University of Alberta

Perceptual Quality Metric for Qualitative Evaluation of 3D Objects

by

Yixin Pan ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**

Department of Computing Science

Edmonton, Alberta
Fall, 2002

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-81459-9

Canada

University of Alberta

Library Release Form

Name of Author: Yixin Pan

Title of Thesis: Perceptual Quality Metric for Qualitative Evaluation of 3D Objects

Degree: Master of Science

Year this Degree Granted: 2002

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

(Signature)......Yixin Pan......

Permanent Address:
2#401, Staff Apartment
Quanzhou Sanitation School,
Quanzhou, Fujian Province, 362000
People's Republic of China

Date: ...Sep 25/2002

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis "Perceptual Quality Metric for Qualitative Evaluation of 3D Objects" submitted by Yixin Pan in partial fulfillment of the requirements for the degree of Master of Science

Dr. Anup Basu

Xiaobo Li
Dr. Xiaobo Li

Dr. Mrinal Mandal

Date: 9/23/02

Abstract

This thesis provides a review of well-known model simplification methods and correlated error metrics, and examines the factors that determine the perceptual quality of 3D images including geometry resolution, texture resolution, shading complexity, frame rate and some psycho-visual factors. Because bandwidth is usually the bottleneck in distributed graphics applications, we focus on the factors of geometry and texture resolution, and perform a user-study to evaluate the effects of geometry and texture resolution on the perceptual quality of a 3D object. Using the results from the user-study, a perceptual metric is introduced which gives a measure of how geometry and texture resolution control the overall quality in an image. Although this metric is not generic enough to evaluate arbitrary visual objects, we view our work as a first step towards providing a comprehensive analysis and modeling of perceptual quality based on physical and psycho-visual factors.

Acknowledgements

I would like to thank my supervisor, Dr. Anup Basu, for his valuable advice, guidance and support throughout this research, and his kind help in revision of this thesis. I am also grateful to the other members in the thesis committee, thanks to chairman Dr. H. James Hoover for his time during the oral examination, and Dr. Xiaobo Li, Dr. Mrinal Mandal for their suggestions and corrections.

I also want to thank all the members in Multimedia Communication group, especially Mr. Yinzhe Yu and Dr. Minghong Pi for discussion of this research, Irene Cheng for providing the experimental resources. Great thanks to my colleagues participated in the experiments, Mr. Juhua Shi, Mr. Xun Tan, Mr. Xiang Wan, Mr. Ling Zhao, Mr. Yue Zhang, Ms. Liu Wu, Ms. Tao Wang, Mr. Junfeng Wu, Ms. Yanxia Jia, Ms. Qiongyan Fang, Mr. Qiang Ye, Mr. Baochun Bai and Mr. Yuxi Li, without their kind help, it is impossible to achieve such interesting results. Financial support from the Department of Computing Science at the University of Alberta is gratefully acknowledged.

Finally, I wish to express my appreciation to my parents for their persistent support and encouragement, and my wife Li Lin for her endless love.

Contents

1. **Introduction** 1

1.1 An Example of Distributed 3D Graphics Application 1

1.2 Constraints of Distributed Graphics Systems 3

1.2.1 Computational Constraint 3

1.2.2 Network Bandwidth Constraint 4

1.3 Rendering under Limited Resources 5

1.4 Thesis Objective and Organization 7

2. **The Model Simplification Methods and Perceptual Quality** 9

2.1 Overview 9

2.2 The Model Simplification Methods 12

2.2.1 Curvature 11

2.2.2 Distance to Plane 13

2.2.3 Volume 15

2.2.4 Energy Function 15

2.2.5 Clustering 16

2.2.6 Comparison of Simplification Methods and Metrics 17

2.3 Appearance Preservation 18

2.4 Perceptual Experiments 20

3. **Examine Factors controlling 3D Images Degradation** 22

3.1 Factors Controlling Perceptual Quality 22

3.2 Incorporate Factors to Design Perceptual Metric 27

4. **Experiment Environment and Design of Perceptual Metric** 29

4.1 Experiment Environment 29

4.2 A Metric for Estimating Perceptual Quality 35

5. Experimental Results and Analysis **38**

5.1 Experimental Results 38

5.2 Contributions and Limitations 44

5.3 Use the Metric on Bandwidth-Constraint System 45

6. Conclusions and Future work **56**

6.1 Conclusions 56

6.2 Future Work 56

References **58**

List of Figures

1.1 Flat shading vs. Gouraud shading 3

1.2 Stanford Bunny 5

1.3 Stanford Bunny in different simplification levels 6

4.1 Orientation of Axis and Rotations of View Platform in Virtual World 29

4.2 The User Interface and the 3D Virtual World 30

4.3 Views of Nutcracker from Multiple Angle 31

4.4 The Zoomage 3D Scanner 32

4.5 Texture, Wireframe, and the Canonical View of Nutcracker 32

4.6 Evaluation Example 35

5.1 Quality vs. Texture Resolution (100% Geometry Resolution) 42

5.2 Mean Quality vs. Wireframe and Texture Resolution: Nutcracker 46

5.3 Best Fitting with Experimental *f(t)*: Nutcracker 47

5.4 Best Fitting with Estimated *f(t)*: Nutcracker 47

5.5 Mean Quality vs. Wireframe and Texture Resolution: Head 48

5.6 Best Fitting with Experimental *f(t)* : Head 49

5.7 Best Fitting with Estimated *f(t)* : Head 49

5.8 Mean Quality vs. Wireframe and Texture Resolution: Dog 50

5.9 Best Fitting with Experimental *f(t)* : Dog 51

5.10 Best Fitting with Estimated *f(t)* : Dog 51

5.11 Mean Quality vs. Wireframe and Texture Resolution: Doll 52

5.12 Best Fitting with Experimental *f(t)* : Doll 53

5.13 Best Fitting with Estimated *f(t)* : Doll 53

5.14 Mean Quality vs. Wireframe and Texture Resolution: Pot 54

5.15 Best Fitting with Experimental *f(t)* : Pot 55

5.16 Best Fitting with Estimated *f(t)* : Pot 55

List of Tables

5.1 Standard Deviation of Residuals with Experimental *f(t)*. 40

5.2 Quality vs. Texture Resolution (100% Geometry Resolution) 41

5.3 Standard Deviation of Residuals with Estimated *f(t)*. 43

5.4 Mean Quality of Users' Evaluations: Nutcracker 46

5.5 Standard Deviation of Users' Evaluations: Nutcracker 46

5.6 Mean Quality of Users' Evaluations: Head 48

5.7 Standard Deviation of Users' Evaluations: Head 48

5.8 Mean Quality of Users' Evaluations: Dog 50

5.9 Standard Deviation of Users' Evaluations: Dog 50

5.10 Mean Quality of Users' Evaluations: Doll 52

5.11 Standard Deviation of Users' Evaluations: Doll 52

5.12 Mean Quality of Users' Evaluations: Pot 54

5.13 Standard Deviation of Users' Evaluations: Pot 54

Chapter One

Introduction

Three-dimensional (3D) computer graphics is traditionally used in high-end graphics workstation for specialized applications such as computer-aided design (CAD) and feature movie production. In recent years, 3D graphics has taken center stage in interactive, networked applications such as computer games, e-commerce and educational software. On one hand, the rapid progress in graphics hardware development has made realistic visual effect possible in personal computers (PC). For example, the latest Graphics Processing Unit (GPU) for desktop PC is capable of handling more than one hundred million triangles per second [Nvid02]. On the other hand, the increase in data complexity still unfortunately surpasses improvements in hardware performance. Ten years ago, most 3D models were carefully designed by manual effort and consisted of relatively low number of polygons. Today, highly detailed models are generated in many applications. In computer vision, range data of an object is acquired via 3D scanning systems; in computer-aided design, polygonal model are produced by the subdivision of curved parametric surfaces; in medicine, organs and tissues are reconstructed from radiological and nuclear images; and in remote sensing, terrain data is obtained from satellite photographs. Each of these technologies can easily generate complex models consisting of millions of polygons. Since the representation and display of detailed 3D objects require substantial computer resources, trade-offs have to be made at the time of rendering to balance between perceptual quality and interactivity.

1.1 An Example of Distributed 3D Graphics Application

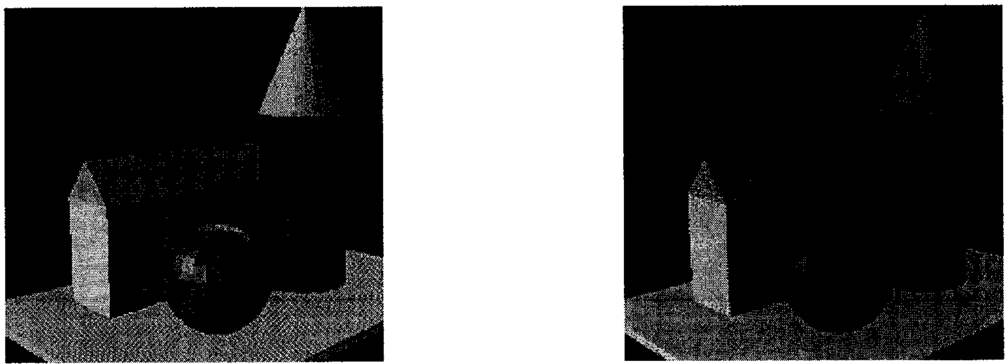
Usually a 3D application would be rendered as a virtual world, where a number of still or animated 3D objects are scattered inside the world. Users experience the 3D environment by moving in their desired directions inside this world, with their movement traced by tracker devices or simulators, and they are able to interact with 3D objects to perform tasks like steering a vehicle, touching and feeling objects with data gloves, or simply simulate the interaction by clicking and moving objects with mouse or keyboard. The underlying 3D rendering system records the position and the behavior of input devices, then reflects and projects the corresponding changes onto the users' platform. Traditionally, most 3D graphics systems are stand-alone applications that only allow one user at a time. Therefore, the 3D media data is usually stored on local hard disk or CD-ROM. Since Internet grows increasingly popular and high-speed connection becomes available for home usage, people have been trying to incorporate animation and interaction into Web content. The advent of new 3D authoring tools such as Java 3D [Sun02] and VRML [Care97] facilitates this trend. Nowadays more and more 3D graphics applications have been developed as distributed systems, in which media data is located on multiple hosts and many users can participate in the interaction at the same time, or even in the same virtual location. A typical example is 3D MUD (Multi-User Dungeon), which is a game hosted on one or several servers and played by as many as thousands of players simultaneously. MUD usually simulates a virtual society and involves a loosely organized context, e.g., adventure in an old castle with many rooms. MUD players adopt a character when they log in to a MUD. The characters are highly customizable, the players appear and disappear (login and logout) dynamically, and a significant body of information such as actions and messages is created unpredictably at players' will. Therefore, it would be unnecessary and overwhelming if all information were kept in local storage on every user end. In fact, substantial 3D model data and other game information are transmitted at runtime only when requested. For example, when a player moves to a new location and meets other players, the related model and profile information will be exchanged between the newly arrived and previously existing players in that virtual location. Ideally, all these changes in the virtual world should be immediately effective on players' display if the changes take place in their view angle. However, if the computer resource available is relatively limited compared to the complexity of data, the players would experience distinct latency, which may impair players' interactions.

1.2 Constraints of Distributed Graphics Systems

In the example of a distributed graphics systems above, there are mainly two constraints restricting the system from smooth rendering and interaction.

1.2.1 Computational Constraint

First, the rendering of high-quality 3D graphics is rather computationally intensive. One 3D object may consist of millions of polygons and usually many objects exist in a scene. Each polygon on a visual object will be warped with some texture, and then shaded according to shading model, lighting attributes such as color, direction and position, and texture attributes such as object material, vertex color and surface normal. A polygon may be mapped with several levels of textures and lit by multiple light sources in the scene, and for photo-realistic effects the inter-reflections and shadows should be taken into account, which will further increase the computation cost. Depending on the shading model, the complexity can be quite different. For example, in flat shading, all pixels for a polygon are assigned the shade value from one vertex of the polygon, and in Gouraud shading, each pixel is shaded with a value derived from tri-linear interpolation of the vertices in its enclosing triangle. Flat shading is faster than Gouraud shading but has more artifacts. Figure 1.1 shows a scene shaded by flat shading and Gouraud shading respectively.



Flat shading **Gouraud shading**

Figure 1.1 Flat shading vs. Gouraud shading

In order to achieve smooth interaction without flickering, the frame rate must be at least 25 frames / second, therefore, for highly detailed scenes it is still very difficult to obtain satisfactory interactive visualization even with the newest line of graphics cards.

1.2.2 Network Bandwidth Constraint

A 3D scene usually includes a number of 3D objects, and each object is associated with a large amount of geometric data, texture attributes, and one or more high quality texture images. When a user travels in the virtual world and new objects come into the user's sight, a smooth rendering would require a burst of data transmission in a short period. Although we may store all the objects information in a local system before rendering, and this approach works well for applications such as a frequently played game, for more general applications, since the behavior of a user is unpredictable, local caching often results in overly redundancy and does not improve much performance. Therefore, even if the user has a high-speed connection and a high performance graphics system, he/she may observe rendering delays from time to time. 3D compression technologies have been rapidly developed to help relieve this problem. The compression approaches use efficient means to describe polygon meshes that constitute the wire frame of 3D objects, then scale and quantize the vertex data, and further take advantage of entropy or other coding techniques to reduce the storage size. The compressed geometric data can be 10-20 times smaller than the original data without noticeable distortions [Deer95]. Figure 1.2 shows the image of Bunny, one of the most widely used models in computer graphics. It is a collection of 35947 vertices and 69451 triangles, and the size of uncompressed model is 22MB.

Note that while data compression techniques are very useful to save storage and bandwidth, they do not save computation power as well. The spatial data is kept in original resolution, so every polygon in the compressed 3D object is shaded and rendered as much as in the original data sets. Therefore, rendering cost remains constant, but additional computation cost is imposed on the rendering system as the overhead for decompression.

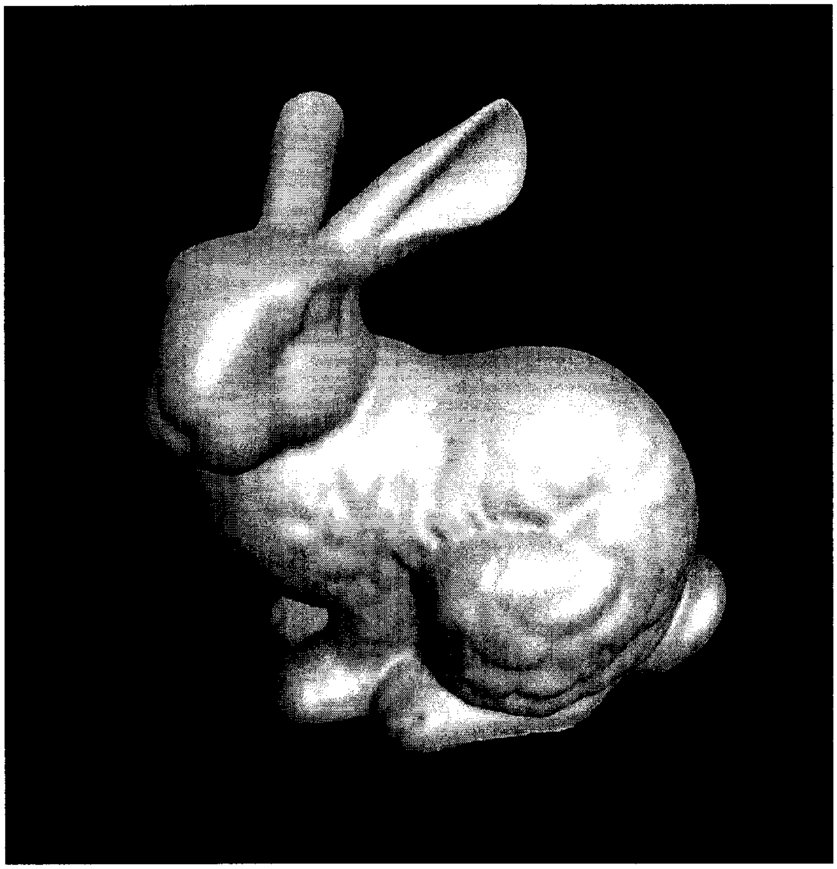
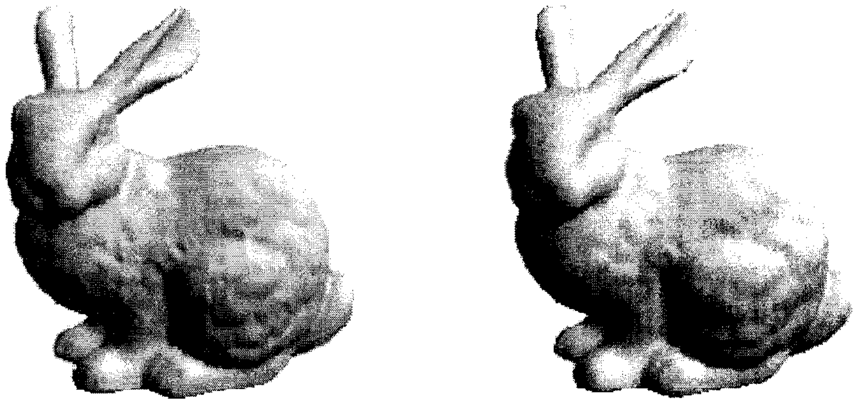


Figure 1.2 Stanford Bunny

1.3 Rendering under Limited Resources

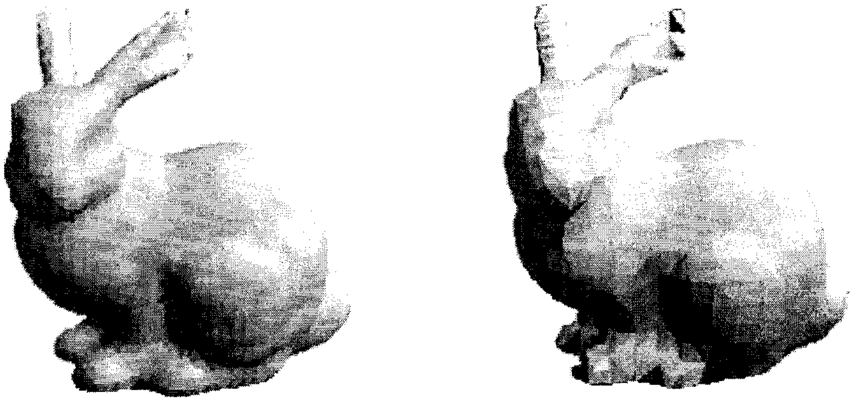
When graphics systems have the constraints discussed above, either with low-bandwidth, or under limited computational resources, rendering with fully size model cannot achieve interactive visualization. Fortunately, a highly complex data representation is not always required. Model complexity depends on the characteristics of the acquisition or fitting process, typical scanned and CAD-generated models are over-sampled. Simplification algorithms have been used to reduce the number of polygons and vertices. A compact representation of object geometric model can reduce the size of storage, the requirement of memory, as well as the computational complexity in rendering. Since high-resolution texture images usually range from several hundred KB to several hundred MB, it is also important to compress texture images, and in cases of distributed applications, decreasing resolution of textures should be considered. All these methods can speed up network transmission and decrease loading latency. Besides making the object visualization more efficient, a simplified model can diminish other computational cost involved in interaction, such as collision detection and visibility testing. In less interactive applications, e.g., realistic image synthesis, reduction of unnecessary object details that are imperceptible to the human visual system (HVS) can diminish substantial computer resource requirement and computation time. Other acceleration methods, although not as closely related to storage and bandwidth issue, can also help utilize the limited computation resources more efficiently. For example, a sophisticated graphics system may use multiple Level-of-Detail (LOD) models and adaptively update objects in the scene according to their velocities and positions. Objects close to users are rendered with higher LOD than those at a distance, and fast moving objects are updated at higher frame rate than those still or slow moving.

In short, one of the critical issues in designing effective interactive systems is to reduce the model representation and rendering complexity, while preserving as much as possible the model's visual appearance. Figure 1.3 shows the Bunny model at different simplification levels.

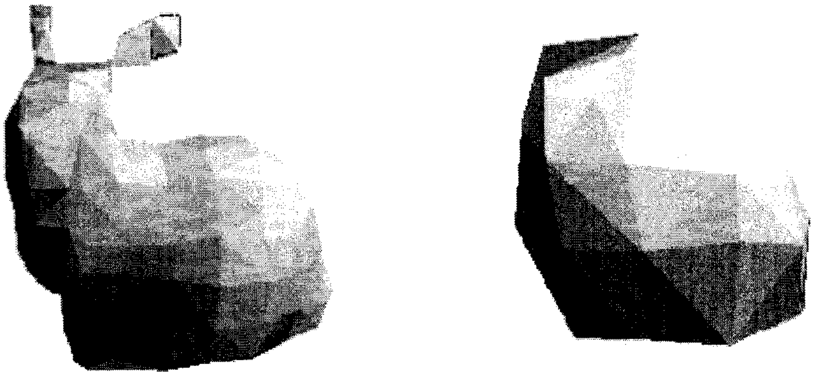


(a) 69,451 triangles

(b) 26,140 triangles



(c) 7,385 triangles (d) 1,919 triangles



(e) 462 triangles (f) 100 triangles

Figure 1.3 Stanford Bunny at different simplification levels.

1.4 Thesis Objective and Organization

A significant body of research has focused on the issue of model simplification. In most of these approaches, geometric metrics are defined to predict the fidelity of simplified model. On one hand, a simplification process demands an automatic measure or heuristic so that one of several alternative simplification operations can be selected in runtime. On the other hand, although some of these metrics do represent a bounded physical difference between the original and simplified objects, there is no guarantee how much the simplified model will resemble the original one. Ultimately, the perceptual quality is a subjective issue and can only be determined by human observers. Moreover, most research in the literature focus on stand-alone rendering systems, in which texture mapping is implemented in hardware for most systems and is relatively inexpensive, so the highest resolution of textures are always applied and the complexity of textures is usually ignored. In this article, we assume the framework of distributed applications, in which both computation and bandwidth constraints should be considered. The size of textures becomes an important factor under this assumption.

This thesis tries to address how individual factors, in particular *geometry resolution* and *texture resolution*, affect the overall perceptual quality and fidelity through a controlled study with human observers. The perceptual quality metric we derive from the experiments can be used in online 3D visualization to select an appropriate level of detail (LOD) based on bandwidth constraint.

The remainder of the thesis is organized as follows: Chapter 2 reviews the model simplification methods and the experiments that compare automatic measures to perceptual experimental results. Chapter 3 examines the factors that control the fidelity of 3D images and gives the intuition behind the experiments. Chapter 4 presents the user interface and environment used for subjective evaluation and proposes a perceptual metric for estimating the perceptual quality of 3D objects. The metric is compared with participants' evaluation results in Chapter 5. Finally, conclusion and future work are given in Chapter 6.

Chapter Two

Model Simplification Methods and Perceptual Quality

2.1 Overview

Perceptual quality is mostly studied in the fields of image compression and realistic image synthesis. Visual quality depends on human vision and viewing conditions, and accurate assessment relies on subjective viewing tests. Unfortunately, such tests are time-consuming and expensive. Therefore, people invent automatic metrics based on physical difference between the original and simplified models, and expect the metric to be close to the subject evaluation results.

In the field of image compression, Mean Square Error (MSE) has long been used as a quality predictor. MSE is defined as:

$$MSE = \sum_{0}^{M-1}\sum_{0}^{N-1}\frac{(P_0 - P_c)^2}{MN} \tag{2.1}$$

where P_0 is the original pixel value and P_c is the compressed pixel value, M and N are the width and height of the image respectively. However, recent research indicates that MSE is not well correlated to subjective perception quality [Giro93]. A number of new image quality metrics based on human visual system have been developed to tackle this problem [Daly93] [Lamb96a] [Lamb96b][Teo94][West95]. To predict perceptual quality, most of these methods incorporate spatio-temporal frequency response, color perception, contrast sensitivity and contrast masking results achieved from different psychophysical experiments. One of the perceptually based models is also extended to evaluate distortion of digital color video [Wink99].

In the field of realistic image synthesis, rendering systems can now closely approximate the physical distribution of lights in an environment. Since physical accuracy cannot guarantee that the rendering produces realistic visual effect, visual perception must be taken into account to achieve realism. That is, the computer image should not only be physically close but also perceptually equivalent to the represented scene. A number of perception-driven rendering algorithms are developed to incorporate models of the Human Visual System (HVS) directly into global illumination computations so as to improve the efficiency [Ferw97] [Boli98] [Rama99].

In the research on model simplification, most methods use a physical error metric to predict the difference between the original and simplified models. While such metric generally monotonously control the visual appearance, it does not answer whether the metrics well represent the perceptual quality. Most researchers understandably avoid this question and list a number of images to depict their result of simplification. Until very recently, a few authors developed several perceptual experiments trying to examine the validity of error metrics introduced in model simplification methods. However, their results are not encouraging.

The goals of perceptual quality research in these three areas are rather different. Those in image compression are simplest in the sense that they only need to deal with 2D still images, and the images are ready beforehand. In contrast to image compression, realistic image synthesis methods compute images in runtime based on 3D models and global illumination. These methods focus on incorporating perceptual quality in the synthesis process to reduce computational complexity, e.g., eliminating unnecessary indirect illumination. Unlike image compression, in which an overall fidelity metric is desired, image synthesis approaches predict the locations of visual distortion, and adaptively give more sampling to these areas until such distortion is invisible. Since realistic image synthesis is usually discussed in the manner of offline rendering, although the rendering is 3D-based, the evaluation of quality is based on view-dependent 2D images. Perceptual quality discussed in the context of model simplification is comparatively generalized and difficult. The quality of 3D images taken into account includes animation and interaction, and is thus view-independent. Researchers are seeking a view-independent metric to guide the simplification process. One of several alternative simplification operations is chosen in each step and ideally is able to produce the best overall quality possible. In a larger context, such simplification operation is one of the options among decreasing geometry resolution, decreasing texture resolution, or using a simpler shading model, etc. In a smaller context, it is the decision of folding which node, or reducing which part of texture sampling on single object.

In the following sections, we review the simplification methods, including those only considering geometric model and those taking account of appearance. The related metrics are also discussed, as well as those perceptual experiments testing these metrics.

2.2 Geometric Model Simplification Methods

Cignoni *et al.* described a number of characteristics to classify existing simplification approaches [Cign97], namely, I/O data domain, simplification goal, heuristics for the evaluation of meshes optimization, and incrementality of simplification. The classification in this thesis is also based on heuristics for the evaluation of meshes approximation. However, we consider a different aspect: how the error metric is constructed? In this way, the model simplification strategies can be roughly classified into five groups: Curvature, Distance to Plane, Volume, Energy Function and Clustering. The curvature of an area depends on the normals of the constituent planes. If the normals of adjacent planes are nearly parallel, then the area is relatively smooth, and these planes may be merged to have a more concise representation, which is called patch decimation. Distance to Plane is the most intuitive metric defined in model simplification, the metric is generally defined as a distance from the input model to the output model, for example, the largest distance from any input vertex to the output model surface (Hausdorff Norm). The smaller this metric, the closer are the input and output models. The Volume metrics stress preserving both the topology and the volume of the model, and it is only valid for manifolds. Energy function is an *ad hoc* method to achieve meshes optimization, and the parameters in the method can ensure that a minimum topology is preserved. Clustering examines the spatial proximity of vertices, and merges close vertices to get a concise representation.

2.2.1 Curvature

Patch Decimation

Kalvin *et al.* used a simple patch decimation method in their efforts to create surface models from medical data [Kalv91], after an initial polygonal surface is created to approximate the input data, adjacent coplanar polygons are merged to simplify the model. Since only precisely coplanar faces are merged, the degree of simplification is highly dependent on the curvature of the object, and thus only limited simplification is obtained. Hinker *et al.* extended the patch decimation method to merge the nearly coplanar polygons [Hink93]. If the angle between the normal vectors of two adjacent triangles is below a given bound error, the two triangles are merged. Finally, the merged polygons are re-triangulated with a simple and robust method. However, this method is highly ineffective for surfaces with high curvature.

Curvature Equalization

A curvature equalization method is described by Scarlatos *et al.* [Scar92]. They tried to balance the curvature of the input data within each triangle by adjusting the triangulation of the original surface. There are three steps in their strategy: First, triangles with high curvatures are collapsed into single vertices. Second, adjacent triangles are merged if they share the same plane. A re-triangulation is performed after each of the first two steps. Finally, edge swapping is attempted in order to achieve a better shape and fit.

Re-tiling

Turk developed a simplification algorithm that takes a polygon surface as input and re-tiles the surface with a user-specified number of vertices [Turk92]. Unlike most other approximation methods, the output vertices are not restricted to be on the vertices or edges of the original meshes, instead, they can be anywhere on the surface. First, a user-specified number of points are randomly distributed on polygons, then an iterative relaxation procedure is used to repel the new vertices from each other, a vertex that is pushed off one triangle is moved onto an adjacent one. This procedure will make the new points uniformly spread on the surface. These new vertices, together with the original vertices, are used to triangulate the original polygon surface, so-called *mutual tessellation*. The original vertices are then deleted one by one from the mutual tessellation, and the resulting holes are re-triangulated to keep the topology the same as the original surface but comprising of only the new vertices. Turk further improved his method by taking into account the curvature of vertices. Giving points with higher curvature less impulsion, very curved areas will contain many more points than those smooth areas in the relaxation procedure, which results in a better re-tiled surface.

2.2.2 Distance to Plane

Superfaces

In their later work, Kalvin *et al.* developed a *superfaces* method [Kalv94][Kalv96], which is quite similar to Hinker's approach [Hink93], the surface is first segmented into approximately planar patches, then the patches are subdivided into star polygons and re-triangulated. The tolerance error of patches they define is the distance to the original plane. Compared to Hinker's approach in which angles between normals of adjacent faces are used, the former is less sensitive to noise in the sense that a noise vertex seldom departs far away from the actual surface, but often consists a rather distorted oriented plane.

Vertex Decimation

The original vertex decimation algorithm proposed by Schroeder *et al.* applies multiple passes over all the vertices in the meshes. In each pass, for any vertex that is not on a boundary and have an error below the given threshold, the vertex and all triangles using the vertex are deleted. The resulted holes in the meshes are then re-triangulated. The error is measured as the distance from candidate vertex to the average plane of the surrounding polygon. The decimation threshold can be adjusted in each iteration and the algorithm terminates when none of the vertices meets the decimation criteria or the desirable percentage of reduction from original meshes has been reached. This method is fast since the computation of error measure is inexpensive and multiple vertices are removed in each pass.

Extension of Vertex Decimation

Soucy *et al.* used more accurate error bounds in their decimation heuristics

[Souc92]. Their approach differs from Schroeder *et al.*'s mainly in two respects. First, no threshold is used, in each pass, the errors of all vertices are compared and only the one with least error is deleted. Second, not only the vertices in current meshes, but also all points from input are considered for distance-to-plane error calculation. All removed vertices (ancestors) are associated with re-tessellated triangles. During the decimation process, each vertex deletion is attempted with re-triangulation, and the error is defined as the maximum distance between the vertex or neighboring ancestors and the re-triangulated surface. After the least error vertex is selected and the surrounding polygon re-triangulated, the ancestors are redistributed among the new triangles. This method can yield higher quality than the method of Schroeder *et al.*, but it is much slower and uses more memory in order to keep the deleted points.

Some recent vertex removal algorithms have focused on more sophisticated error control. Ciampalini *et al.* presented a new meshes decimation method called JADE (Just Another DEcimation) [Ciam97]. This algorithm is similar to Schroeder's decimation algorithm, but instead of using a local error measure, the decimation is based on a global error. Bajaj *et al.* keep track of "positive" and "negative" error bounds at each vertex throughout the iterative removal process [Baja96]. These error bounds are computed from the planar projections of old and new triangles in a region. Klein *et al.* tried to achieve higher reduction rates measuring surface distances in the more natural Hausdorff Norm [Klei96]. Cohen *et al.* developed the simplification envelope algorithm that has a strictly bounded decimation error [Cohe96]. Two offset surfaces, one inward, one outward from the original surface with a given tolerance distance form the two bounded envelopes. In each pass of the method, the vertices decimation and re-triangulation is temporarily performed on every vertex in the meshes. A candidate triangle is then attempted to fill the hole. The candidate triangle is checked for validity by testing that it neither intersects the envelope nor overlaps the previously inserted triangles, if no valid triangle exists, the whole operation is cancelled and the vertex will not be deleted. In the processing of hole filling, priority is given to the triangle that *covers* the greatest number of previously uncovered vertices. Most of these sophisticated error control algorithms seem to produce better quality meshes than the original one, in return for more computational expense.

Edge Collapse by Distance to Plane

Ronfard *et al.* created an edge collapse method. They defined the notion of *zone*, which is a set of planes associated with each vertex. The zone for each vertex is initialized to the adjoining planes of the vertex. When an edge collapses, the zone for the new vertex is the union of zones of the two end vertices forming the edge, the increase of maximum distance from the vertex to its zone is considered as the contraction cost. In each iteration, the edge of lowest cost is selected and collapsed.

2.2.3 Volume

Edge Collapse by Volume

Gueziec proposed a method using edge collapse for simplifying manifolds [Guiz95]. An error radius is associated with each vertex. During simplification, an error volume is defined based on all the radii across the surface. The error volume always encloses the original meshes. When an edge collapses to a vertex, the error radius for the new vertex is the minimum radius that makes the new error volume enclose the old error volume. Edge collapse has been given a lot of attention in recent years because re-triangulation is not required in this method and it is relatively efficient.

2.2.4 Energy Function

Hoppe et al. [Hopp93a][Hopp93b] proposed a solution to optimize triangle meshes under a global energy function. Defining meshes $M=(K, V)$, meshes are optimized by minimizing the energy function:

$$E(K,V) = E_{dist}(K,V) + E_{rep}(K) + E_{spring}(K,V) \qquad (2.2)$$

Here, $E_{dist}(K,V)$ represents the geometric distance of simplified model from original one, $E_{rep}(K)$ is a penalty for large number of vertices, and $E_{spring}(K,V)$ penalizes long edges in the triangulation. The optimization randomly selected one edge from the meshes, and either collapse it, split it, or swap it. Hoppe *et al.* furthered their approach with progressive meshes [Hopp96] that offers an elegant solution for a continuous resolution representation of triangular meshes. A triangular meshes with n vertices M^n is simplified into successively coarser meshes M^{n-i} by applying a sequence of edge collapses. The interpolation of two adjacent meshes can provide a smooth morphing in the rendering. Hoppe also suggested that during optimization, edge collapse alone is as good as, or even better than the more general operations taken in original algorithm. Popovic *et al.* developed an encoding strategy for progressive meshes to enhance the storage and transmission efficiency [Popo97].

2.2.5 Clustering

Rossogmac and Borrel developed a simplification algorithm based on vertices clustering [Ross93]. One feature of this approach is its generality. While most of the methods described above can only handle manifold surfaces, this method is able to process non-manifold surface as well. In this algorithm, the bounding box of the original surface is uniformly subdivided into a regular grid of cells, those vertices found to be in the same cell are considered to be close to one another, and a new representative vertex is created to replace them. In order to determine which vertex in the cell is most important (representative), all vertices are graded (weighted) according to a metric consisting of two factors: (1) probability of lying on the object's silhouettes from an arbitrary viewing direction, and (2) size of the faces bounded by the vertex. After the vertices in a cell are merged, the model is simplified by removing duplicated triangles, edges and points. This method is fast and it can achieve arbitrary simplification level, but on the other hand, it does not preserve details well and shows the artifact of clustering grid [Heck94].

Based on Rossogmac and Borrel's approach, Low and Tan proposed several enhancements to provide higher quality for simplified model [Low97]. They modified the grading scheme to more accurately compute the weight of vertices, and replaced the uniform-subdivision cells with "floating cells". These floating cells are determined by multiple iterations. All the vertices are sorted in non-increasing order on their weights. In each iteration, one floating cell, typically a cube or a sphere centered on the vertex with highest weight, is selected and all vertices in this cell are removed from the vertices list. Since the new method can better preserve important features, the quality of approximate model is improved. Reddy also provided an extension of vertex clustering algorithm to preserve shape details [Redd96], instead of simply merging all vertices in a cell, all the edges in a cell are tested according to a curvature and size criterion, and only those located in a relatively smooth area or in a small triangle are contracted. In brief, proximity of vertices is considered in the first phase of simplification, and in the second phase, curvatures of surface inside cells take over to decide the actual contraction.

One significant algorithm in this category is the pair contraction algorithm based on quadric error metrics described by Garland and Heckbert [Garl97]. Pairs of vertices with smallest quadric error metrics are successively merged to produce approximate models. A candidate pair is either two endpoints sharing the same edge or two unconnected vertices following some proximity criterion. The quadric error is equivalent to a distance-to-plane metric but can be computed quickly as a product of two vectors and a quadric matrix. This method is capable of producing high quality approximations close to those distance-to-plane metric approaches in much shorter time accredited to the efficient error evaluation.

2.2.6 Comparison of Simplification Methods and Metrics

In comparison to the methods above, curvature metric is mostly used in patch decimation methods, these approaches are fast but highly simplified model may not be achieved. Distant to Plane and Volume metrics can ensure a rigid physical bound for simplified model, however, the computation is relatively inefficient. In the methods employing these metrics, edge contraction is most flexible and produces the best quality. Energy function, as an *ad hoc* method, produces similar results to other edge contraction schemes. Vertices Clustering is fastest and able to handle non-manifold, although the quality is worse than those resulted from other methods. The quadric error metrics in this category is promising to generate acceptable simplified model in real time. With a large number of methods varied in efficiency, quality and generality, the proper selection of error metric and simplification method in practice is dependent on the purpose of specific application.

Note that in the discussions above, only geometric model is considered, and metrics are usually specific to the simplification approach. In order to evaluate arbitrary simplified meshes accuracy, Cignoni *et al.* developed the *Metro* tool, which is able to calculate a distance-to-plane error metric without prior knowledge of the simplification method [Cigo97].

2.3 Appearance Preservation

Most polygon simplification methods consider only surface geometric positions, which are represented by the coordinates of the polygon vertices. However, three other surface attributes also contribute to the visual appearance of a polygon model, in this thesis we use the term *appearance attributes* to distinct them from surface geometry:

1. Surface curvature, represented by surface normal vectors across the polygons, is used in shading to calculate the light reflection direction. Both the diffuse and specular reflections require proper normal specification. In triangular meshes, there is only one normal associated with each triangle because triangle is always planar. For application requiring rendering of both outside and inside face, the inside face normals are specified as the inverse of the outside face normals.

2. Surface color, represented as 3-tuple RGB or 4-tuple RGBA scalar values across the polygons, is usually being mapped to one or several 2D rectangular color images. RGB is the intensity of primitive colors Red, Green and Blue, and each color is usually stored as an 8-bit integer representing 256 levels respectively. 4-tuple RGBA value also defines the transparency intensity, blending foreground and background with this weight produces transparent visual appearance. The image maps are called texture images, and every vertex on surface is associated with a 2D texture coordinates. The color value of pixel inside polygon can be looked up in texture maps by performing interpolation over polygon vertices.

3. Surface material property, defined as a set of scalar values, specifies the smoothness of the surface and the spread range of viewing angle for which different types of reflections can be seen. For example, specular reflections

occur naturally with smooth surface, when a surface is sufficiently smooth, it acts like a mirror to reflecting the light and keep the light in nearly the same color and intensity. Consequently, the specular color of an object is normally white. The material property is dependent on the shading model and the

underlying rendering system.

Appearance attributes are taken into account during the simplification process in a few recently published methods. Hoppe made one of the earliest attempts to preserve appearance attributes as well as geometry in the framework of progressive meshes [Hopp96]. A separate energy term is introduced to penalize energy function in order to limit the deviation of colors, other terms to preserve material boundaries and sharp features (e.g. creases and shadow boundaries) are proposed as well. Garland and Heckbert generalized their original quadric error metrics to incorporate color and texture attributes [Garl98]. This is accomplished by assigning a separate quadric to each attribute. A high dimensional vector is used to represent each vertex, e.g., $[x\ y\ z\ r\ g\ b]^T$ in colored model, this vector is then computed with a high dimensional quadric matrix to estimate the error introduced in each pair contraction. Hoppe recently provided a more concise quadric error metric for preserving appearance attributes. The error metric is measured as an integrated deviation of contracted vertex to its projection in the original surface, where the integrated deviation is the sum of squared distance and squared appearance attributes deviation between these two positions [Hopp99]. Erikson and Manocha used point clouds to account for the normals error, colors error and textures coordinates resulted from edge contraction. Their method is efficient enough to achieve 3.5 to 5 times average speedup with little lose in image quality. Cohen *et al.* introduced the texture deviation metric that incorporates color and normal errors in simplifying textured meshes [Cohe98]. The metric restricts the map shift occurred in simplification to be no more than the user-specified number of pixels on the screen. This metric is defined as the deviation with the same parametric location in the texture domain between a point on simplified meshes and the projection of this point on original meshes. In simplification process, a simple geometry error metric is computed to determine the new vertex location of single edge collapse operation, and then the costs of edge collapses are sorted in increasing order of texture deviation metric. Cohen *et al.* suggested texture deviation bound is sufficient to guarantee a bound on both the surface and the texture deviation. Following the same framework and two phases simplification process, Lindstrom and Turk used the simple heuristic introduced in [Lind98] to determine where to place the new vertex, and they developed the first image-based simplification error metric to order edge collapse operations [Lind00]. The image-based error metric calculates the root mean squared difference of a collection of rendering images between original model and simplified model. In order to capture approximately the entire appearance of an object, the collection of rendering images are carefully designed to capture from a number of different camera positions around the object. One virtue of this approach is that not only coordinates of the texture parameterization is considered, but also the color variations within the texture is concerned. Therefore, if the texture image degrades due to either higher compression rate or lower resolution, none but this image-based approach will be sensitive to the degradation.

2.4 Perceptual Experiments

Although all automatic measures above are related to perceptual quality, very few studies have performed psycho-visual experiments to evaluate object quality probably for two reasons. First, interactive visualization has a short history for less than ten years since the time high-performance graphics accelerators became available. Moreover, perceptual experiments are time-consuming and expensive processes in which fewer researchers would like to participate. Watson *et al.* used naming time to study the correlation of between automatic measures and perceptual quality [Wats00]. Naming time is the period to recognize a given object, which has long been used in cognitive psychology research as a measure of recognition. The participants were given 30 models, each at three different simplification levels (50%, 80%, 100%). The naming times for each participant were recorded and analyzed. Their results show that naming time is poorly correlated to either geometric measure or image-based measure. However, it remains doubtful whether naming time is a good measure of visual quality. For example, each model is repeated three times in the test, the participant is likely to have had prior knowledge of the model when he/she views it at the third time, and probably name it quickly even if it is a more concise model. Watson *et al.* included ratings and forced choices as perceptual quality measures in their later work [Wats01], they suggested ratings and forced choices are better measures than naming times for non-significant number of participants. In addition, BM, MSE and MetroMn are excellent predictors of fidelity as measured by ratings. Nevertheless, Watson *et al.* only used still images in their experiments, thus their results are view-dependent. Are view-dependent results equivalent to more generalized view-independent results? Rogowitz *et al.* provided a negative answer in their experiments [Rogo01]. Their models are illuminated from two different directions, and for the same 3D object, one stimulus is the still 2D image of front view, while the other is the view of the object rotating from front to side. The results indicate that visual quality varies significantly with the direction of illumination, and the perception quality of 2D images are not correlated to 3D images at all.

Although more experiments are needed to draw robust conclusion, we can observe from above that neither physical-based metric Metro nor image-based metric MSE is a good predictor of view-independent perceptual quality for 3D object. Nevertheless, some useful results are given in these researches, for instance, rating is a better measure than naming time for non-significant number of participants, and illumination and animation of objects are important factors in perceptual quality. We will consider these factors in our experiments.

Chapter Three

Examine Factors controlling 3D Image Degradation

Researchers have made many efforts in model simplification, and their methods vary in quality, efficiency and generality. Error metrics are provided with most algorithms as a measure of geometric difference, some error metrics also incorporate the parameterization of appearance attributes. However, in the relatively few available psycho-visual experiments that compare these automatic measures and human visualization evaluation, none of these metrics adequately model perceptual quality of 3D objects. Another drawback of these approaches is the negligence of factors other than geometric difference; for example, the degradation of texture image, although very common in distributed applications, is seldom considered. In this chapter, we first discuss factors that affect the perceptual quality of 3D images, and based on bandwidth constraint, we provide the intuition to construct a perceptual metric from the user study of psycho-visual experiments.

3.1 Factors Controlling Perceptual Quality

Several flexible strategies can provide smooth degradation in the fidelity of 3D images. These strategies include the reduction of geometric complexity, texture resolution, shading complexity, and decreasing the rate at which the 3D image is updated, etc. Each strategy is a dimension in the distortion function, where the distance between the values applied in real rendering and in the ideal image is a metric of error in the corresponding dimension. Each error contributes to the overall distortion and reduces system requirements as well.

(a) Object Geometric Model

The shape of a 3D object is usually specified as polygonal meshes, where the vertices of the meshes represent the depth samplings on the object surface. Dense polygonal meshes can be used to describe accurate and detailed object models. Given an adequate number of polygons, we are able to describe models as complicate and detailed as a carpet or a sweater with high precision. Nevertheless, complex models demand large space for storage and fast computing speed in rendering, which is not suitable for interactive visualization. In fact, performance of graphics systems is often measured in terms of the number of polygons (usually triangles) per second that the system is capable of processing.

To simplify the rendering complexity, a common approach is to store multiple models for the same object, and adaptively select a suitable model according to the distance between the object and the viewer, the so-called *level of detail* (LOD) techniques [Lind96]. If the object is close to a user, a high-resolution model is displayed, otherwise, a simpler model is chosen. The approach can lead to a smooth rendering, and computation resource can be saved when objects are at a distance. Multiple-resolution models can be constructed from the simplification methods reviewed in Chapter 2. In the context of bandwidth-constrained systems, if multiple-resolution models are saved as several distinct simplified models, the transmission of these models will impair the efficient usage of limited bandwidth. As an elegant solution, Hoppe addressed the framework of incremental simplification [Hopp96]. With proper encoding, any level of simplification can be reconstructed based on the same data set. The visualization is available once the coarsest model is transmitted and constructed, and incrementally refined as additional information arrives. Such a process is analogous to the decoding of GIF [Comp90] format image.

3D objects lose details on their shape when the geometry is simplified. For detailed models, artifacts are invisible below some error threshold. Many researches in realistic image synthesis explore the human visual system and try to evaluate whether difference is detectable between two images. Such process can be incorporated into offline shape simplification methods to construct perceptually undistorted simplified geometry.

(b) Texture Resolution and distortion

Construction of high quality 3D objects requires high-resolution texture images for mapping. Texture mapping is the process of wrapping one or several 2D images around a 3D object so that the 3D object acquires a surface texture similar to that of the 2D images. This technique can add realism in areas that are too complex for a geometric model to describe, e.g., cloud, carpet, tree etc. Exceptions exist in particular situations. In the experiment of substitution geometry with texture, Rushmeier *et al.* observed that covering colored geometric models with very low-resolution texture decrease the perceived quality [Rush00]. This is because the spatial frequency of an overly simplified texture is significantly lower than the original one, and can be easily discriminated. From the observation, we are advised not to use very low-resolution texture, but in most cases, texture maps improve the overall quality.

As a fundamental and very powerful technique, texture mapping has been so frequently used that it is now implemented both in software interfaces and in computer graphics hardware [Haeb93]. Texture mapping is still expensive for software-based systems as it is a pixel-by-pixel operation. On hardware-based systems, textures are usually loaded from main memory into special fast memory on graphics accelerators before rendering. Since such on-board memory is limited, and texture-swapping speed is restricted by bus bandwidth, for high-resolution textures, it is often impossible to allocate on-board memory quickly enough to render the scene at adequate frame rate. Dumont *et al.* developed a resource allocation scheme that dynamically choose the resolution to use for each texture in board memory, the resolution is based on a perceptual metric that consists of current view-point, illumination conditions, texture contrast and frequency content [Dumo01].

The residence of large texture images in main memory also increases memory requirement. In distributed applications, the size of texture maps becomes more important since the transmission of these images can be slow and become the bottleneck of the system. Image compression techniques have been extensively used in the last twenty years, however, even with advanced compression techniques such as wavelets or fractals, image files remain relatively large. Reduction of texture resolution is thus a practical solution to addressing system constraints. Simple texture reduction relies on interpolation to get a smaller image, approaches that are more sophisticated adaptively sample the original image based on a viewer's attention, giving the centre of attention higher resolution than the periphery [Basu98], such approaches are based on the observation that fovea has the sharpest acuity in the retina.

Objects lose detail and become granular and fuzzy when texture resolution is diminished. In a LOD scheme, usually multiple levels of textures (MIPmaps) are pre-computed and stored, in runtime, the actual using level is dependent on the distance between the viewer and the object. Storage is traded for performance in this strategy.

For distributed applications, besides decreasing texture resolution, increasing image compression ratio can also reduce data transmission required. Using fewer bits in quantization, for example, can describe the texture image in a more compact format, in return for more color distortion and artifacts in the decompressed image.

(c) Shading

The shading complexity is determined by the type of shading model, the number and the positions of lights illuminating the scene, the textures on objects, and the using of other shading effects such as shadows and inter-reflections. Constant, Faceted, Gouraud, and Phong shading are commonly used scan-line graphics techniques that give increasingly better images at increasingly higher computational expense. Ray tracing and radiosity are two alternative techniques to produce photo-realistic images, both require vast amounts of processing power and usually only being considered in offline rendering. Shadows and inter-reflections are also important in realistic shading. The former one is sometimes simulated by binding an extra object in many graphics systems, while the other is rather complex and usually ignored in interactive visualization. All these techniques provide trade-offs between computational cost and the realism of a 3D image. Reducing shading complexity causes artifacts in the subtleties of lighting and reflection. Rogowitz *et al.* reported that visual quality varies significantly with the directions of illumination [Rogo01], thus, comparison of different shading models should be based on the same illumination environment.

(d) Decrease Frame Rate

Many systems automatically decrease the frame rate to allow enough computation time to render a scene. Since the refreshing rate is inversely proportional to computation time of a single frame, this approach is simple and able to smoothly control the quality of an individual frame. However, it also leads to the problem of flickering that is rather annoying for users. In some interactive applications, even slight inadequacy of refreshing rate may be very costly and unacceptable. For instance, in a shooting game, users may not be able to aim at an object at a particular position or moment. More sophisticated systems would maintain the frame rate, and update the objects within the scene at different rates in order to optimize the performance. For example, in *Talisman* [Torb96], rendered sprites (segments of rendered image) are reused in subsequent new frames until the estimation of the error exceeds the tolerated threshold. Several frameworks are addressed to maximize image quality while maintaining fixed frame rate, in other words, maximizing the overall cost/benefit ratio. Funkhouser and Sequin used the number of polygons, pixels and vertices in the objects to estimate the rendering costs and benefits, eccentricity and velocity of objects are also accounted as *ad hoc* weighting factors [Funk93]. Mason and Blake enhanced that work by clustering objects and organizing them in a hierarchy [Maso94].

(e) Distance

The perceptibility of visual stimuli depends on the distance between the stimuli and the viewer. The contrast sensitivity function (CSF) [Mann74] models how sensitive human beings are to the various frequencies of visual stimuli. Since variation of the distance changes the observed spatial frequency of stimuli, the perceptibility of degradation is dependent on distance. CSF provides threshold information, which is commonly used in image synthesis to predict if a simplification operation is perceptible at a given distance. Luebke and Hallen reverted this function and computed a threshold distance for each simplification operation, then sorted the order of operations according to this distance [Lueb01].

Distance is frequently employed to determine a LOD. Efficient systems choose different simplification representation for objects at different distances. While full polygon models are best for nearby objects, meshed imposters or image with depth are good choices to replace moderate distance objects. For objects far away, a flat imposter as the background is adequate.

(f) Visual Masking and Visual Adaptation

Visual texture may hide faceting due to tessellation of curved surface. Ferwerda *et al.* developed a comprehensive model of visual masking that can predict whether the presence of one visual pattern affects the perceptibility of another visual pattern when one is imposed over another [Ferw97]. Although not a perceptual metric, it suggests that more aggressive imperceptible geometry simplification may be possible for textured models.

Visual adaptation is the phenomenon that the sensitivity of human eyes varies in different luminance, and it takes time to adapt from one luminance to another distinctly different luminance. For example, we sometimes need several minutes to see well when entering a dark theatre. Ferwerda *et al.* proposed a computational model that predicts the visibility of object features and colors at particular illumination levels. The model simulates human perception in the course of light and dark adaptation based on psychophysical experiments [Ferw96].

(g) Other Factors

Some other important factors are discussed in psychology, such as the attention of users. Fovea, the region of highest sensitivity on the retina occupies the central one degree or so of vision. Visual acuity, measured as the highest perceptible spatial frequency, is significantly lower in the visual periphery than at the fovea. Thus, if within an image different objects have different resolution, the perception of visual quality largely depends on the object resolution in focus. Since these factors are subjective and difficult to be controlled in a graphics system, we try to eliminate such influences in our experimental design.

3.2 Incorporating Various Factors to Design Perceptual Metric

From the discussion above, as decreasing frame rate is costly for interactive applications and may be uncomfortable for viewers in experiments, we would not adjust frame rates in our experiments. Among the other factors, geometric model and texture resolution are our focus since they are decisive in both computation-constrained and bandwidth-constrained graphics systems. Other factors such as shading and distance are also important issues in determining perceptual quality. However, as we are in the initial stages of our research, we would not incorporate these factors in our experiments. Instead, we try to exclude the effects of some factors. That is, we perform all our experiments independent of certain factors.

All objects are examined at the same distance and under the same illumination environment, and the evaluation process ensures that the participants pay attention to the same object in a scene. Participants are allowed adequate time to evaluate one object so that visual adaptation effect can be ignored. The influence of visual masking is unpredictable in our experiments. Fortunately, visual masking is only effective to mask artifacts for slightly degraded geometry model, and negligible for further simplified geometry. Since most stimuli in the experiments belong to the latter category, most samples collected in the experiments will behave without the presence of this factor, the values of other samples may change a little if visual masking is taken into account, but the overall fitting curve should not be influenced.

Most error metrics are based on physical difference between the original and simplified models. Unfortunately, perceptual experiments have demonstrated that these physical metrics have limited capability of estimating visual quality. Moreover, the degradation of texture images is rarely considered in the literature despite of its importance. Based on these two observations, we design perceptual experiments to study the sensitivity of viewers to the degradation of geometric model and texture resolution. Our goal is to extract from users' evaluation a statistical model that predicts how these two factors control the quality of 3D images.

Chapter Four

Experimental Environment and Design of a Perceptual Metric

4.1 Experiment Environment

User Interface

A user interface for interactive viewing of 3D objects, along with a client-server implementation on the Internet has been developed. We choose Java 3D as our rendering platform. The implemented client is an applet capable of either being embedded in Web pages or running as a normal application. The interface offers users Six Degrees Of Freedom (SDOF), i.e., the users can move virtually in any direction in the 3D virtual world with the interaction of mouse and buttons. Besides walking in six directions, rotations of view platform parallel and perpendicular to X-Z plane are also provided, which simulate rotating left and right, and looking upward and downward respectively. Figure 4.1 shows the orientation with respect to the viewer in a virtual world and the rotation planes of the view platform.

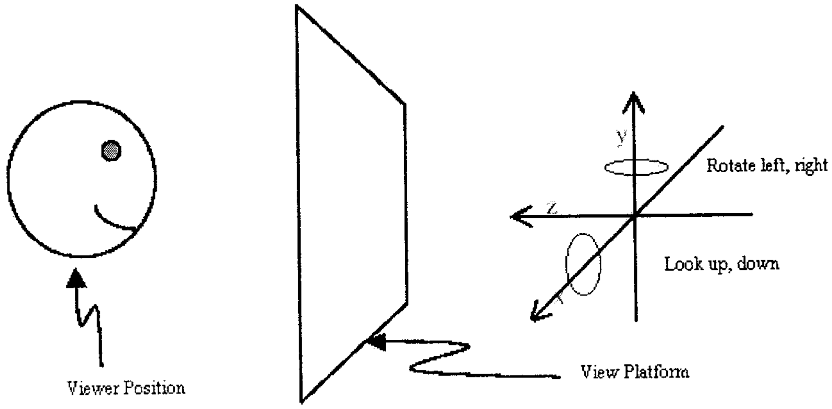
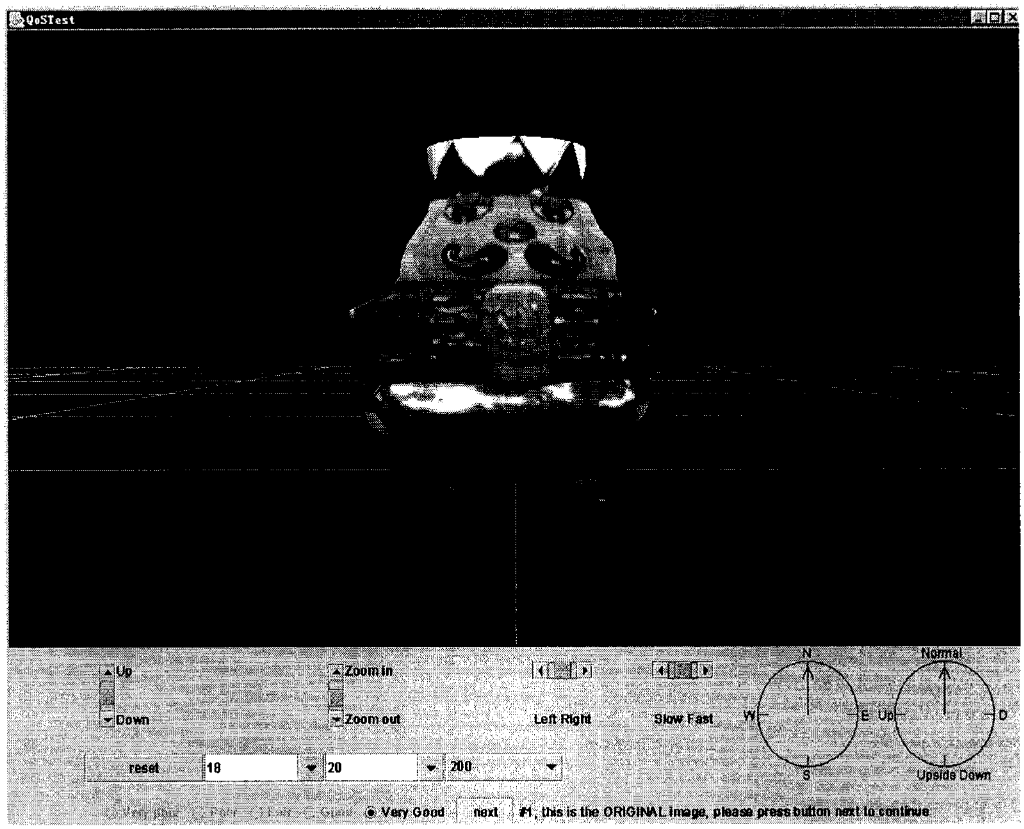


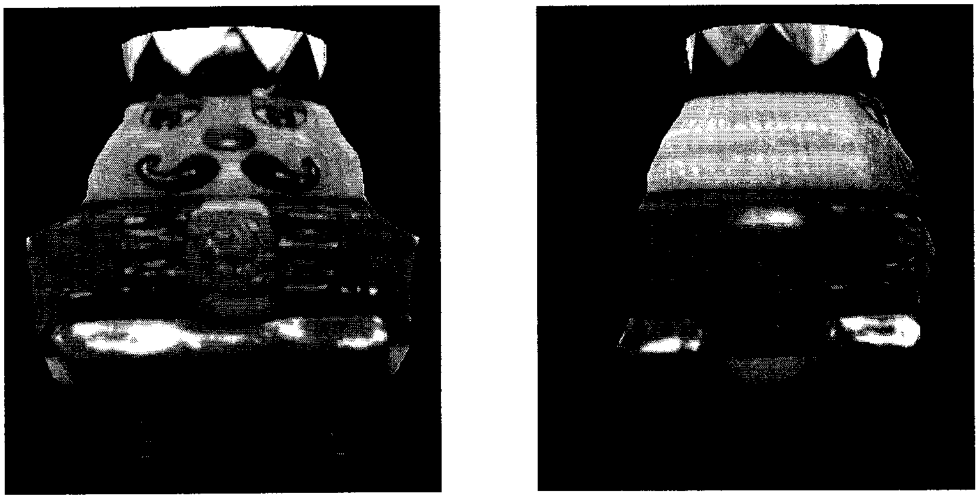
Figure 4.1 Orientation of Axis and Rotations of View Platform in Virtual World

By default, the object rotates at a slow speed, and the users can use a scrollbar to adjust the rotation speed or stop the rotation. A green grid is drawn in the background to represent the floor. Two text fields and one pull down menu are equipped to control the object geometry resolution in the X and Y directions, and the resolution of texture image respectively. Figure 4.2 shows the user interface and the virtual world rendering an experimental object *Nutcracker*. Figure 4.3 shows the object observed from different viewpoints.

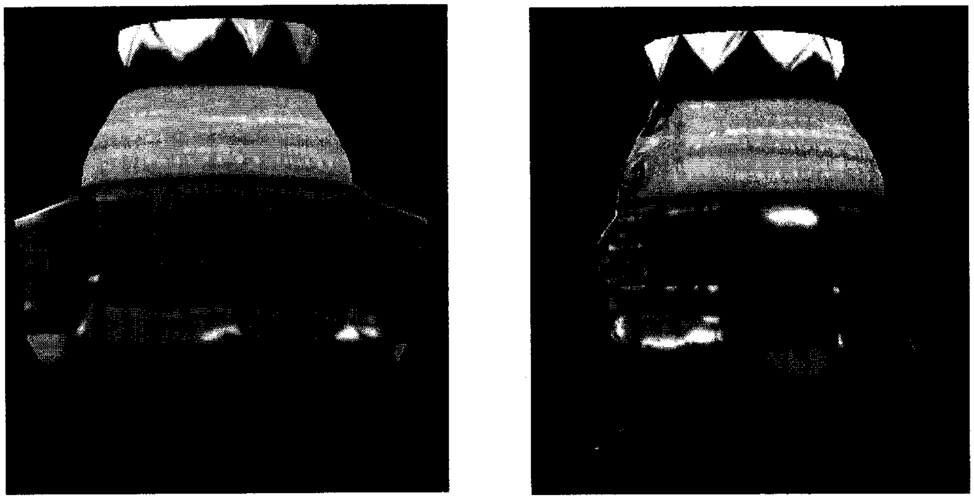


#1, this is the ORIGINAL image, please press button next to continue

Figure 4.2 The User Interface and the 3D Virtual World



(a) Front (b)Left



(c) Rear (d) Right

Figure 4.3 Views of Nutcracker from Multiple Angle

Capture Hardware and Processing

Five 3D objects, named *Pot, Doll, Nutcracker, Head and Dog* are tested in the experiments. These objects are acquired via *Zoomage* 3D scanners. Objects are rotated on a tray and captured by a range scanner. The resolution of the wireframe can be up to 80,000 X 30 polygons for a full 360 degrees, 31 laser-lines scanned model, and texture resolution is up to 5300 pixels (vertical) X 40,000 pixels (horizontal). Since all objects are lit from front in scanning and the viewpoint is fixed in the front of object, the experiments simulate the illumination from front. Figure 4.4 illustrates the scanning process, and Figure 4.5 shows the texture, wireframe, and the canonical view of object Nutcracker.

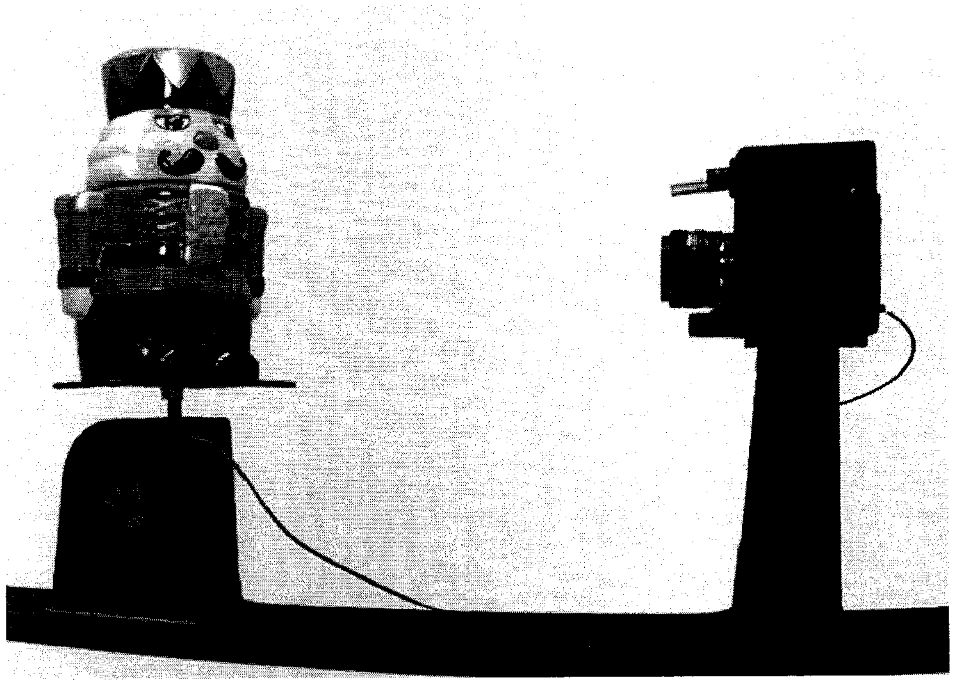
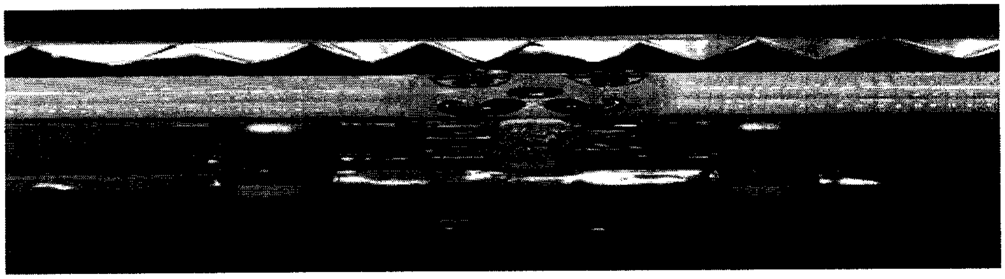
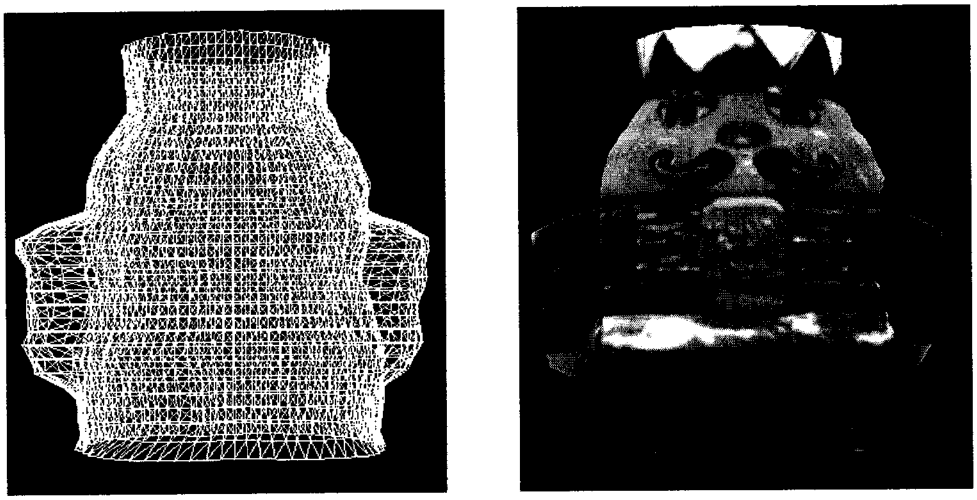


Figure 4.4 The Zoomage 3D Scanner



(a) Texture



(b) Wireframe

(c) Canonical View

Figure 4.5 Texture, Wireframe, and the Canonical View of Nutcracker

The Simplification of Models

Obviously, the original scanned objects are over-sampled models with respect to either geometry or texture resolution. In order to study the quality degradation related to geometry and texture resolution, we simplify the models to such a level that further simplification yields visible degradation.

The reduction of texture resolution is rather straightforward. For a fixed distance from the viewer to the object, the best quality can be achieved when one texel (pixel on texture image) corresponds to one pixel in rendered image. When texture resolution is higher than this level, several texels are averaged to form a pixel and no extra detail is added to the system, thus the texture data is redundant. When texture resolution is lower than this level, one texel represents several pixels and the pattern of texture map is easily observed. In the experiments, all objects are normalized to 1.0 meter high in the virtual world. The view platform is placed at the distance to the object such that the whole object is visible and occupies most visual space in the vertical direction. That is, approximately 750 pixels in height for a display mounted at the resolution of 1280 X 1024. We decrease the resolutions of all texture images 750 pixels in height, which is appropriate to display the visual object in highest resolution without data redundancy. The widths of texture images also decrease in proportional to the heights, but the actual size may vary with different objects.

Many methods are available for simplification of geometry. From the review of model simplification methods presented in Chapter 2, we observe that most error metrics defined in these approaches are specific to their simplification methods. In addition, with the same number of vertices or polygons, visual quality varies remarkably for the same model simplified by different methods. The objective of this thesis is to study how individual factors determine the overall visual quality. Therefore, instead of comparing how well different error metrics model visual fidelity, we only model the visual quality resulted from the simple *Grid Sampling* method with the factors of geometry resolution and texture resolution. Grid sampling keeps a regularly sampled polygonal model across the original surface regardless of the number of vertices. This method is commonly used and extremely efficient, but details in areas with high curvatures are not well preserved with sparse sampling. As for the simplicity and generality of grid sampling, we may be able to connect other simplification methods with this approach in future work. e.g., for an arbitrary simplified polygonal surface, it may be possible to produce a visually-equivalent grid-sampled polyhedron and discuss the quality of simplified model via the evaluation of its equivalent.

In the experiments, the objects are first normalized to 360 X 30 polygons. After that, further simplifications are performed in X-Axis and Y-Axis direction respectively, and each simplification attempt reduces 5% of the vertices in one direction. Three human viewers are requested to compare each pair of continuously simplified objects and the process does not stop until the first pair of visually different objects is found and agreed upon. The simplest polygonal meshes keeping fidelity in this process is used as the full-resolution model in our experiments.

To ensure identical physical luminance, all evaluation experiments are performed on a DELL 1.8GHZ + 512MB + Geforce3 workstation, which is equipped with a 21 inch Trinitron monitor. The resolution of display is 1280 X 1024.

Experimental Process

In the experiments, there are 5 different visual objects, each at 6 levels of wireframe resolutions and 3 levels of texture resolutions, combining 5 X 6 X 3 stimuli. All these stimuli are evaluated by 10 participants, all of whom are students at the Computing Science department and have no prior knowledge of these objects. Two referential stimuli are provided side by side with the rating stimulus for comparison. One is the stimulus with most detailed geometry and highest texture resolution, and the other is the stimulus with coarsest geometry and lowest texture resolution. The quality of the former one is set to the maximum 5 (very good), and the latter one is set to the minimum 1 (very poor). The referential stimuli are rotating at the same speed as the target one, and the adjustment of rotation speed is effective for all three stimuli. The participants are required to compare the target stimulus to the two referential stimuli and rate in one of the following scales:

very poor (1), poor (2), fair (3), good (4), very good (5).

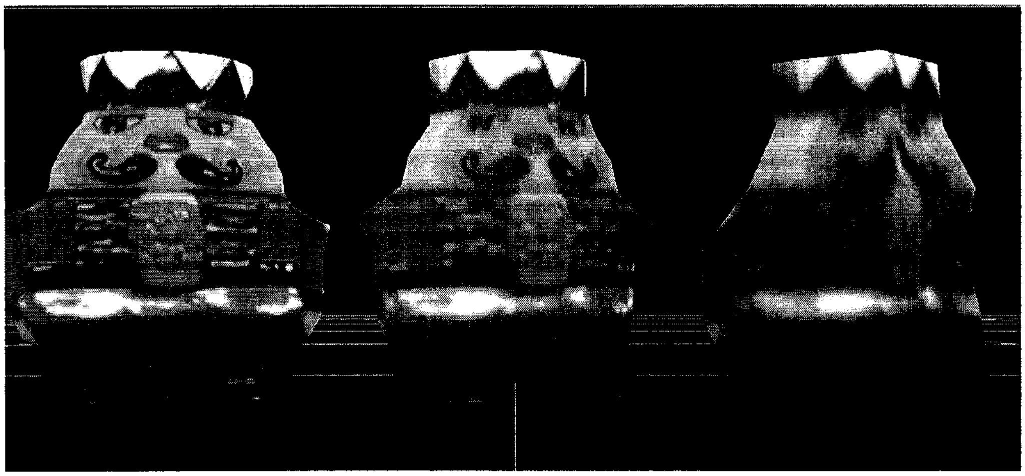


Figure 4.6 Evaluation Example

Figure 4.6 illustrates two referential stimuli (left and right) and one target stimulus (center) in the experiment.

In order to reduce the effect of the temporal sequencing factor, the sequence of these images is randomly generated in a manner that no two participants share the same sequence. In addition, each participant evaluates the images independent of others. All tests are performed on the Dell workstation. Only one object is displayed at a time to a viewer.

4.2 A Metric for Estimating Perceptual Quality

Given the same texture resolution, the quality of images improves with the augmentation of wireframe resolution, which creates a finer geometry. When the wireframe resolution is low, one unit increase in resolution is equivalent to a relatively significant fraction of the current resolution, and results in a significant enhancement of quality; while at the high-resolution end, one unit augmentation in resolution may not be easily detectable. When the wireframe resolution reaches a particular threshold, further increasing is no longer perceivable to the human eyes at a fixed distance, and the best quality is reached for this texture resolution. Thus, the quality curve for images under the same texture resolution and different wireframe resolutions should be approximately exponential, which can be modeled by the following equation:

$$Quality = \frac{1}{a + be^{-cg}} \tag{4.1}$$

Here g is a variable representing the level of detail of geometric data, which is implemented as the square root of the number of vertices on the surface, and *a, b, c* are constant coefficients. We define the minimum quality to be *m*, and the maximum quality to be M. $Q(\infty) = 1/a$, $Q(0) = 1/(a+b) = m$, so we have:

$$a = 1/Q(\infty), \quad b = 1/m - 1/Q(\infty) \tag{4.2}$$

Here $Q(\infty)$ is the image quality with an ideal wireframe and various levels of textures. Since quality varies with texture resolution, $Q(\infty)$ is also a function of texture resolution, which can be represented as $f(t)$, t is texture resolution. Replace *a* and *b* according to (4.2) then:

$$Quality(g,t) = \frac{1}{\frac{1}{f(t)} + (\frac{1}{m} - \frac{1}{f(t)})e^{-cg}} \tag{4.3}$$

In (4.3), $g \in [0, \infty]$, when g is equal to or larger than original geometry G_0, the extra details are not perceptible to the human visual system, thus

$$Quality(\infty,t) = Quality(G_0,t) \tag{4.4}$$

For convenience, it is desirable to normalize the value of geometry to the interval of [0, 1]. We use variable $g'=g/G_0$ to indicate the percentage of geometry, here $g' \in [0, 1]$.

We map g from interval $[0, \infty]$ to [0, 1] using the logarithmic function:

$$g = \log\frac{1}{1-g'} \tag{4.5}$$

Combining (4.3) and (4.5) gives:

$$Quality(g',t) = \frac{1}{\frac{1}{f(t)} + (\frac{1}{m} - \frac{1}{f(t)})(1-g')^{c}} \tag{4.6}$$

There are three problems remained in this quality metric:

(1) How closely the quality curve for images under the same texture resolution and different wireframe resolutions follow the exponential property?

(2) How to resolve the function of texture resolution $f(t)$?

(3) How to determine coefficient c?

We are going to answer the first two questions in the analysis of evaluation results in the next chapter, and discuss the other question in a specific distributed application.

Chapter Five

Experimental Results and Analysis

5.1 Experimental Results

The experimental results and figures related to each object are listed at the end of this chapter. Table 5.4, 5.6, 5.8, 5.10, 5.12 enumerate the average of a total of 10 participants' quality evaluation values for 5 X 18 stimuli. The tables are indexed by the percentage of ideal (the minimum size visual equivalent of original model) geometry and texture resolution. Table 5.5, 5.7, 5.9, 5.11, 5.13 give the standard deviation of these evaluations values. Values in Table 5.4, 5.6, 5.8, 5.10, 5.12 are plotted in Figure 5.2, 5.5, 5.8, 5.11, 5.14.

From the figures noted above, we can observe that visual quality roughly follows exponential property as we predicted. For each level of texture resolution, quality decreases slowly with high geometry resolution, and drops quickly at the low geometry resolution end. Some exceptional points exist in these figures, for example, in Table 5.4, the sample point Q(g, t) = Q(87%, 25%) has the value 2.30, and Q(100%, 25%) has the value 2.20, but Q(100%, 25%) is supposed to be larger than Q(87%, 25%) for a better geometry resolution. Several factors combine to produce such exceptional points. First, stimuli are evaluated in a randomized sequence, and each stimulus is compared to the referential stimuli independently, only one target appears at a time. Viewers evaluate the quality according to the proximity between target stimulus and referential stimuli. If the viewer feels that the target is closer to the best quality stimulus, he/she will give a rating of 5 or 4 depending on proximity degree, closer to the other end produces a result of 1 or 2; if the viewer has no preference, he/she probably will rate it at the middle value of 3. Therefore, the target stimuli are not directly compared to one another, and an evaluation error is likely to occur for two stimuli with close quality. The results follow this observation in that all the exceptional points are located at the high geometry resolution regions, where visual quality changes slightly with the variation of geometry resolution. Second, the number of participants, 10, is relatively small, and any exceptions such as viewer illusion or operational mistakes (values are recorded from choosing one of five buttons) can change the average from 0.1 or 0.2. Although such errors are unpredictable, with a large group of participants the effect of the errors can be reduced. Fortunately, evaluation errors introduced in the results are generally small and do not change the overall curves, hence we did not screen the sample points in our experiments.

The values of standard deviations (SD) (Table 5.5, 5.7, 5.9, 5.11, 5.13) indicate how much the participants agree with one another. SDs are as high as 1.15 and as low as 0.32 (Note the quality of best and worst stimuli are pre-set at the values of 5 and 1 respectively, and SDs at these two points are always 0.00), most points fall into the interval of [0.6, 0.9]. Compared with the dynamic range [1, 5], the values are slightly high. This, again, is largely the effect of a relatively small number of participants and the absence of a screening process. The coarse granularity of quality description, to a lesser extent, may also be a contributing factor. For example, if half of the viewers rate a stimulus at 2, and another half rate it at 3, probably many will desire 2.5 if such a scale is available, thereby reducing the SD. On the other hand, a finer granularity gives the viewers more freedom in choosing an outlier value, and a screening process becomes more important. With the limited number of participants, it may be more appropriate to use the current coarse 5-scale system. It should be noted that SDs are visibly larger in the upper left corner in the tables. Most SDs larger than 1.00 belong to the first column. This observation gives us the insight that people are more accordant in assessing the variance of texture resolution than that of geometry resolution. The acceptance of "low" geometry resolution varies distinctly among people.

In the last chapter, the quality metric model is derived as:

$$Quality(g,t) = \frac{1}{\frac{1}{f(t)} + (\frac{1}{m} - \frac{1}{f(t)})(1-g)^c} \qquad (5.1)$$

According to (5.1), we perform curve fitting minimizing the sum of square errors over each of the five datasets. The results are illustrated in Figure 5.3, 5.6, 5.9, 5.12, 5.15. In the legends, $a = 1/f(t)$, $b = 1/m - 1/f(t)$, and c is the same as the one in Equation (5.1).

Object / Texture	Nutcracker	Head	Dog	Doll	Pot
SD (Texture 100%)	0.21	0.29	0.30	0.08	0.47
SD (Texture 50%)	0.21	0.18	0.24	0.15	0.13
SD (Texture 25%)	0.18	0.45	0.31	0.24	0.37
SD_{all}	0.22	0.34	0.32	0.20	0.36
c	4.34	4.39	4.59	4.19	7.03

Table 5.1 Standard Deviation of Residuals with Experimental *f(t)*

In Table 5.1, the first three rows are the Standard Deviation of the vertical distances from the sample points to the associated fitting curves of three-level texture resolutions. The quality values of best and worst stimuli are fixed and produce no error, thus the sizes of the groups are counted as 5, 6, and 5 respectively. Since the distances of the points from the fitting curves are also called residuals, SD is the standard deviation of the residuals for each texture level, as denoted by (5.2), in (5.2), r is the value of residual, and N is the size of the group.

$$SD = \sqrt{\frac{\sum_i r_i^2}{N-1}} \tag{5.2}$$

The overall standard deviation SD_{all} is calculated as (5.3).

$$SD_{all} = \sqrt{\frac{\sum_i r_i^2}{df}} \tag{5.3}$$

Here df is degree of freedom and equal to the number of sample points minus the number of parameters fitted. $df = 16 - 2 = 14$ in our experiments.

Despite of the poor performance of Object "Pot", all the four other objects have a good fit. The standard deviation ranges from 0.20 – 0.34, which is less than 10% of the dynamic range. The exponential coefficient c has a reasonable value around 4.40. More importantly, sample points are fairly evenly distributed around the fitting curve. The SD_{all} value of Pot is not large either, but an exponential coefficient as high as 7.03 describes a curve that is nearly constant in the intermediate to high geometry resolution, and drops abruptly for low geometry resolution. A closer examination reveals uneven deviation of sample points around the fitted curve. The points are close to a straight line in higher geometry resolution area and rather poorly fit in lower geometry resolution area. Thus, the fitting does not serve much better than a constant function. Given more samples in low geometry resolution end, the SD_{all} for this example will become too large to be acceptable.

Note the sample point $Q(g, t) = Q(26\%, 25\%)$, which is used as the referential stimulus of worst quality, is excluded in the curve fitting. In the derivation of (5.1), we assume $Q(0, t) = m$, since an object without geometry has no visual appearance, for comparison purpose, $Q(26\%, 25\%) = m$ is set. Consequently, this sample point does not follow (5.1). As a matter of fact, if $c=4.4$ and $f(t)=2.0$, $Q(26\%, 25\%)=1.58$. With a finer-granularity rating system in which $x.5$ is available, 1.5 is a better setup value for the worst quality referential stimulus. In our experiments, the worst quality reference is under-estimated. For that reason, sample points close to this one are likely to be under-estimated as well. This is also evident in the Figures and Table 5.1, the values of sample points at 25% texture resolution decrease faster than predicted, resulting in comparably high value of standard deviation (Row 3 in Table 5.1).

Once the exponential relation between geometry parameter and visual quality is determined, the problem remained is to study the property of quality related to texture resolution. That is, how to resolve $f(t)$ in Equation (5.1). $f(t)$ is the visual quality for maximum geometry resolution and various texture resolutions. More samples are collected to resolve this function. For each object other than Pot, 4 more stimuli are tested. All of these stimuli have maximum geometry resolution and different texture resolutions. Combined with samples from former experiments, there are 6 samples along the Quality Axis direction for each object in 100% geometry resolution. The data are listed in Table 5.2 and plotted in Figure 5.1.

Texture Object	25%	37.5%	50%	62.5%	75%	87.5%
Nutcracker	2.2	2.5	3.2	3.6	4.1	4.5
Head	2.3	2.6	3.1	3.3	3.9	4.3
Dog	2.2	2.6	3.2	3.7	4.2	4.7
Doll	2.0	2.5	3.2	3.6	4.1	4.6

Table 5.2 Quality vs. Texture Resolution (100% Geometry Resolution)

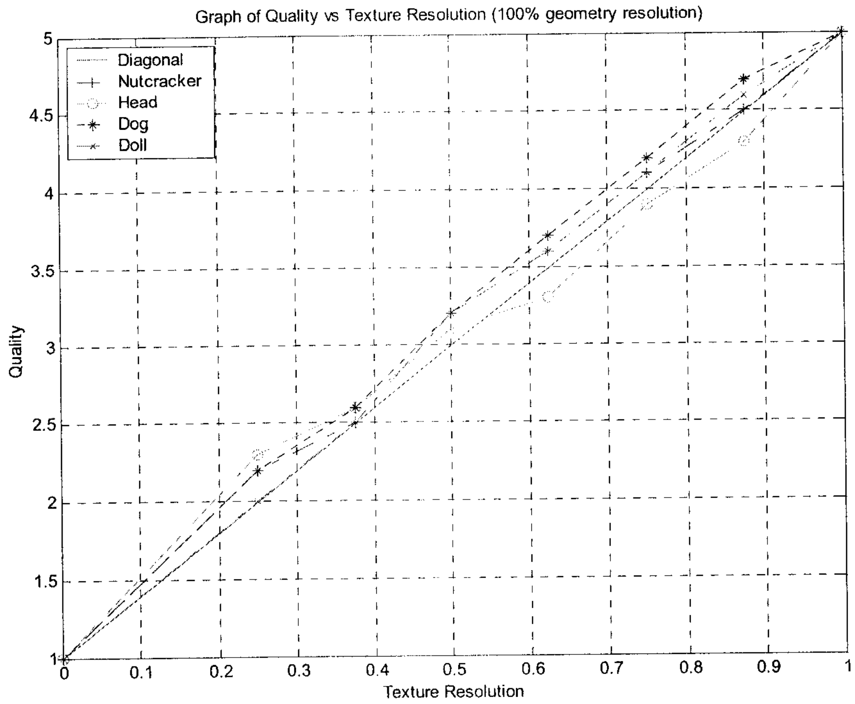


Figure 5.1 Quality vs. Texture Resolution (100% Geometry Resolution)

In Figure 5.1, all samples points are distributed close to the diagonal, that is, *f(t)* approximately follows linear regression.

$$f(t) = m + (M - m)t, \ t \in [0,1] \tag{5.4}$$

Replacing *f(t)* in (5.1) produces:

$$Quality(g,t) = \frac{1}{\frac{1}{m+(M-m)t} + (\frac{1}{m} - \frac{1}{m+(M-m)t})(1-g)^c} \quad (5.5)$$

We perform a curve fitting based upon Equation (5.5) for validation. The results are listed in Table 5.3 and plotted in Figure 5.4, 5.7, 5.10, 5.13, 5.16. Since the values of *f(t)* is from Equation (5.4), we use the notation "estimated *f(t)*" in distinction to the former results in which the values of *f(t)*are from experiments.

Object / Texture	Nutcracker	Head	Dog	Doll	Pot
SD (Texture 100%)	0.18	0.25	0.27	0.11	0.47
SD (Texture 50%)	0.16	0.16	0.18	0.20	0.13
SD (Texture 25%)	0.19	0.29	0.23	0.27	0.37
SD_{all}	0.19	0.25	0.26	0.22	0.36
c	4.52	4.59	4.41	4.38	7.03

Table 5.3 Standard Deviation of Residuals with Estimated *f(t)*

Comparing Table 5.3 with Table 5.1, the SD_{all} value of Doll remains nearly the same, but the SD_{all} values of the other three objects become smaller, which means a comparatively better fitting. This phenomenon can be explained as follows. The former fitting ensures exact matches along 100% geometry resolution axis, and matching in other regions is not as good. However, the experimental values of *f(t)* may have accidental errors. Therefore, the entire sample space, more than samples in 100% geometry resolution axis alone, are conforming to Equation (5.5). We notice that the exponential coefficient *c*, although not a constant for different objects, varies inside a narrow interval. For those applications that do not require high accuracy, 4.50 can be used as an empirical constant, otherwise, some samples should be collected to better determine this parameter.

Our metric does not estimate well the quality of object Pot. From the feedback of viewers, we found that while users were requested to rate based on comparing the target stimuli to two referential stimuli, not all of them follow the guideline equally well. The first impression played an important role in their evaluations. Since square pots exist in real world, when the geometry resolution of Pot is decreased, some of them felt "pot can be square, not necessarily round", and unconsciously gave a higher than expected rating to some distorted stimuli. Psychologically, this proves prior knowledge is an important issue in quality evaluation. For instance, the quality degradation of a face is easier to detect than that of a rock because people are more familiar with the structure of a face. In our experiments, besides of referential stimuli, sometimes their knowledge also serves as a pattern in the matching. Although such effects cannot be bypassed in any psycho-visual experiments, cautious selection of stimuli may minimize such effects. In conclusion, the Pot object is not an appropriate candidate for our experiments.

5.2 Contributions and Limitations

Compared with the related perceptual experiments [Rogo01][Wats00][Wats01], in which none of the metrics model perceptual quality well and only comparative accuracies of different metrics are provided, our research excels in providing a perceptual-based and qualitative metric that accurately fits the evaluation results. In addition, while previous experiments only consider non-textured model, our metric describes how texture resolution as well as geometry resolution control the overall quality of 3D images. The experiments in [Wats00][Wats01] use still 2D images as stimuli, and Rogowitz *et al.* proved it is not appropriate by comparing the results of view-dependent 2D and view-independent 3D stimuli evaluations [Rogo01]. We also advance a step further upon Rogowitz *et al.* to provide a rotation speed control for visual stimuli. Given such flexibility in the experimental interface, the spatial factor can be reduced. Viewers have a more comprehensive impression and are more affirmatory in rating the quality.

However, there are several limitations inherent in the proposed metric. First, it is not a generic metric that can be applied for any models. Instead, it is specific to models simplified by grid sampling. To generalize this metric, it is necessary to find a method to generate the evenly sampled visual equivalent of an arbitrary textured model. This will become an important and hard topic in our future work. Second, the number of stimuli and participants is relatively small in our experiments, for a robust conclusion, more stimuli and participants are required. Third, we use human viewers to perform the reduction from the original model to the simplest visual equivalent, this process is inconvenient and tedious. It is desirable to automate this process. A number of researches in images synthesis have been dedicated to this area.

5.3 Use of the Metric on a Bandwidth-Constrained System

A distributed graphics system is usually a constrained system with limited computational resource and network bandwidth. Our metric models the relation between perceptual quality and the factors of geometric complexity and texture resolution, and it can be applied to utilize the limited bandwidth more efficiently.

The distributed system includes one server and multiple clients. 3D scenes are rendered on clients' hosts, while the information of 3D objects is stored on the server. When a user requests an object, he/she sets a tolerable delay interval. According to the current network bandwidth, an estimate of maximum transmission size can be computed and packed in the client request. Upon receiving the request, the server adaptively trades off the transmission of geometry and texture data to produce maximum quality with the given data size.

If perceptual quality is denoted by *Q(g, t)* as in (5.1), and *SG(g)*, *ST(t)* represent the functions of geometry and texture data size respectively, the problem can be described as: Seeking *Max(Q(g, t))* under the constraint (5.6)

$$SG(g) + ST(t) = C \tag{5.6}$$

According to Euler-Lagrange equation, we have

$$\frac{\partial Q / \partial g}{\partial SG / \partial g} = \frac{\partial Q / \partial t}{\partial ST / \partial t} \tag{5.7}$$

The value of g and t can be determined from (5.6) and (5.7). The assumption here is that the computation delay is negligible or very small compared to the transmission delay.

Geometry Texture	26%	39%	55%	71%	87%	100%
100%	2.80	3.50	4.60	4.70	4.90	5.00
50%	1.80	2.40	2.60	3.00	3.10	3.20
25%	1.00	1.60	2.00	2.00	2.30	2.20

Table 5.4 Mean Quality of Users' Evaluations: Nutcracker

Geometry Texture	26%	39%	55%	71%	87%	100%
100%	1.15	0.85	0.70	0.42	0.48	0.00
50%	0.82	0.82	0.92	0.67	0.97	0.79
25%	0.00	0.84	0.82	0.82	0.67	0.79

Table 5.5 Standard Deviation of Users' Evaluations: Nutcracker

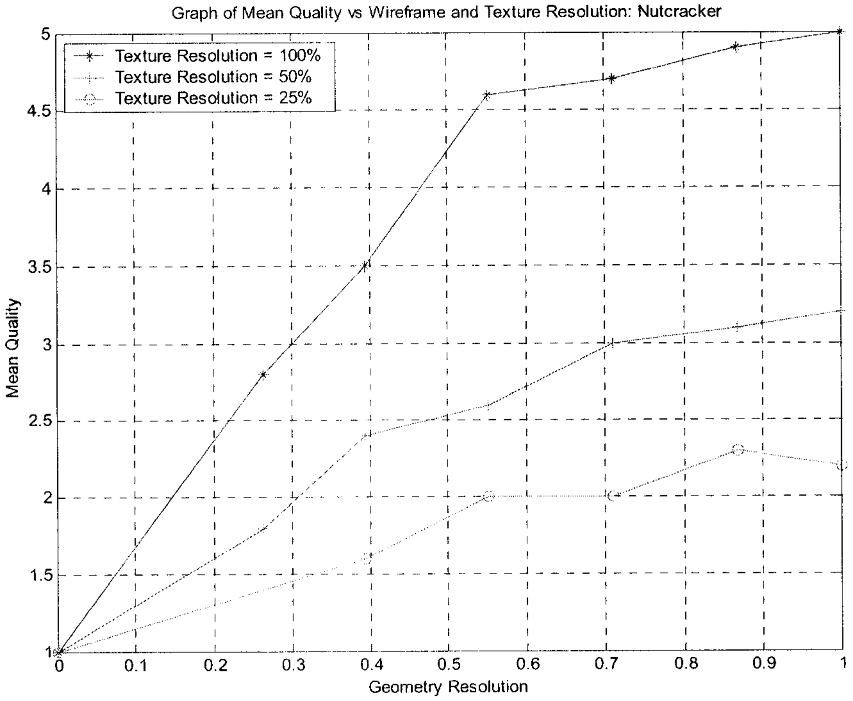


Figure 5.2 Mean Quality vs. Wireframe and Texture Resolution: Nutcracker

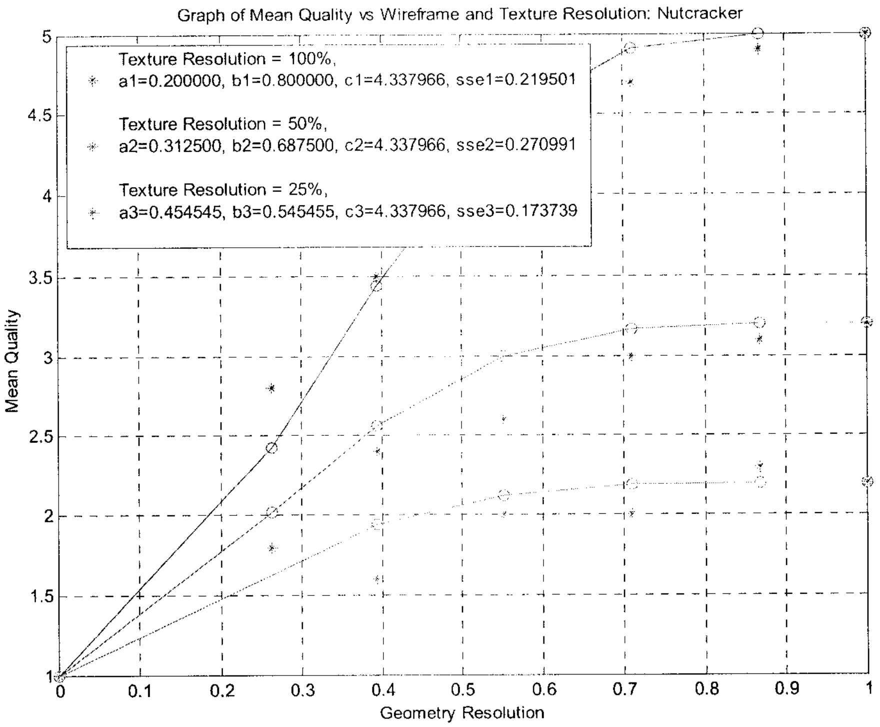


Figure 5.3 Best Fitting with Experimental *f(t)*: Nutcracker

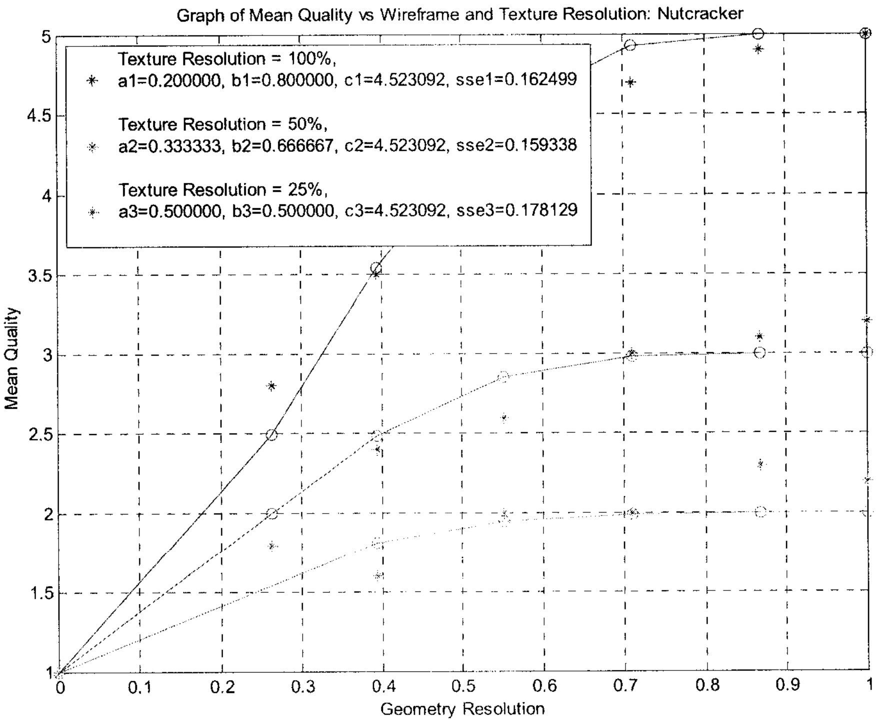


Figure 5.4 Best Fitting with Estimated *f(t)*: Nutcracker

Geometry Texture	26%	39%	55%	71%	87%	100%
100%	2.90	3.80	4.30	4.70	4.90	5.00
50%	1.70	2.30	2.80	3.10	3.00	3.10
25%	1.00	1.30	1.70	1.90	2.00	2.30

Table 5.6 Mean Quality of Users' Evaluations: Head

Geometry Texture	26%	39%	55%	71%	87%	100%
100%	0.92	0.82	0.67	0.88	0.85	0.00
50%	0.71	0.57	0.63	0.32	0.84	0.48
25%	0.00	0.52	0.48	0.57	0.52	0.71

Table 5.7 Standard Deviation of Users' Evaluations: Head

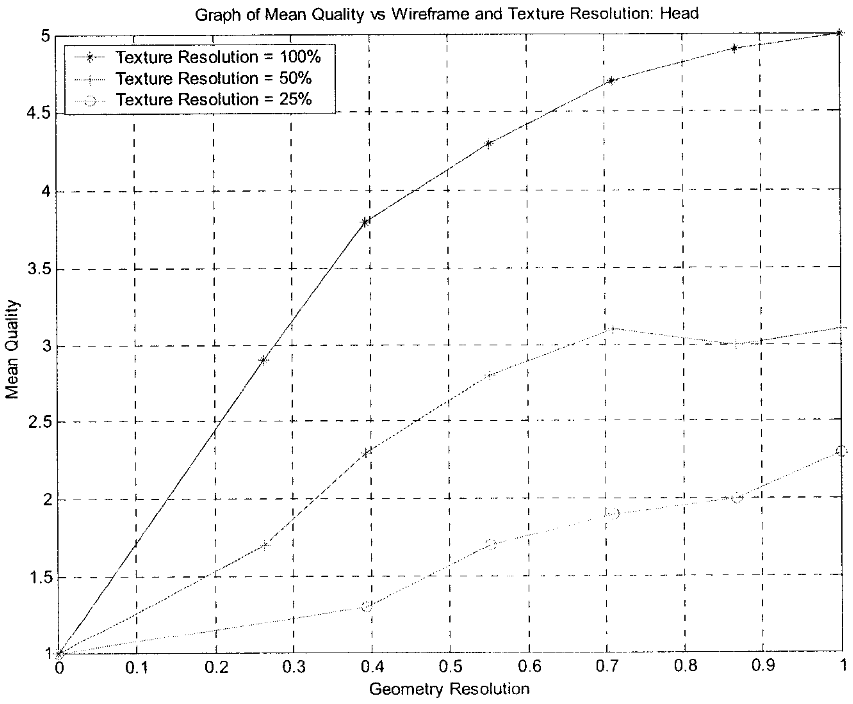


Figure 5.5 Mean Quality vs. Wireframe and Texture Resolution: Head

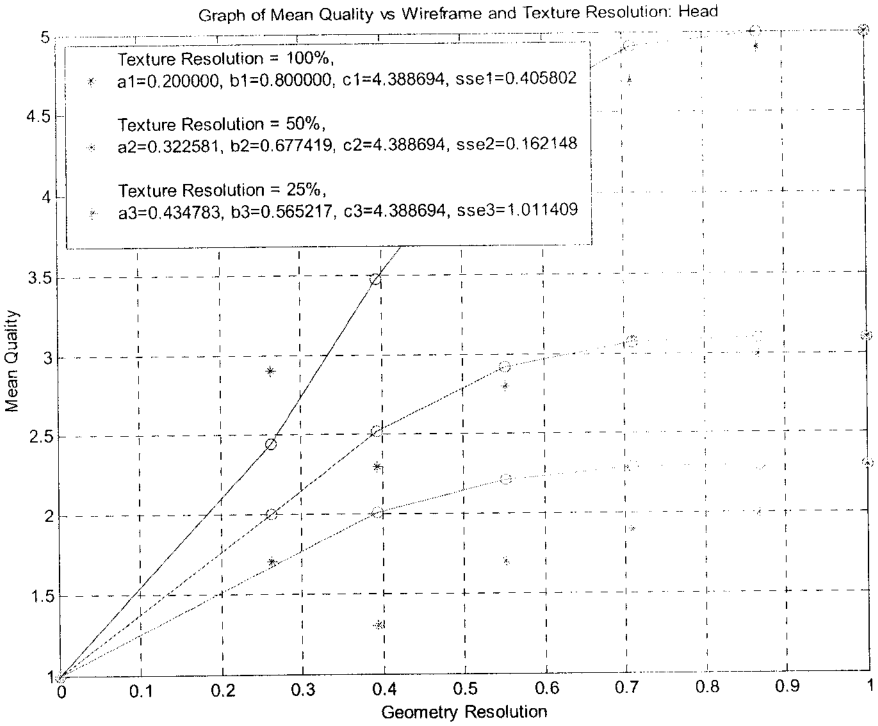


Figure 5.6 Best Fitting with Experimental *f(t)* : Head

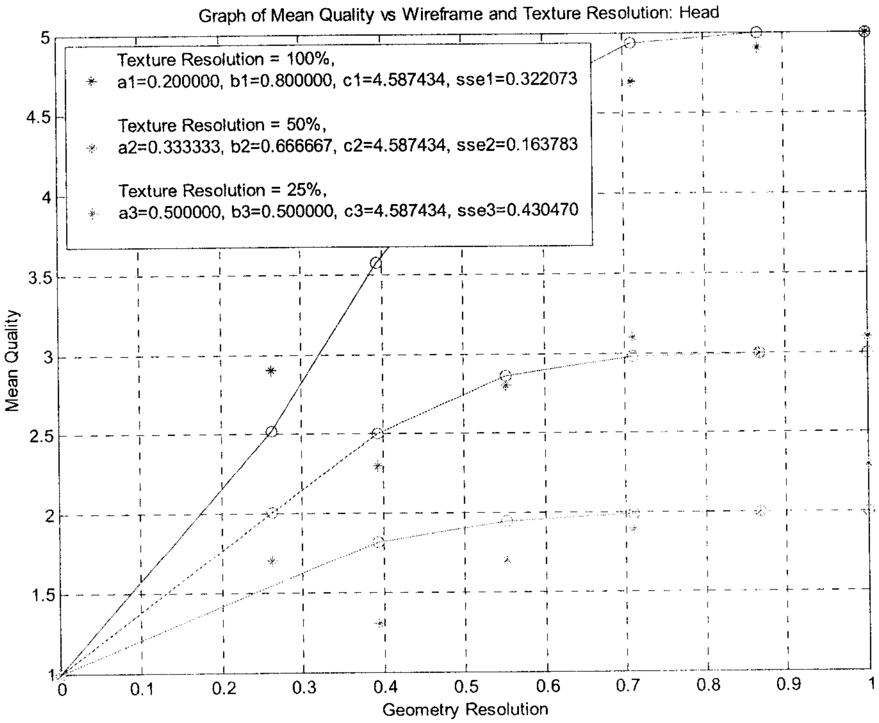


Figure 5.7 Best Fitting with Estimated *f(t)*: Head

Geometry / Texture	26%	39%	55%	71%	87%	100%
100%	2.90	3.60	4.40	4.60	4.80	5.00
50%	1.80	2.20	2.60	3.00	3.00	3.20
25%	1.00	1.40	1.70	2.00	2.10	2.20

Table 5.8 Mean Quality of Users' Evaluations: Dog

Geometry / Texture	26%	39%	55%	71%	87%	100%
100%	0.95	0.57	0.82	0.82	0.53	0.00
50%	1.10	0.97	0.79	0.53	0.67	0.67
25%	0.00	0.52	0.67	0.95	0.70	0.70

Table 5.9 Standard Deviation of Users' Evaluations: Dog

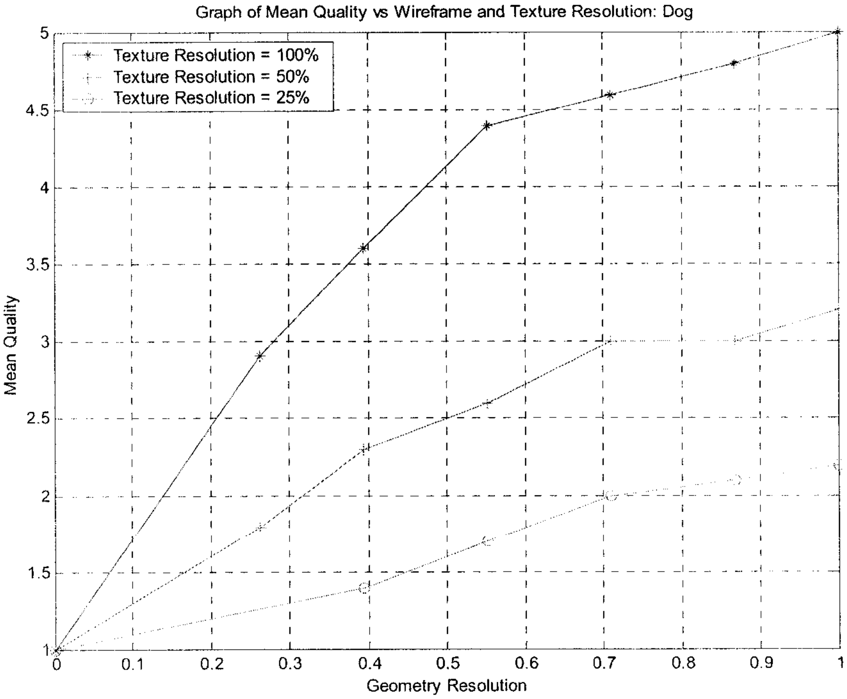


Figure 5.8 Mean Quality vs. Wireframe and Texture Resolution: Dog

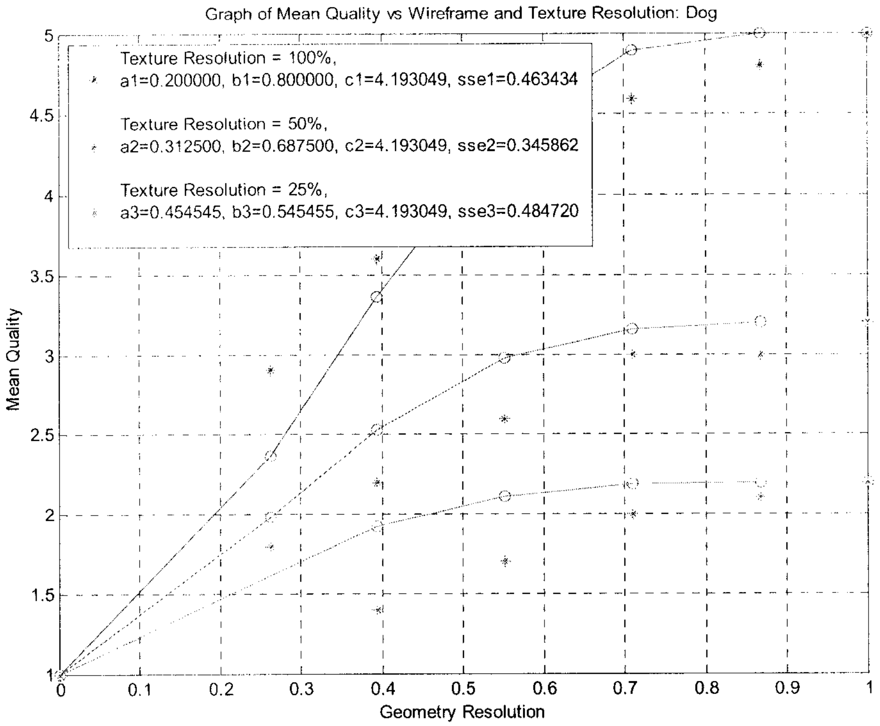


Figure 5.9 Best Fitting with Experimental *f(t)* : Dog

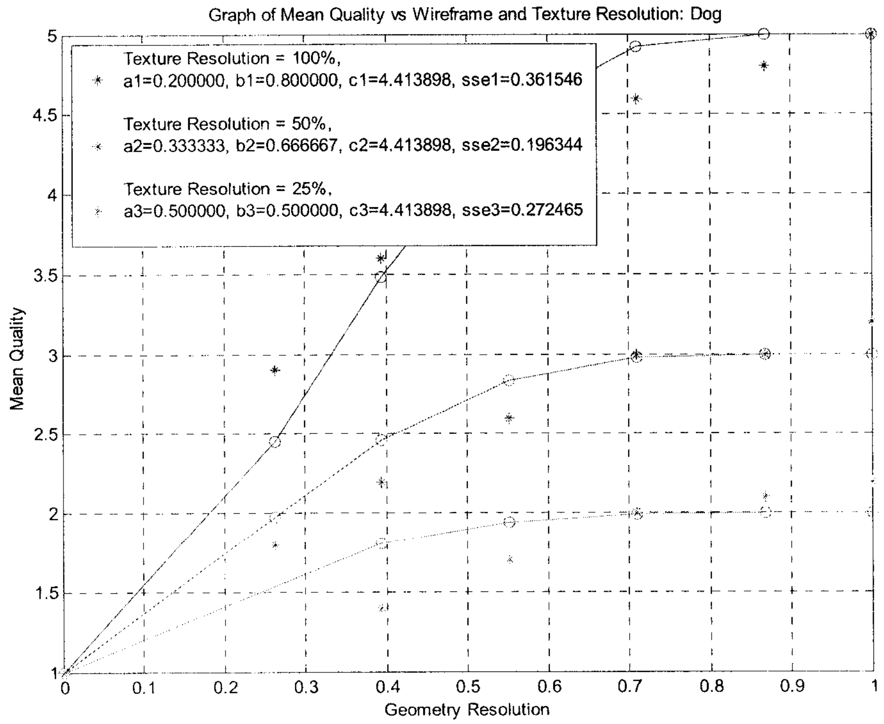


Figure 5.10 Best Fitting with Estimated *f(t)*: Dog

Geometry Texture	26%	39%	55%	71%	87%	100%
100%	2.60	3.60	4.50	4.90	4.90	5.00
50%	1.80	2.60	2.80	2.80	3.10	3.20
25%	1.00	1.40	1.60	1.70	1.90	2.00

Table 5.10 Mean Quality of Users' Evaluations: Doll

Geometry Texture	26%	39%	55%	71%	87%	100%
100%	1.07	1.07	0.53	0.32	0.32	0.00
50%	1.07	0.79	0.74	0.84	0.85	0.85
25%	0.00	0.42	0.48	0.94	0.70	0.71

Table 5.11 Standard Deviation of Users' Evaluations: Doll

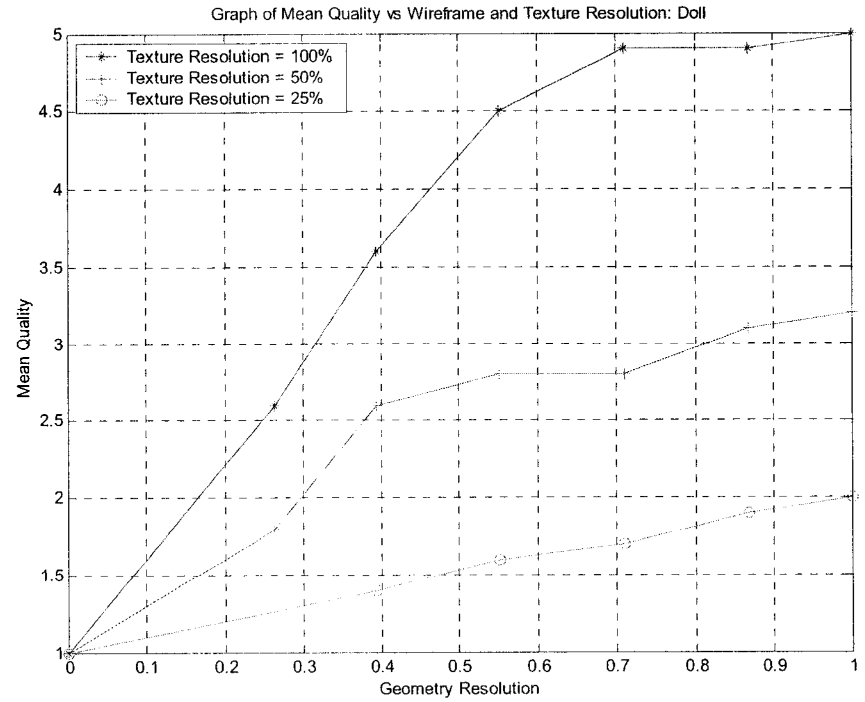


Figure 5.11 Mean Quality vs. Wireframe and Texture Resolution: Doll

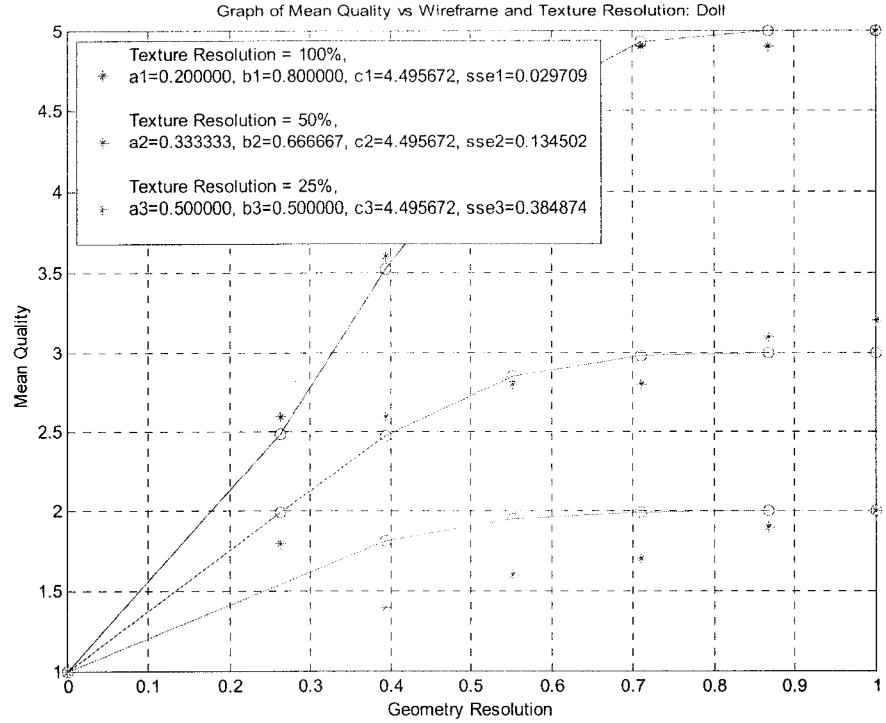


Figure 5.12 Best Fitting with Experimental *f(t)*: Doll

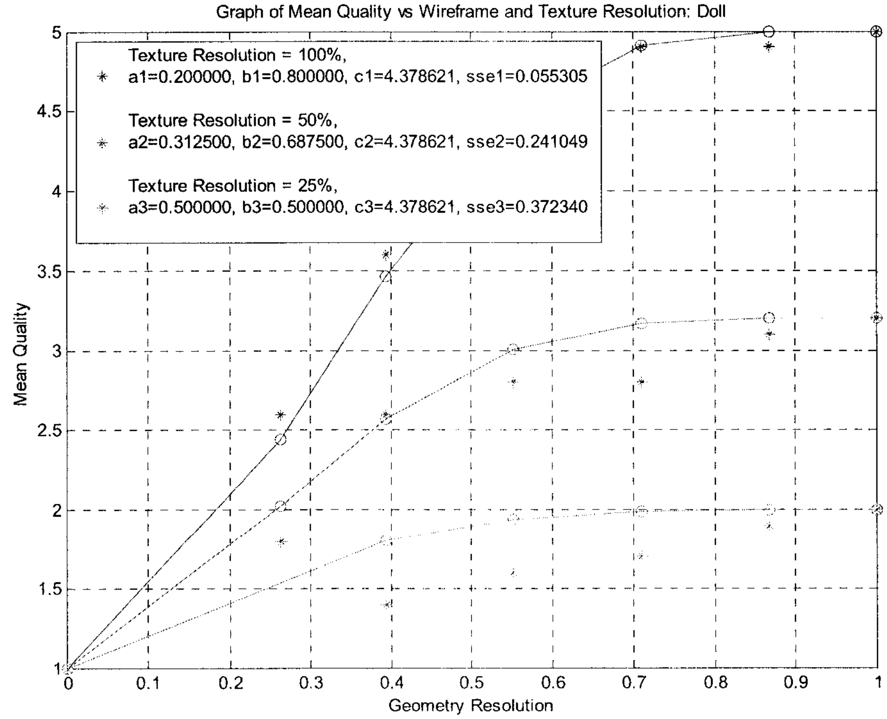


Figure 5.13 Best Fitting with Estimated *f(t)*: Doll

Geometry Texture	26%	39%	55%	71%	87%	100%
100%	3.80	3.90	4.30	4.70	4.80	5.00
50%	2.90	3.20	3.20	3.30	3.30	3.30
25%	1.00	1.20	1.40	1.90	1.90	1.90

Table 5.12 Mean Quality of Users' Evaluations: Pot

Geometry Texture	26%	39%	55%	71%	87%	100%
100%	1.03	1.06	0.67	0.67	0.95	0.00
50%	1.03	0.71	0.71	0.74	0.94	0.63
25%	0.00	0.52	0.52	1.10	0.99	0.74

Table 5.13 Standard Deviation of Users' Evaluations: Pot

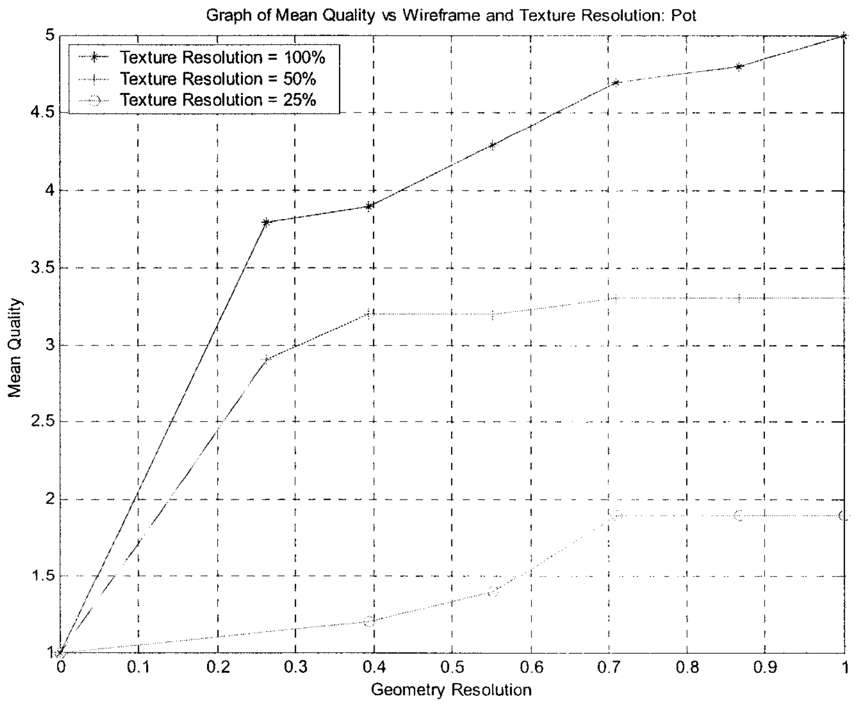


Figure 5.14 Mean Quality vs. Wireframe and Texture Resolution: Pot

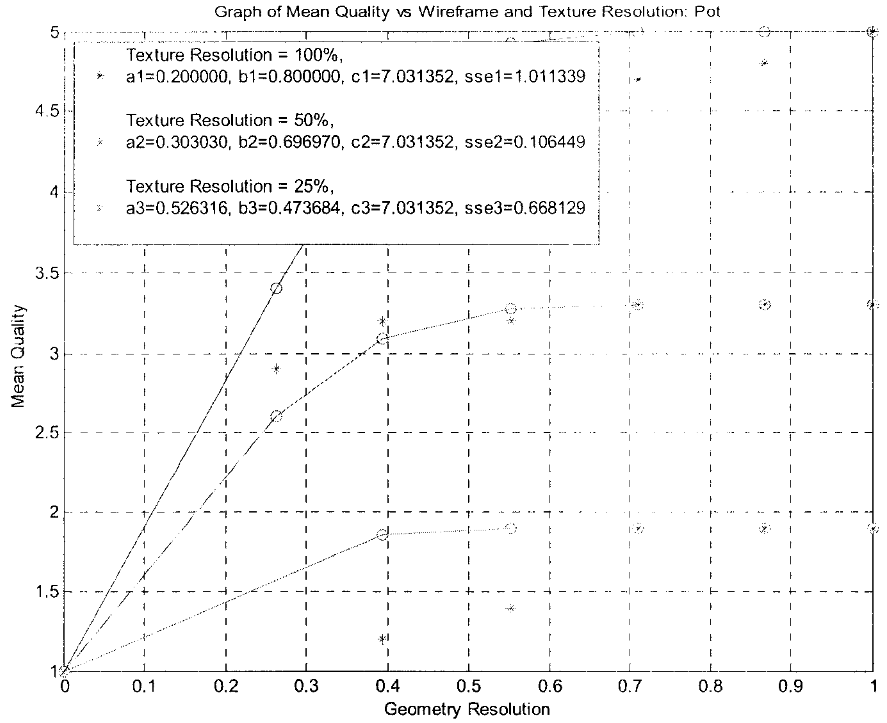


Figure 5.15 Best Fitting with Experimental *f(t)*: Pot

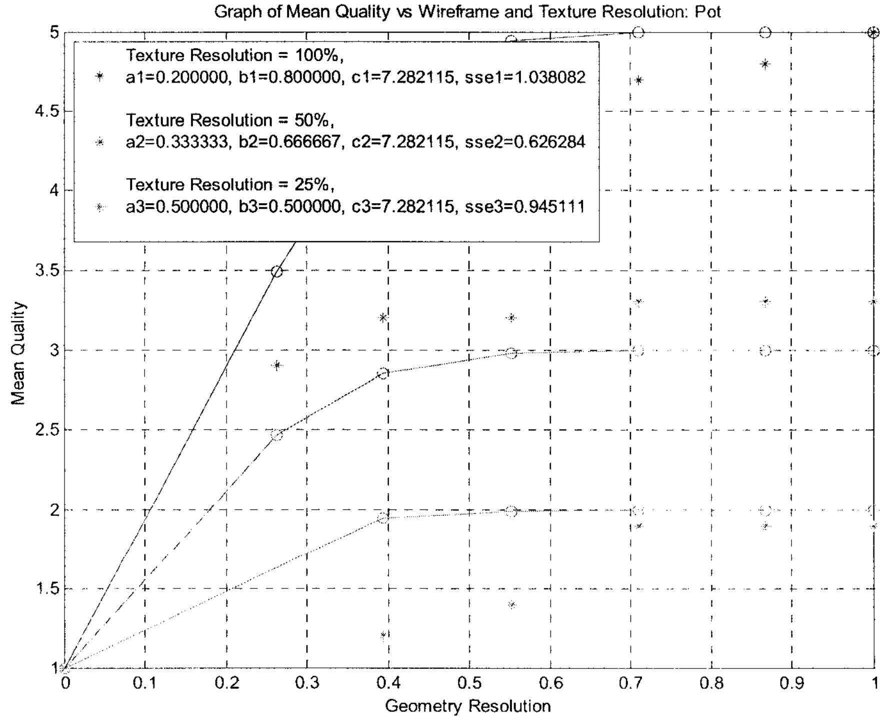


Figure 5.16 Best Fitting with Estimated *f(t)*: Pot

Chapter Six

Conclusions and Future Work

6.1 Conclusions

In this thesis, we provide a detailed survey of model simplification methods and associated error metrics. In the previous perceptual experiments modeling these metrics to visual evaluation, the results suggest that such error metrics are not good indicators of perceptual quality. Based on the observation above, we examine the factors that determine the quality of 3D images including geometry resolution, texture resolution, shading complexity, frame rate and other psycho-visual factors. We design a perceptual experiment and derive from the results a statistical metric, which reflects how geometry resolution and texture resolution control the overall quality of 3D images. The strengths and limitations of this metric are analyzed, along with a description of applying this metric in a bandwidth-constrained system.

From the figures plotting the experiment results, we observe that the degradation of visual quality follows exponential regression for geometry resolution parameter, and linear regression for texture resolution parameter. This suggests that human viewers are far more sensitive to the distortion of texture image than to that of geometry.

6.1 Future Work

We plan to increase the reliability of our conclusion by increasing the number of models and participants. In that case, we also want to check whether a finer-granularity rating system provides a better result as suggested by current experiments. To estimate quality of arbitrary distorted models, we need to explore a method to generate evenly sampled visual equivalent for a model. Automatic simplification methods should be adopted to obtain the simplest ideal model, which alleviates the preprocessing efforts, especially if a large number of models are used as stimuli. Finally, it is important to incorporate more factors such as distance, shading, and visual masking into the existing metric.

References

[Baja96] C. L. Bajaj and D.R. Schikore. Error bounded reduction of triangle meshes with multivariate data. *SPIE*, 2656:34–45, 1996.

[Basu98] A. Basu, K. J. Wiebe, "Enhancing Videoconferencing Using Spatially Varying Sensing. *IEEE Transactions on Systems, Man, and cybernetics Part A: Systems and Humans,* Vol.28, No.2 March 1998,

[Boli98] Mark R. Bolin and Gary W. Meyer. A Perceptually Based Adaptive Sampling Algorithm. *Proceedings of SIGGRAPH'98*, 299-309, 1998.

[Care97] The Virtual Reality Modeling Language Specification, ISO/IEC DIS 14772-1:1997, http://www.web3d.org/Specifications.

[Ciam97] A. Ciampalini, P. Cignoni, C. Montani, and R. Scopigno. Multiresolution decimation based on global error. *The Visual Computer*, 13(5):228–246, June 1997.

[Cign97] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms. In Computers & Graphics, volume 22, 1997. Pergamon Press.

[Cohe96] Jonathan Cohen, Amitabh Varshney, Dinesh Manocha, Greg Turk, Hans Weber, Pankaj Agarwal, Frederick Brooks, and William Wright. Simplification envelopes. In *SIGGRAPH '96 Proc.*, pages 119–128, Aug. 1996. http://www.cs.unc.edu/_geom/envelope.html

[Cohe98] Cohen, J., Olano, M. and Manocha, D. Appearance-preserving simplification. *SIGGRAPH* 1998 pp. 115-122

[Comp90] CompuServe Incorporated, *GIF GRAPHICS INTERCHANGE FORMAT*, Columbus, Ohio, Jul. 1990 http://astronomy.swin.edu.au/~pbourke/dataformats/gif/

[Daly93] Daly, S. The visible differences predictor: an algorithm for the assessment of image fidelity. In Watson, A. B. (ed.). *Digital Images and Human Vision. MIT Press*, Cambridge, MA, 1993, 179-206.

[Deer95] Michael Deering. "Geometry Compression." *Computer Graphics Proceedings*, Annual Conference Series, 1995, ACM SIGGRAPH, pp 13-19

[Dumo01] Reynald Dumont, Fabio Pellacini, James A. Ferwerda , A Perceptually-Based Texture Caching Algorithm for Hardware-Based Rendering *Eurographics Workshop on Rendering* 2001.

[Ferw96] J. A. Ferwerda, S. N. Pattanaik, P. Shirley, and D. P. Greenberg. A model of visual adaptation for realistic image synthesis. *In SIGGRAPH '96 Conference Proceedings, Annual Conference Series*, pages 249--258. ACM SIGGRAPH, Addison Wesley, August 1996. Held in New Orleans, Louisiana, 4--9 August 1996.

[Ferw97] J.A. Ferwerda, S.N. Pattanaik, P. Shirley, and D.P. Greenberg, A Model of Visual Masking for Computer Graphics , *Proceedings of SIGGRAPH 97*, pp 143 152.

[Funk93] T. A. Funkhouser and C. H. Sequin. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. *Computer Graphics Annual Conference Series,* pages 247--254, August 1993.

[Garl97] M Garland and P.S. Heckbert. Surface simplification using quadric error metrics. In *Comp. Graph. Proc., Annual Conf. Series (Siggraph '97), ACM Press,* 1997.

[Garl98] Garland, M. and Heckbert, P. S. Simplifying surfaces with Color and Texture using Quadric Error Metrics. In *IEEE Visualization* '98 Proceedings, October 1998, pp. 263-269

[Giro93] Girod, B. What's wrong with mean-squared error? In Watson, A. B. (ed.). *Digital Images and Human Vision.* MIT Press, Cambridge, MA, 1993. 207-220. A. Baddelay (Eds.), Attention & Performance IX, Hillsdale, NJ : Erlbaum, 135-151

[Guiz95] Guiziec, A. Surface simplification with variable tolerance. In *Proceedings of the Second International Symposium on Medical Robotics and Computer Assisted Surgery* (November 1995), pp. 132-139.

[Haeb93] Paul Haeberli and Mark Segal, Texture Mapping as a Fundamental Drawing Primitive. June 1993 http://www.sgi.com/grafica/texmap.

[Heck94] Paul S. Heckbert and Michael Garland. Multiresolution modeling for fast rendering. In *Proc. Graphics Interface'94*, pages 43–50, Banff, Canada, May 1994. Canadian Inf. Proc. Soc. http://www.cs.cmu.edu/~ph.

[Hink93] Paul Hinker and Charles Hansen. Geometric optimization. In *Proc. Visualization '93*, pages 189–195, San Jose, CA, October 1993

[Hopp93a] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proceedings of SIGGRAPH 93 (Anaheim, California, August 1–6, 1993),* ComputerGraphics Proceedings,Annual Conference Series, pages 19–26..

[Hopp93b] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle.

Mesh Optimization. TR 93-01-01, *Dept. of Computer Science and Engineering*, University of Washington, January 1993

[Hopp96]H. Hoppe. Progressive meshes. *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, 1996.

[Hopp99] Hoppe, H. New Quadric Metric for Simplifying Meshes with Appearance Attributes. In *IEEE Visualization* '99 Proceedings, October 1999, pp. 59-66

[Kalv91] A.D. Kalvin, C.B. Cutting, B. Haddad, and M.E. Noz. Constructing topologically connected surfaces for the comprehensive analysis of 3D medical structures. *SPIE Vol. 1445 Image Processing*, pages 247–259, 1991.

[Kalv94] Alan D. Kalvin and Russell H. Taylor. Superfaces: Polyhedral approximation with bounded error. In *Medical Imaging: Image Capture, Formatting, and Display*, volume 2164, pages 2–13. SPIE, Feb. 1994. (Also IBMWatson Research Center tech report RC 19135).

[Kalv96] Alan D. Kalvin and Russell H. Taylor. Superfaces: polygonal mesh simplification with bounded error. *IEEE Computer Graphics and Appl.*, 16(3), May 1996. http://www.computer.org/pubs/cg&a/articles/g30064.pdf.

[Klei96] R. Klein, G. Liebich, and W. Straßer. Mesh reduction with error control. In R. Yagel and G. Nielson, editors, *Proceedings of Visualization '96*, pages 311–318, 1996.

[Lamb96a] Christian J. van den Branden Lambrecht, Olivier Verscheure, "Perceptual Quality Measure using a Spatio-Temporal Model of the Human Visual System," in *Proceedings of the SPIE,* vol. 2668, pp. 450--461, January 1996.

[Lamb96b]. Christian J. van den Branden Lambrecht and Joyce E. Farrel. Perceptual quality metric for digitally coded color images. In *Proceedings European conference on signal processing*, 1996.

[Lind96] Real-Time, Continuous Level of Detail Rendering of Height Fields. Peter Lindstrom, David Koller, William Ribarsky, Larry F. Hodges, Nick Faust. *Proceedings of ACM SIGGRAPH 96,* August 1996, pp. 109-118.

[Lind98] Lindstrom P. and Turk, G. Fast and Memory Efficient Polygonal Simplification.In *IEEE Visualization* '98 Proceedings, October 1998, pp. 279-286

[Lind00] Peter Lindstrom, Greg Turk, Image-Driven Simplification *ACM Transactions on Graphics*, 19(3), July 2000, pp. 204-241

[Low97] Kok-Lim Low and Tiow-Seng Tan. Model simplification using vertex-clustering. In *1997 Symposium on Interactive 3D Graphics*. ACM SIGGRAPH, 1997.

[Lueb01] David Luebke and Ben Hallen, Perceptually Driven Simplification for Interactive Rendering. *Rendering Techniques,* Ed. Steven Gortler and Karol Myszkowski, Springer-Verlag, London (June 2001).

[Mann74] J. L. Mannos, D. J. Sakrison, ``The Effects of a Visual Fidelity Criterion on the Encoding of Images", IEEE Transactions on Information Theory, pp. 525-535, Vol. 20, No 4, (1974)

[Maso94] Mason, A.E.W., Blake, E.H. Automatic Hierarchical Level of Detail Optimization in Computer Animation. *Computer Graphics Forum*, 1997, 16(3), 191-200.

[Nvid02] NVIDIA® Corporation 2002, FEROCIOUS GRAPHICS POWER. http://www.nvidia.com/docs/lo/1467/SUPP/PO_GF4Ti_2.05.02.pdf

[Popo97] Popovic and Hugues Hoppe. Progressive simplicial complexes. Computer Graphics (*SIGGRAPH '97 Proceedings*), pages 217--224, 1997

[Rama99] Mahesh Ramasubramanian, Sumanta N. Pattanaik, Donald P. Greenberg, "A Perceptually Based Physical Error Metric for Realistic Image Synthesis," *Computer Graphics Proceedings, Annual Conference Serie*s, 1999, ACM SIGGRAPH, pp. 73-82.

[Redd96] M. Reddy. Scrooge: Perceptually-driven polygon reduction. *Computer Graphics Forum*, 15(4):191–203, 1996.

[Rogo01] B. Rogowitz and H. Rushmeier "Are image quality metrics adequate to evaluate the quality of geometric objects?" *SPIE Proceedings*, Volume 4299, 2001

[Ross93] Jarek Rossignac and Paul Borrel. Multi-resolution 3D approximations for rendering complex scenes. In B. Falcidieno and T. Kunii, editors, *Modeling in Computer Graphics: Methods and Applications*, pages 455–465,

[Rush00] H. Rushmeier, B. Rogowitz, and C. Piatko, "Perceptual issues in substituting texture for geometry," in *Proc. SPIE Human Vision and Electronic V*, vol. 3959, pp. 372–383, 2000.

[Scar92] Lori L. Scarlatos and Theo Pavlidis. Optimizing triangulations by curvature equalization. In *Proc. Visualization'92*, pages 333–339. IEEE Comput. Soc. Press, 1992.

[Schr92] William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen. Decimation of triangle meshes. *Computer Graphics (SIGGRAPH '92 Proc.)*, 26(2):65–70, July 1992.

[Souc92] Marc Soucy and Denis Laurendeau. Multi-resolution surface modeling from multiple range views. In *Conf. On Computer Vision and Pattern Recognition (CVPR '92)*, pages 348–353, June 1992.

[Sun02] Sun ® Corporation 2002, Java 3D API homepage. http://Java.sun.com/products/Java-media/3D/.

[Teo94] Patrick Teo and David Heeger, Perceptual Image Distortion, Human Vision, Visual Processing, and Digital Display V, *SPIE Proceedings*, Vol. 2179, pp 127-141, February 1994

[Torb96] Jay Torborg and James T. Kajiya. Talisman: Commodity realtime 3d graphics for the pc. In *Computer Graphics* (*SIGGRAPH '94 Proceedings*), pages 353-363, August 1996.

[Turk92] Greg Turk. Re-tiling polygonal surfaces. *Computer Graphics SIGGRAPH '92 Proc.*, 26(2):55–64, July 1992.

[Wats00] B. Watson, A. Friedman, and A. McGaffey, "Using naming time to evaluate quality predictors for model simplification," in *Proceedings of ACM SIG CHI Conference*, pp. 113–120, 2000.

[Wats01] Watson, B., Friedman, A. & McGaffey, A. (2001). Measuring and predicting visual fidelity. In Computer Graphics Proceedings, *Annual Conference Series, ACM SIGGRAPH*, 213-220. New York, NY

[West95] S.J.P. Westen, R.L. Lagendijk, J. Biemond, Perceptual Image Quality Based On A Multiple Channel HVS Model. Proceedings ICASSP-95 (*IEEE International Conference on Acoustics, Speech and Signal Processing*) 1995

[Wink99] Winkler S., "A Perceptual Distortion Metric for Digital Color Video," in *Proceedings of SPIE Human Vision and Electronic Imaging*, San Jose, CA, Jan. 23-29, 1999, vol. 3644.