distance error - maximum distance travelled by any of the vertices as a quality parameter.

5. To analyze the incremental error in the model during simplification process that will help in optimising the simplification process.

1.3 Organization of the thesis:

This thesis contains 7 chapters. The details of these chapters are as follows

Chapter 1: In this chapter, section 1.1 gives the overview of the simplification; Section 1.2 presents the objectives of the work; Section 1.3 gives the organization of the thesis.

Chapter 2: This chapter provides background and describes related research efforts. Section 2.1 presents introduction of the chapter; Section 2.2 presents back ground of the simplification. Section 2.3 gives different simplification approaches. Section 2.4 describes different error measuring metrics; section 2.5 provides implementations through MAYA API.
Chapter 3: In this, Section 3.1 gives introduction of this chapter; Section 3.2 presents about Maya; Section 3.3 presents an overview of Maya architecture that describes how polygonal components are organised; Section 3.4 gives introduction to Maya API programming; Section 3.5 gives procedure for creating and final implementation of Maya plug-in.

Chapter 4: In this chapter simplification algorithm will be presented. It is classified in to two components: iterative mid edge contractions and area weighted quadric error metric. Section 4.1 presents introduction of the chapter. Section 4.2 presents objective of the proposed algorithm; Section 4.3 describes the presentation of the simplification algorithm in detail; Section 4.4 provides calculations related to the area weighted quadric error metrics; Section 4.5 discusses how to evaluate the cost of contraction for overall quality checking purpose along with implementation of the algorithm; section 4.6 gives non quality simplification; Section 4.7 presents experimental results and discussion.

Chapter 5: In this chapter optimized simplification process from an original complex 3D model in to simplified model will be presented. It is classified in to two fundamental components: iterative edge contraction and the quadric error metric with optimized the target position after edge being contracted. Section 5.1 presents introduction of the chapter. Section 5.2 presents the design of the algorithm; section 5.3 describes the proposed simplification algorithm in detail; section 5.4 provides calculation related to the quadric error metrics and optimization; section 5.5 discusses
how to evaluate the cost of contraction for overall quality checking purpose along with the implementation of the algorithm; section 5.6 gives non quality simplification; section 5.7 presents experimental results and discussion.

Chapter 6: In this chapter, an incremental error analysis of simplification will be presented. It is classified in to two fundamental components: simplification and relative error measurement. Section 6.1 presents the introduction of the chapter. Section 6.2 gives the need and over view of the incremental error measurement; Section 6.3 presents the implementation of the algorithm; section 6.4 presents experimental results and discussion.

Chapter 7 describes the collusions and the future scope of the work carried out in the thesis.