University of Alberta

# Feature Extraction and Adaptive On-line Visualization of 3D TexMesh Using Scale-space Analysis and Perceptual Evaluation

by

Irene Cheng ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of

the requirements for the degree of Doctor of Philosophy

Department of Computing Science

Edmonton, Alberta
Fall 2005

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

0-494-08622-X

*Your file Votre référence*
*ISBN:*
*Our file Notre reterence*
*ISBN:*

*In memory of my*

*parents and my sister-in-law*

# Abstract

Efficient on-line visualization of 3D textured models is essential for a variety of applications including not only video games and e-commerce, but also virtual museums and Tele-health. To visualize 3D objects online, it is necessary to quickly adapt both mesh and texture to the available computational or network resources. Early research has shown that, after reaching a minimum required mesh density, high-resolution texture has more impact on human perception than a denser mesh. Given limited bandwidth, important issues include how to extract features that best represent the original object, and how to allocate resources between mesh and texture data to achieve optimal perceptual quality. In this thesis, I propose a textured mesh (TexMesh) framework, which applies scale-space filtering (SSF) and perceptual evaluation to extract 3D features for textured mesh simplification and transmission. The appropriate level-of-detail (LOD) is automatically selected, based on the object size on the display device. Mesh refinement is guided by the just-noticeable-difference (JND) threshold, below which redundant mesh data are suppressed. Weber's law was applied to locate the JND, and perceptual experimental results showed that the threshold for 3D TexMesh is around 0.10, which is consistent with other psycho-visual experimental results on Weber's law in the literature. I apply the fragmentation approach on texture transmission to facilitate quality and bandwidth adaptation. Texture quality assignment is based on a visual quality prediction (VQP) model. On-line transmission can be performed efficiently using statistics gathered during preprocessing, which are stored in a priority queue and lookup tables. Quality of Service (QoS) requested by a client site is met by applying an efficient adaptive Harmonic Time Compensation Algorithm (HTCA) to ensure optimal use of the specified time and available bandwidth, while preserving satisfactory quality.

This thesis presents a new approach integrating feature extraction, mesh simplification, texture reduction, bandwidth adaptation, and perceptual evaluation into a multi-scale visualization framework.

# Acknowledgements

The successful completion of this thesis is contributed from the support of various people. I would like to thank my supervisor **Professor Pierre Boulanger**, who provided the necessary guidance and freedom enabling me to explore my research interests and innovative ideas. Life in the last few years would have been difficult if there was not the patience, understanding and mental support from **my husband**.

I would like to express my gratitude to the **Defense Committee** who provided valuable comments. The professional advice from **Professor Walter Bischof** on perceptual issues enabled me to perform an indepth analysis into Perceptually Adaptive 3D Visualization which is one of the unique contributions of this thesis.

The thesis work is supported by **The National Science and Engineering Research Council (NSERC) of Canada Scholarship, Alberta Informatics Circle of Research Excellence (iCore) Scholarship, University of Alberta (UofA) Walter H. Johns Graduate Fellowship Award, UofA J. Gordin Kaplan Graduate Student Award** and **UofA Graduate Student Association Professional Development Grant**.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

2AFC – Two Alternatives Forced Choice

HTCA – Harmonic Time Compensation Algorithm

HVS – Human Visual System

JND – Just Noticeable Difference

LOD – Level-of-Detail

LoG – Laplacian of Gaussian

MSE – Mean Square Error

QoE – Quality of Experience

QoS – Quality of Service

SSF – Scale Space Filtering

TexMesh – Textured Mesh

VQP – Visual Quality Predication

# Special terms used in this thesis

Viewing distance – The distance between a 3D object and the viewing platform in the virtual world.

Physical distance – The distance between the display device and the retina of the user.

# Chapter 1

# Introduction

Efficient on-line visualization of 3D objects is essential in multimedia applications such as video games and virtual museums. In the past, because of the constraints on rendering hardware and on scanning techniques, only a limited number of polygons could be used to represent 3D geometry. With advances in rendering and scanning technologies, 3D objects in the virtual world can now be represented using dense meshes, and with high resolution textures corresponding to more than hundreds of million of pixels. One of the goals of this thesis is to explore the perception limits of the Human Visual System (HVS) in the virtual world. It is important for an application to represent realism in the virtual world, but without creating redundant data which waste computational and network resources. Virtual reality projects, such as the Tele-immersion project of the University of Alberta Man-Machine Interface Laboratory in Canada [UA05] and similar real-time projects in two other labs in USA [NU05] and Germany [US05], have been set up to explore these issues. The increased amount of 3D range data imposes challenges to researchers, especially for on-line transmission. Although network speed has been improved significantly since the emergence of fiber optic technology, it is still not fast enough to transmit data with sufficiently low latency to satisfy the HVS. Problems on the Internet, such as fluctuating bandwidth, and bottlenecks in times of high traffic, create additional problems in many applications. To solve these problems and to provide reasonable support for on-line applications (such as Tele-learning, Tele-health and electronic games), the transmitted data has to be adaptively adjusted, based on an estimate of the current bandwidth and display resolution, with the additional constraint of preserving best-achievable visual quality.

An adaptive approach is possible since the transmission of the complete set of data over the network is not always necessary. In a heterogeneous multi-client environment, the amount of required data for individual clients can be dictated by the resolution of the display device, human perception, or the Quality of Service (QoS) requested by the viewer. For example, when an object is displayed close to the viewer, a denser mesh is required in order to represent sufficient detail. When the object is far away, rendering the same number of polygons is a waste of resources and time. It can be seen in Fig. 1.1 that using 180, instead of 1,800 polygons is sufficient, causing no apparent degradation on visual quality.

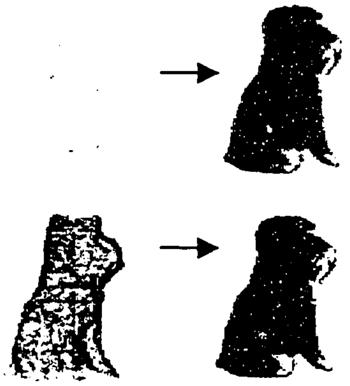


**Fig. 1.1:** *A dog object containing 180 polygons (top left) and 1800 polygons (bottom left) respectively. Texture is mapped onto the 180 and 1,800 polygon meshes for comparison (top right and bottom right).*

Efficient usage of network resources can therefore be achieved by choosing an appropriate level-of-detail (LOD), which best satisfies the available resources or user-defined constraints. An important issue related to LOD is how much detail should be removed without reducing perceptual fidelity. An application needs to have some criteria to guide the decision. A common approach is to apply an error metric, based on geometric measurements, to decide which 3D vertex should be removed from the mesh, and when the simplification operation should terminate [CVM*96]. However, the HVS is insensitive to quantitative measures that are less than a certain threshold. In other words, two visual stimuli with different geometry can appear visually identical to the HVS. A geometric metric may provide some hints about how to perform a simplification operation, but the visual quality can be assessed correctly only by using a perceptual metric, because visual fidelity is ultimately determined by human viewers. Although LOD was introduced as early as 1976 [Cla76], to our knowledge there is no real perceptual analysis relating multiple scales to viewing distance. For the purpose of this thesis, viewing distance (VD) refers to the object distance from the viewpoint in the virtual world. Physical distance (PD) will be used when referring to the distance between the display device and the retina. A scale versus viewing distance relationship is important in order to support an automatic selection of LOD based on human perception.

The main contribution of this thesis is the introduction of an integrated textured mesh (TexMesh) approach which brings together 3D feature extraction, mesh simplification, texture reduction, scale selection based on viewing distance and perceptual evaluation, and bandwidth adaptation, into an efficient online visualization framework. The TexMesh model makes use of scale-space filtering, supporting both global and local simplification, associated with photo-realistic texture mapping, which is often omitted by other simplification techniques. Furthermore, an automatic approach to scale selection is discussed, based on the concept of Just Noticeable Difference (JND), which is more efficient than the screen pixel constraint approach discussed in the literature. The next section will highlight the difference between the proposed TexMesh model and other techniques.

## 1.1 Previous LOD Approaches

When displaying 3D objects, the factors affecting visualization include geometric representation, texture, illumination, reflectance properties and human perception, among others. A large number of modeling and simplification techniques focusing on specific application requirements and constraints have been proposed. Cartography was one of the early applications using model simplification. When 2-dimensional maps of large scale are produced, "*Line generalization*" is generally used to represent rivers, roads, and other features. The terrain model and height fields [CPS97] [FMP00] are useful for freight simulation applications which require an aerial view of the scene. In the last decade, many mesh simplification techniques have been proposed for more general 3-dimensional models in the last decade. Detailed surveys on mesh simplification can be found in [HG97] [Red97] [CMS98] [Lue01] [PGK02]. Simplification techniques applied to parametric surfaces can be classified as follows:

- Regular Grids Methods [HG97] – This method is simple to implement. Regular range data are obtained directly from a range sensor. A simplified version can easily be generated by keeping one in every $k$ scan-points in both the $x$ and $y$ directions. The sequence of simplified versions forms a pyramid hierarchy providing different levels-of-detail (LOD). However, the simplified version may miss critical points or important 3D features of the object, thereby degrading visual fidelity.

- Hierarchical Methods – In contrast to the non-adaptive property of regular grids, this method provides the adaptive counterpart of the pyramid structure. Regions are subdivided recursively, forming a tree-like hierarchy, such as R-Simp and BSP-Tree [BW00] [SG01]. Quadtree, k-d tree, or other divide and conquer strategies can also be used [HG97]. Each branch node can have only one parent node. The hierarchical approach is different from other general triangulation methods, such as Delaunay triangulation [HG97], where each triangle can belong to more than one parent. Hierarchical subdivision methods are fast, simple and facilitate multi-resolution modeling; however the tradeoff is a poorer visual quality than Delaunay triangulation. The latter ensures that triangles have good aspect ratios avoiding sliver (long and thin) triangles.

- Features Based Methods – This approach performs triangulation based on a set of features or critical points. Edges between feature points are known as break lines. 2x2 and 3x3 linear or nonlinear filters are commonly used as feature detectors. Very often a weeding process is required to remove features that are too close together. An interesting research work from Southard [Sou91] uses Laplacian computation to compare curvatures and rank feature points, however this approach applies only to planar surface models and tends to distribute points uniformly, causing redundancy on low curvature surfaces and insufficient geometric data on high curvature surfaces. Some other feature detection techniques found in the literature are designed for surface reconstruction, and not for model simplification. For example, an extended marching cube algorithm is used to detect feature points for surface reconstruction [KBS*01], and a neighborhood graph is used to extract feature lines from surface point clouds, in order to reconstruct a model from noisy data or under-sampled data [GWM01].

- Triangle Refinement Methods – An early refinement technique can be traced back to Douglas' algorithm on 2D curve simplification [DP73]. Many 3D refinement methods start with a minimal approximation on a set of selected points and apply multiple passes. In each pass, the set is split and the region is re-triangulated until the final high-resolution triangulation is reached. Fowler [FL79] applied a hill-climbing technique in order to locate a candidate point to insert into the triangulation. A test point is initialized at the center of the triangle and repeatedly steps up in the neighborhood until a local maximum is reached. However, this approach may fail to find the global maximum within the triangle. Schmitt [SC91] used a two-stage split-and-merge process. The split process is similar to other vertex insertion algorithms. The merging process joins adjacent regions with similar face normals.

- Triangle Decimation Methods – As opposed to refinement, the idea of decimation methods is to start with all the range data or scan-points, and recursively remove vertices from the triangulated mesh [KCH*91] [HH93]. Scarlatos [SP92] suggested an algorithm involving an initial triangulation and three phases: shrinking triangles with high curvature, merging adjacent coplanar triangles, and swapping edges to improve shape.

- Optimal Methods – In general, optimal methods are less common than heuristic methods because they are slower and more expensive to implement. A minimum error or tolerance is defined based on some criteria. The best-effort simplification is obtained within the tolerance limit [BG94] [DG95]. This approach, however, is not efficient for online interactive applications.

- Perceptually Driven Methods – As opposed to the above techniques, which are based on a geometric metric, perceptually driven simplification methods are guided by human perception and visual quality preservation [COM98] [LT00]. Vertices are removed only if they are imperceptible and do not degrade the visual quality. Most perceptually driven techniques in the literature are designed for view-dependent visualization [LH01] [Red01] [WLC*03].

The idea of applying LOD was discussed as early as 1976 by Clark [Cla76]. LOD can be static, dynamic or view-dependent. The traditional static approach preprocesses a number of discrete versions at different resolutions for each 3D object, corresponding to different levels of detail. At runtime, the appropriate version is chosen and rendered. This approach is simple, and requires minimum runtime computation; however, only a limited number of levels are available. Instead of creating discrete versions, a dynamic approach uses a data structure to encode a continuous spectrum of detail (CLOD); the desired level can be extracted at runtime. This approach provides better granularity and uses the optimal number of polygons, however, the tradeoff is additional computation before rendering. The view-dependent approach is an extension of the dynamic approach. Criteria are defined to select the most appropriate level for the current view. Each approach has its advantages and disadvantages, depending on the requirements of the application. A comparison of some frequently referenced simplification techniques in the literature is given in Table 1.1.

| **Simplification techniques** | *To* | *Vi* | *In* | *Dr* | *Gl* | *Pe* | *Co* |
|---|---|---|---|---|---|---|---|
| Decimation of triangle meshes (1992) | yes | yes | yes | no | no | no | no |
| Geometric optimization (1993) | yes | yes | yes | yes | no | no | yes |
| Voxel based object simplification (1995) | no | yes | no | yes | no | no | yes |
| Simplification envelopes (1996) | yes | yes | yes | no | no | no | no |
| Progressive meshes (1996) | no | yes | no | yes | yes | no | yes |
| Simplification using quadric error metrics (1997) | no | yes | no | yes | yes | no | yes |
| Model simplification using vertex clustering (1997) | no | no | no | no | yes | no | yes |
| Adaptive real-time level-of-detail-based rendering for polygonal models (1997) | no | no | no | yes | no | yes | yes |
| Appearance preserving simplification (1998) | yes | yes/ no | no | no | no | yes | yes/ no |
| Image driven simplification (2000) | no | yes | no | yes | yes | yes | no |

**Table 1.1:** *A comparison of some commonly referenced simplification techniques in the literature:*

*To – Topology preserved.*

*Vi – View-independent.*

*In – Incremental; vertices in a coarse version is a subset of those in a finer version.*

*Dr – Drastic simplification allowed.*

*Gl – Global and local surface properties are both considered.*

*Pe – Perceptual quality is taken into consideration.*

*Co – Continuous level-of-detail is supported.*

*The appearance preserving simplification technique provides a framework which supports both view-dependent and view-independent visualization, and supports LOD and CLOD.*

## 1.2 Motivation

Despite the excellent research on simplification techniques in the last decade, what is missing is an integrated simplification approach associating texture reduction with mesh simplification at multiple scales, taking human perception and network adaptation into account. The simplification techniques in the literature are inadequate in one or more of the following ways:

- Capability of drastic simplification – Topology preservation [EDD*95] is able to maintain high fidelity, but is not efficient when considering LOD. When viewing an object at a far distance, the HVS is insensitive to small genera embedded in the object and therefore, rendering more triangles in order to preserve these holes is not an effective way to achieve simplification. The error minimization techniques are useful in reducing over-sampled data, or data packed too close together; however, an error metric [SZL92] can prevent drastic simplification when generating coarse objects displayed far away from the viewpoint. The simplification envelopes algorithm [CVM*96] is based on the constraint that the simplified surface is within ±ε (a user defined tolerance) of the original surface. Avoidance of self-intersection of triangles in the neighborhood prevents drastic simplification.

- Local and global consideration – Simplification can be based on ranking the importance of vertices, or feature points. For example, vertices attached to large triangles, and vertices of high curvature are considered more important in geometric representation [RB93] [LT97] [Lin00]. The quadric error approach uses a 4x4 matrix to compute the sum of the squared distances from a vertex to the planes of neighboring triangles [GH97]. Limited filter window size considers the characteristics of local surfaces only; it does not consider the global structure – a factor that is taken into account by the scale-space filtering (SSF) technique used in the TexMesh model. Mesh simplification based only on local geometry may remove important vertices at an early stage. An example can be seen when applying the voxel-based simplification algorithm [HHK*95] on models with high-frequency details, such as sharp edges and squared-off corners. Ignoring global surface property may result in removing large noticeable structures from the 3D surface, instead of removing less important small details at an early stage of simplification. For example, techniques merging nearly coplanar neighboring triangles will simplify a surface from (a) to (b) (Fig. 1.2), because the faces adjacent to vertex *V* represent a more coplanar surface than the faces forming the smaller local structures. However, simplifying from (a) to (c) is more consistent with the way in which the human visual system perceives a change of the structure from close to far.

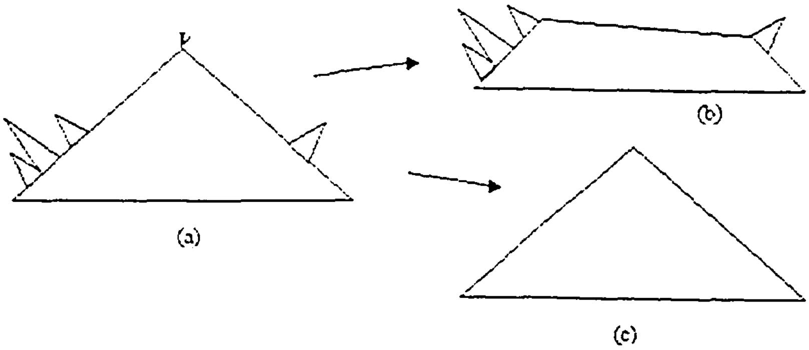


**Fig. 1.2:** *An example of a simplification technique considering only local surface curvature, i.e., (a) to (b), and that taking into account of global structures i.e. (a) to (c).*

Simplification techniques taking both local and global 3D structures into account can thus improve visual fidelity.

- Efficient online performance – Error minimization or optimization techniques are often slow and expensive to implement [Nad86] [BG94]. An efficient approach is to collect sufficient statistics during preprocessing to achieve fast online retrieval, preferably in constant time. In other words, when refining a model from a coarse level to a more detailed one, only additional vertices need to be transmitted. This incremental approach requires the coarse set to be a subset of the refined set. Many simplification techniques involve relocation of vertices and thus online transmission cannot be incremental [BW00] [GH97] [SG01] [Tur92]. In the progressive meshes method, although the original mesh can be recovered exactly, after all data are received, the edge collapse transformation creates new vertices and the *vsplit* record stream increases network workload [Hop96]. The adaptive real-time LOD technique also involves vertex relocation [XEV97]. Some techniques are view-dependent and, although they may provide more precise visual quality for each view, their main disadvantage is the increased runtime load for selecting an appropriate LOD [LE97]. The image-driven simplification technique [LT00] compares the image after an edge collapse operation, with the original image. Mean square error (MSE) is used to compute the deviation from the original image. Images are taken from multiple views in order to evaluate the quality of the entire object, and the next collapsed edge is the one generating the least deviation. However, rendering the entire model for every edge in every viewpoint for different scales is expensive, even with hardware-accelerated rendering. Computational cost is a major burden for view-dependent approaches.

- Automatic selection of scale based on viewing distance – While LOD was discussed extensively in the literature, there has been no analysis of how a scale is selected based on viewing distance. For example, there have been no experiments showing whether scale relates to viewing distance linearly or exponentially, or by any other function. When selecting LOD, a smaller and coarser object is displayed at a greater distance, in contrast to a bigger and finer object close to the viewpoint. A scale-distance function is therefore introduced in the TexMesh model, and viewing distance is defined as the location of the 3D object relative to the viewing platform in the virtual world. Viewing distance is different from the physical distance between the retina and the display device, as defined in [LH01]. Cohen et al. mentioned that the choice of LOD depends on the object's area in screen pixels, which is related to its distance from the viewpoint [COM98], but no discussion was presented in the paper to explain how LOD can be automatically selected based on viewing distance.

- Association of texture reduction with the mesh simplification analysis using photo-realistic texture mapping – Perception of depth and realistic texture are the main factors required to achieve realism and visual fidelity in the virtual world. In recent years, researchers began to incorporate color and texture into their mesh simplification models. When texture is mentioned in the literature, it often refers to texture that can be described procedurally or animated texture [Tur91]. Synthetic texture or per pixel color stored in each vertex [COM98] [GH98] [SGR96] [SSG*01] can be estimated or interpolated. For example, when walking through an animated scene, the next frame can be predicted based on available neighboring data [CMF99]. Using interpolated or animated texture is a compromise in applications which require fast interactive rendering. For applications requiring real life texture, interpolating colors or estimating patterns between vertices is not acceptable. The non-interpolated texture used in the TexMesh model has resolution much higher than the mesh. It was observed in perceptual experiments that the HVS is more sensitive to higher texture resolution after the mesh reaches an optimal density [PCB05]. Increased texture resolution also improves visual fidelity for a given geometry in general. Since high-resolution texture data is large, compared to mesh data, texture reduction can speed up 3D visualization. The TexMesh integrated approach simplifies non-interpolated photo-realistic textured mesh based on 3D feature points extracted using scale-space analysis. Photo-realistic texture maps are used in [YFM00], but their focus is on recovering geometry from texture patches retrieved from multiple photographs, and not on generating LOD. A distance-based technique is applied to photo-textured terrain in [LKH*95]; however, color interpolation between pixels is necessary in their technique to avoid a blocky appearance of terrain texture.

- Simplification based on objective perceptual evaluation – Often, the number of vertices or faces, or an error metric, is used as a criterion to measure efficiency [PGK02] when comparing simplification techniques in the literature. In fact, a geometric deviation may not cause any degradation of perception, because the HVS is insensitive to minute details below a certain limit. Experiments described later in this thesis show that perceptual quality is a more reliable and efficient measure when human perception is involved. A perceptually driven simplification technique was introduced in [LH01], but their method is used for the rendering of a view-dependent image, while the TexMesh model is applied to a view-independent 3D object. Luebke et al. [LH01] use Gouraud-shaded meshes, while photorealistic texture mapping is applied on a TexMesh object. Furthermore, their simplified models still contain redundant data, indicated by the authors' statement that their models could be reduced two to three times further in polygon count, without perceptible effect. An efficient simplification technique should avoid generating redundant data since they do not improve visual quality. Watson [WFM01] compared the naming times, and rating and preference techniques, but selected only one set of views for each object, and did not apply a full 360° interactive comparison. Naming times tend to be affected by an individual's prior knowledge of the stimuli, *i.e.*, different LOD of an object can be recognized and named within the same response time period due to some prominent features on the 3D surface.

- Adaptation to bandwidth fluctuation – The joint geometry/texture progressive coding method applies wavelet transform to encode the mesh and texture data for transmission [OC00], however, the method is not adaptive to fluctuating bandwidth. In order to support online applications, geometry/texture data must be delivered within a reasonable or user-defined time. A dynamic approach by adjusting geometry/texture resolution based on current bandwidth (as discussed in the TexMesh model) offers greater efficiency. Furthermore, a perceptual metric is used in the TexMesh to determine when mesh refinement should be terminated, thereby allocating the remaining bandwidth to texture resolution in order to enhance the overall visual fidelity.

In recent years, perceptually adaptive graphics [OHM*04] has received increasing attention in the graphics and visualization community. A state-of-the-art report was presented on visual perception in EUROGRAPHICS 2000 [Mcn00]. Then, a group of researchers from computer graphics, psychology, and other disciplines gathered in 2001 – as a result of a joint effort between EUROGRAPHICS and SIGGRAPH – to discuss the importance of incorporating human perception when striving for realism in the virtual world [LH01] [Red01] [WFM01] [WLC*03]. Considerable effort has been expended on verifying geometric error estimation with perceptual evaluation experiments in order to achieve higher visual fidelity of 3D display. Most perceptually-driven techniques developed so far focus on view-dependent rendering. These techniques can be applied to dynamic scenes [Red01] [ODG*03], and can be used to compute the relative resolutions between the region of interest and the periphery [Red01] [BDD*03]. In order to achieve visual quality, user-guided simplifications were also suggested in [PS03] [KG03]. In contrast, the TexMesh approach introduced in this thesis is view-independent, applied to relatively static 3D objects, and does not require user intervention when predicting visual quality.

*Perceptual analysis is a fundamental component in the TexMesh framework.* Luebke [Lue01] suggested that the field of polygonal simplification has reached maturity. The simplification algorithms developed to date provide a good foundation upon which other research, such as perceptual evaluation of 3D quality, can be built. As 3D scenes become commonplace in online applications (such as those on the Internet, with limited and varying bandwidth) good performance and efficient adaptation is an important goal. The objective of this thesis is to integrate related research issues from different directions, and correct the inadequacies of previous approaches. While visual fidelity has traditionally been measured based on geometric metrics, the TexMesh model is perceptually driven, and is adaptive to fluctuating bandwidth (Fig. 1.3).

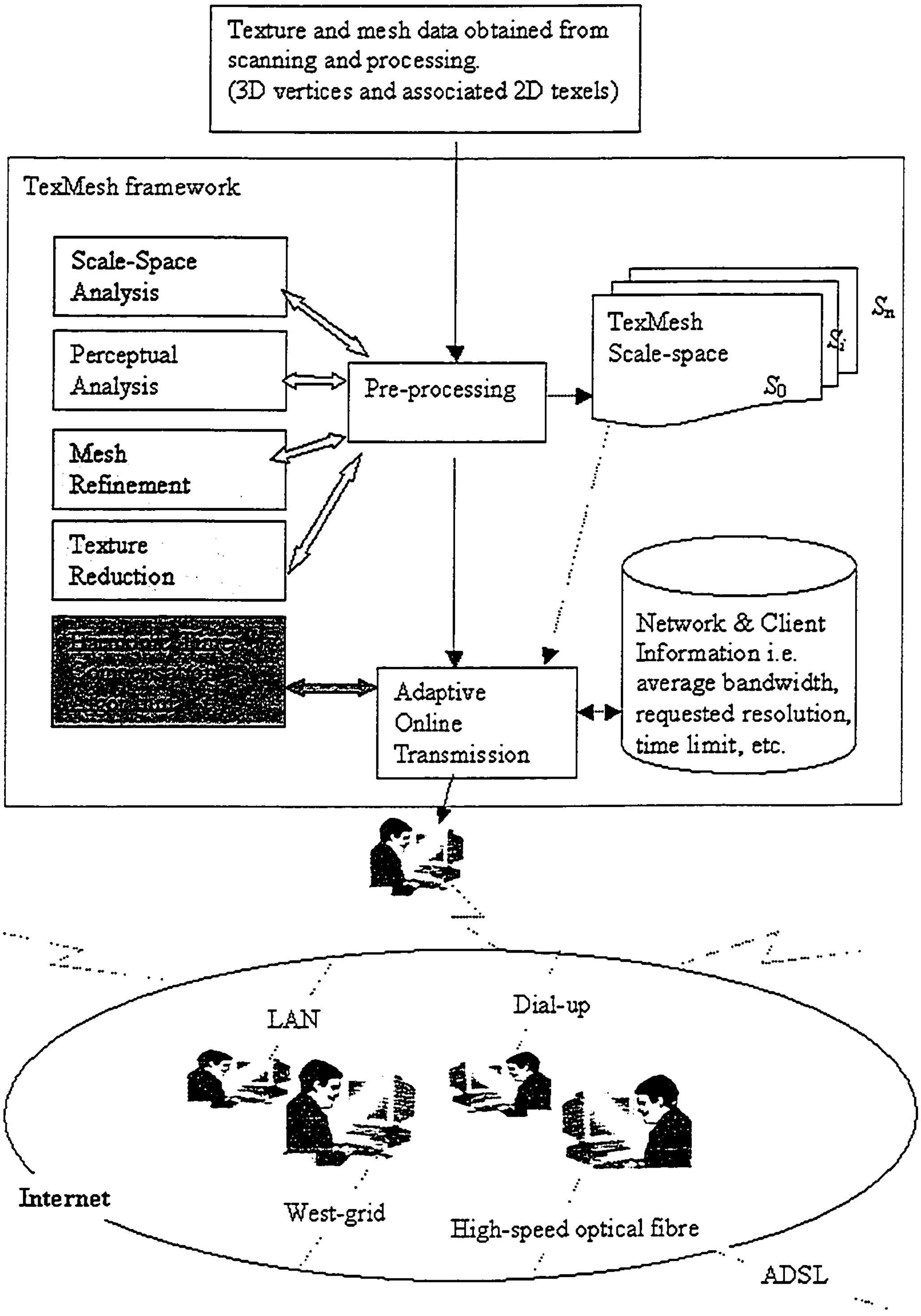

Texture and mesh data obtained from scanning and processing. (3D vertices and associated 2D texels)
TexMesh framework
Scale-Space Analysis
Perceptual Analysis
Mesh Refinement
Texture Reduction
Pre-processing
TexMesh Scale-space
$S_0$
$S_i$
$S_n$
Adaptive Online Transmission
Network & Client Information i.e. average bandwidth, requested resolution, time limit, etc.
LAN
Dial-up
Internet
West-grid
High-speed optical fibre
ADSL

**Fig. 1.3:** *A block diagram of the TexMesh framework and environment.*

## 1.3 Thesis Contributions

The contributions of this thesis include:

- Generating a level-of-detail framework based on feature points extracted using scale-space analysis and human perception.
- Developing joint texture and mesh simplification based on feature point distribution.
- Developing an adaptive approach for transmission based on the harmonic time compensation algorithm, which minimizes the estimation error caused by bandwidth fluctuation.
- Relating scales to viewing distances by a step function, representing the surface properties of a 3D object.
- Performing mesh refinement using perceptual estimation based on just-noticeable-difference.
- Supporting automatic selection of mesh scale based on just-noticeable-difference and object size on screen.
- Introducing a new method to predict texture visual quality based on 3D as well as 2D properties.

## 1.4 Thesis Organization

The remainder of the thesis is organized as follows: Chapter 2 discusses scale-space filtering and how it is used to generate continuous level-of-detail of 3D objects. Chapter 3 introduces the TexMesh model and explains how joint texture and mesh simplification is performed. Chapter 4 proposes the Harmonic Time Compensation Algorithm to achieve efficient online transmission. Chapter 5 discusses the main perception issues relating to a 3D TexMesh model. Chapter 6 relates perception of scale with viewing distance. Chapter 7 introduces a perceptual metric for mesh refinement based on JND. Chapter 8 proposes a visual quality prediction model for 3D TexMesh. Chapter 9 integrates feature extraction, texture reduction, mesh simplification, bandwidth adaptation, and perceptual evaluation into the TexMesh framework and, finally, Chapter 10 offers conclusions and discusses future research directions.

# Chapter 2

## Scale-Space Filtering and LOD

The Gaussian filter is an efficient smoothing operation used in computer vision. Based on the Gaussian's kernel, Witkin [Wit83] and Koenderink [Koe84] formally defined the scale-space concept in 1983-84, bringing together an unrelated set of derived images at different levels of detail along the scale parameter and allowing image data to be integrated into a continuous perspective. Since then, the Gaussian kernel has often been studied in conjunction with the so called "multiple scale approach." However, scale-space filtering had mainly been applied to 2D images [KF01] [BP02] [Lindberg95] [Lindberg98], and only recently has this technique been used in computer graphics, with limited applications in 3D visualization. An intrinsic filter, a generalization of scale space filtering, is used to eliminate noise from image and scanned data [BJB02]. Relatively fewer researchers have looked into 3D model (or mesh) simplification based on feature point analysis at multiple scales. Southard applied Laplacian to rank uniformly distributed feature points on planar surfaces [Sou91], and Gaussian filter is used to detect features at multiple scales [PKG03], but their analysis is not extended to mesh integrated with real texture and adaptive on-line transmission.

## 2.1 LOD and Scale-Space

In this thesis a new scale-space filtering (SSF) algorithm is proposed to extract 3D features based on which adaptive on-line compression and transmission of 3D models can be performed. Traversal between the different scales, or levels, is achieved by varying the standard deviation parameter $\sigma$; the higher the value the more the mesh is smoothed. SSF is based on locating the zero-crossings of a signal at multiple scales. Zero-crossings can be used to detect the degree of persistence of a structure (feature) in a 3D model. Minor structures tend to diminish as $\sigma$ increases, and only major structures persist at higher scales. In other words, minor features will be removed before major features in the simplification process. In contrast, major features are inserted before minor features in a refining process. The advantage of using SSF is its ability to smooth locally or globally, moderately or drastically, depending on the filter window size and the value of $\sigma$. When using a small window size, SSF eliminates signal noise in the local region. By using a bigger window size, the filtering or averaging effect covers a larger surface. Fig. 2.1 is an example of global smoothing using a window size of 201 on a signal of 256 sample values. To achieve global smoothing, the window size has to be at least twice the standard deviation computed from the sample space (covering at least 97.7% of the sample data in a normal distribution). If a smaller window size is used, smoothing will be restricted and terminated before reaching the bottom scale in Fig. 2.1.

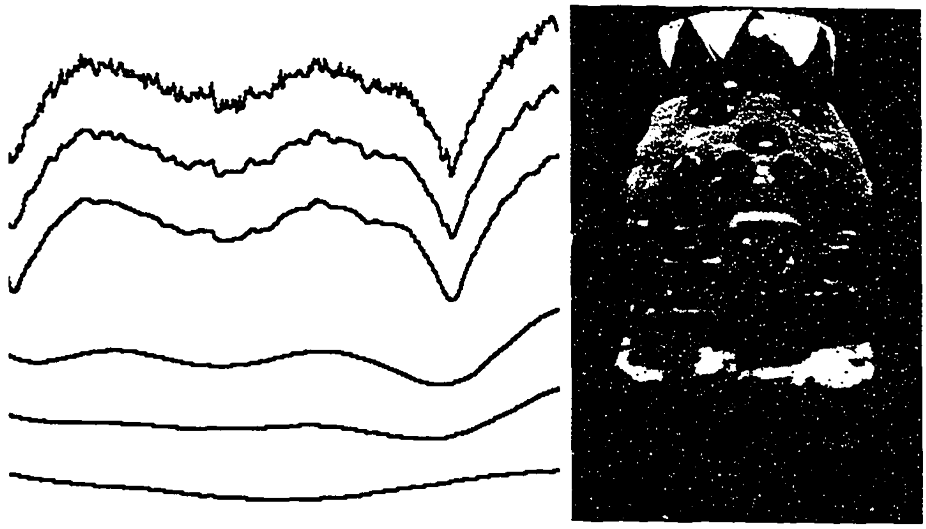


**Fig. 2.1:** *Scale $S_i$ increases from top to bottom. $S_0$ is the original signal extracted near the bottom of the Nutcracker model. Note that the local variation (fine detail) in the original signal is gradually removed and the scaled signal becomes smoother and flatter.*

Theoretically, 100% global smoothing will end up with a monotonous surface losing all the surface features. However, the human visual system (HVS) is insensitive to details beyond a certain level. Thus, for a perceivable object, the filter window can be smaller than twice the standard deviation in order to save computation time. In the experiments, I applied a window size of 1.4 times the standard deviation $\sigma$ (normally it should be $3\sigma$), and found that this window size provides sufficient simplification for objects placed at a distance close to infinity in the virtual world. Further simplification beyond this point by using a bigger window is not necessary. How scale is related to viewing distance and the perceptual impact of changing scale will be discussed in Chapter 6.

The zero-crossings at different scales can be computed by applying the second derivative of the Gaussian (called Laplacian-of-Gaussian or *LoG*). Eighteen feature points are identified in the original signal (Fig. 2.2, right). By increasing the $\sigma$ value, the number of feature points decreases from 18 to 2 as reflected by the increasing smoothness of the scaled values from the top to the bottom scales (Fig. 2.2, left).

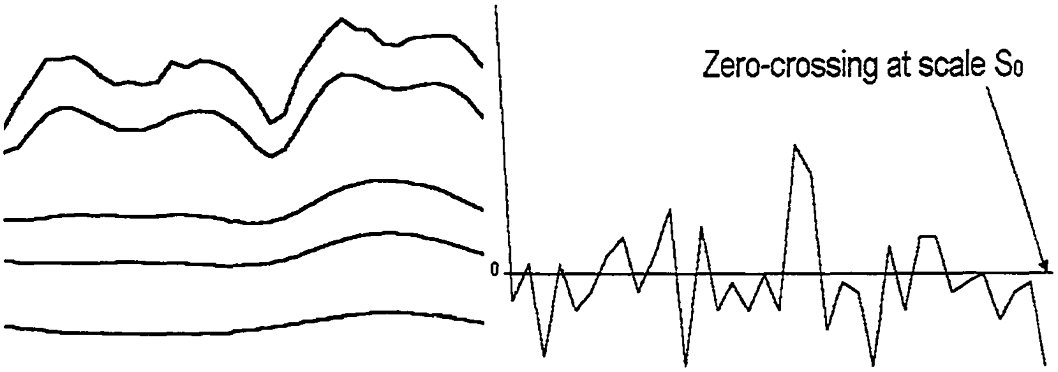


**Fig. 2.2:** *(Left) The top is the original signal with 18 zero crossings, generated by signals extracted from the Nutcracker model. The other four smoothed scales have 8, 6, 4, and 2 zero-crossings respectively from top to bottom. (Right) 18 zero crossings detected in the original signal* $S_0$.

The SSF operation in 2D can be summarized by the following equations:

$$w_G(x,y) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x^2+y^2)}{(2\sigma^2)}} & (x,y) \in W \\ 0 & elsewhere \end{cases} \tag{2.1}$$

$$w_{LoG}(x,y) = \begin{cases} -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} & (x,y) \in W \\ 0 & elsewhere \end{cases} \tag{2.2}$$

$$f * S(x,y) = \int_{-t}^{t} \int_{-t}^{t} f(x+u, y+v) w(u,v) du dv \tag{2.3}$$

Where $w_G(x,y)$ represents the weight at pixel $(x,y)$, $f$ represents the original signal (image) and $f*S$ the smoothed image and the weights are assumed to be defined in a square window $W$ of length $2t+1$. In the discrete case, *e.g.*, with a real image, summation is used instead of integrals, and the Gaussian weights are normalized so that the sum of all the weights equals 1.

## 2.2 Spherical Scanned Data

Modern laser scanners detect depths and generate 3D vertices in the form of point clouds. Fig. 2.3 (left) shows a 6-inch dog object. The generated point cloud (Fig. 2.3 middle) is then triangulated, and mapped with the scanned texture (Fig. 2.3 right) to generate a texture mapped 3D object (Fig. 1.1).

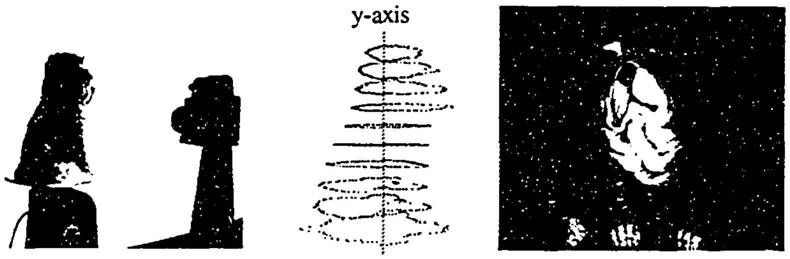


**Fig. 2.3:** *(left) Zoomage® 3D scanner and a dog object, (middle) sample of 3D scan-points, and (right) scanned texture image.*

The SSF of a 3D model is achieved as follows: First note that the data acquired (Fig. 2.3 middle) can be represented as $r_x(\alpha,y)$; where $\alpha$ is the angle on a horizontal plane around the $y$-axis of rotation of an object, $y$ is the vertical coordinate, and $r_x$ denotes the perpendicular distance from the $y$-axis to the surface of an object for a given $(\alpha,y)$ pair. SSF for a 3D model is thus similar to a 2D image, for the simplified mesh representation considered here, with $f(x,y)$ replaced by $r_x(\alpha,y)$. Also, the appropriate scaling factors along the horizontal and vertical directions can be significantly different, depending on the variance of the sample points for a given region. Thus, Equations (2.1) and (2.3) need to be modified to (2.4) and (2.5):

$$w_G(\alpha,y)=\begin{cases}\dfrac{1}{\sigma\sqrt{2\pi}}e^{-\dfrac{(\phi\alpha^2+\psi y^2)}{2\sigma^2}} & (\alpha,y)\in W\\ 0 & elsewhere\end{cases} \qquad (2.4)$$

$$Rx^{*}S(\alpha,y) = \int_{-t}^{t}\int_{-t}^{t} Rx(\alpha+u, y+v)w(u,v)dudv \qquad (2.5)$$

For uniform sample points, $\phi$ and $\psi$ equal 1, but for irregular sampling, $\phi$ and $\psi$ are used to accommodate the variable inter-sample distance along different axes. Note that in the actual implementation, we use two passes of 1-D filters, since all the filters discussed above are orthogonal. The vertices are first smoothed along the $x$-axis and then the resulting values are filtered once more along the $y$-axis using the filtered values. Fig. 2.4 shows the face features, of a head model, changing towards a smoother spherical surface when going from low to high scales (left to right). The original mesh contains 1,872 vertices and 3,672 faces. The other five meshes are of increasing scales as described in Table 2.1:

| Scale $S_i$ | # of vertices removed | # of faces in mesh |
|---|---|---|
| 5 | 685 | 2226 |
| 7 | 744 | 2108 |
| 10 | 824 | 1948 |
| 15 | 985 | 1624 |
| 20 | 1196 | 1190 |

**Table 2.1:** *(Head model) Number of faces remaining at each scale $S_i$, after removing low priority vertices.*

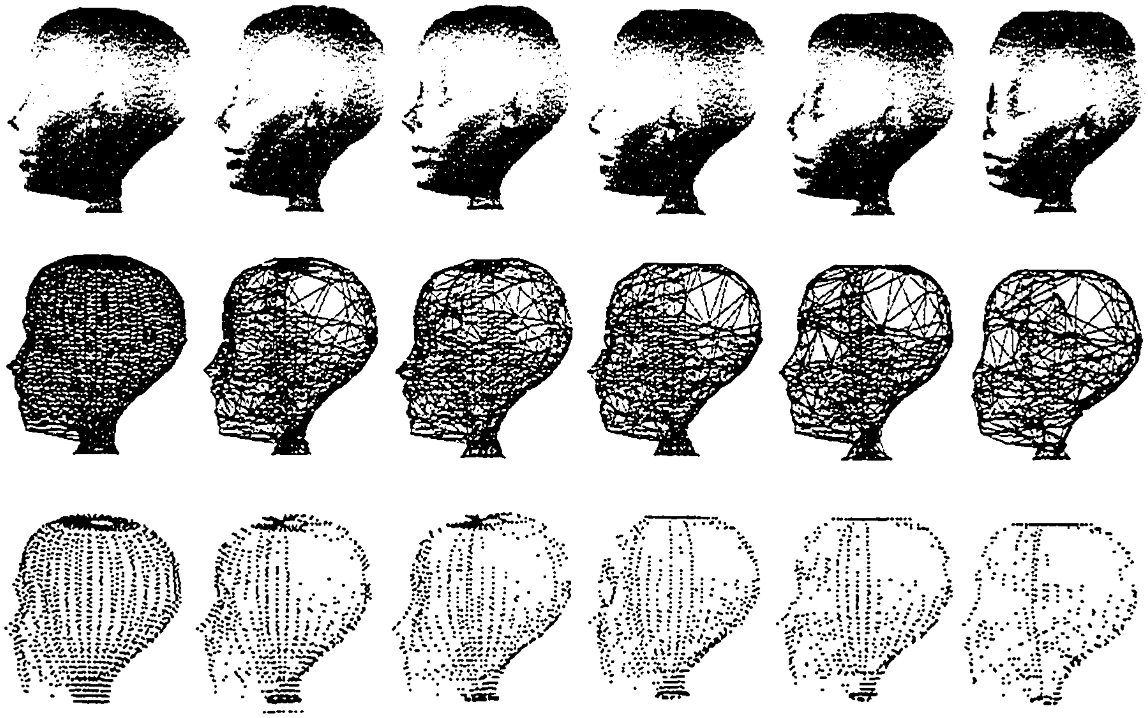


**Fig. 2.4:** *Increasing scales from left to right of a head model: (Top) texture mapped 3D mesh, (Middle) 3D mesh and (Bottom) feature points extracted at scale $S_i$.*

Note that for this head model, the surface structures of the ears and the cheeks are more prominent than that of the eyes, which are quite shallow. The back of the head is comparatively smooth. Features points are preserved longer, toward higher scales, in areas where structure are more prominent.

In the TexMesh model, statistics relating to LOD are generated during preprocessing based on feature point extraction at different scales.

## 2.3 Conclusion

In Chapter 2, we have reviewed the scale-space theory introduced by Witkin in 1983. While scale-space analysis has been applied in image processing and computer vision extensively, its applications on 3D visualization are not fully explored. This chapter explains how SSF is used to analyze 3D surface structures and how feature points at different scales are extracted based on the detection of zero-crossing. In Chapter 3, we will discuss how SSF is integrated into the TexMesh framework.

# Chapter 3

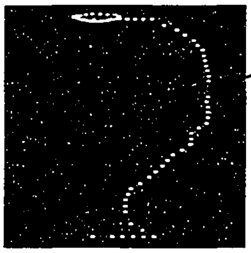
## The TexMesh Model

Simplification techniques developed in the past focused mainly on geometry, shaded with color per vertex or mapped with synthetic material, without integrating real life texture in a coherent manner. Many studies emphasized the importance of million of triangles in order to preserve fine details, but ignored high resolution real texture which have been shown through user evaluations to have major impact on perceptual quality in 3D visualization [PCB05]. In view of the size of texture data, compression of texture is essential for on-line transmission given limited and fluctuating bandwidth. An integrated approach combining mesh simplification and texture reduction, based on feature point extraction and distribution, is proposed in this thesis.

Feature points in the TexMesh model is defined as a group of vertices, which can best represent the geometry of 3D object at a given viewing distance. In Fig. 2.4, for example, the original head model contains 1872 feature points (scale $S_0$). After removing 1196 vertices, it is represented by 676 feature points at scale $S_{20}$. At any scale $S_i$, feature points are detected by using *LoG*. Vertices creating zero crossings at scale $S_i$, are recorded as feature points and assigned the value *i*. Each feature point is represented by three components: *(i, (tx,ty) , (gx,gy,gz))*. The second and third components are the 2D texture and 3D vertex coordinates, respectively.

Vertices with stronger persistence will have higher $i$ values (higher priority during mesh refinement), generating a priority list. During simplification, features of low priority are first removed, leaving more prominent features at higher scales. Iterative vertex removals, by integrating the removed vertex with its closest neighbor, are applied to generate each simplified version. No new vertex is generated in the process. When selecting the next edge or vertex to remove, the simplification criteria used in most methods choose the candidate which will generate the minimum deviation from the previous simplified surface. For example, in progressive meshes [Hop96], the minimum energy cost in the neighborhood is affected by an edge collapse operation, and has to be recomputed. The simplified surface then affects the choice of the next collapse operation and the vertices to remove. In the TexMesh approach, the order of vertex removal follows the priority predetermined by applying scale-space filtering on the original 3D surface.

During refinement, features of higher priority are inserted first into the coarse model. Since real texture mapping is used in the TexMesh model, texture border has to be taken into account during the simplification process. Texel values are normalized in the range [0,1]. Vertices associated with border texels are removed only if such an operation does not distort the texture pattern. A collection of the texture border vertices of the head model (Fig. 2.4) generated from the spherical approach is shown in Fig. 3.1.



Border of texture pattern. Triangles overlapping this border will distort the pattern.

**Fig. 3.1:** *An illustration of border vertices taken from the head model.*

Preprocessing generates a scale map and a fragment map. The scale map records the feature points extracted at each scale, and the fragment map records the feature point distribution in each texture fragment.

## 3.1 Scale Map

Three-dimensional mesh vertices are computed from the signals captured by a 3D ranger sensor. Fig. 3.2 (right) shows an example of the scanned signal, which can be processed to compute the depth information. Putting aside the depth information ($z$-coordinate), $N$ vertices can be sorted and assigned unique identification numbers $L$, *i.e.*, $0 \leq L < N$, based on their $y$ then $x$ coordinates. Note that $x$ is derived from $r_x$ and $\alpha \in [0,360]$ in the spherical approach, and thus is ordered for a given $y$ value. A Scale Map is a 2D display of all 3D vertices in a matrix with rows and columns corresponding to the $y$ and $x$ values respectively. The default value for each vertex is zero corresponding to scale 0 (original signal). At each scale $S_i$ only feature points detected at that scale are updated with the value $i$. During preprocessing, Gaussian filters with increasing sigma (standard deviation) values $\sigma_i$ are used from scale $S_0$ to $S_{\max}$, *i.e.*, $0 \leq i \leq$ max scale, with $S_{\max}$ corresponding to the scale at infinity. Zero crossings are detected from the filtered space $G_i$ where $G_0$ represents the set of original unfiltered range data. Starting from a coarse model, feature points not already included in the mesh can be retrieved from the scale map, in decreasing $i$ values, to refine the model progressively. The scale map can be implemented as a priority queue, where the next required vertex is popped from the queue in constant time.

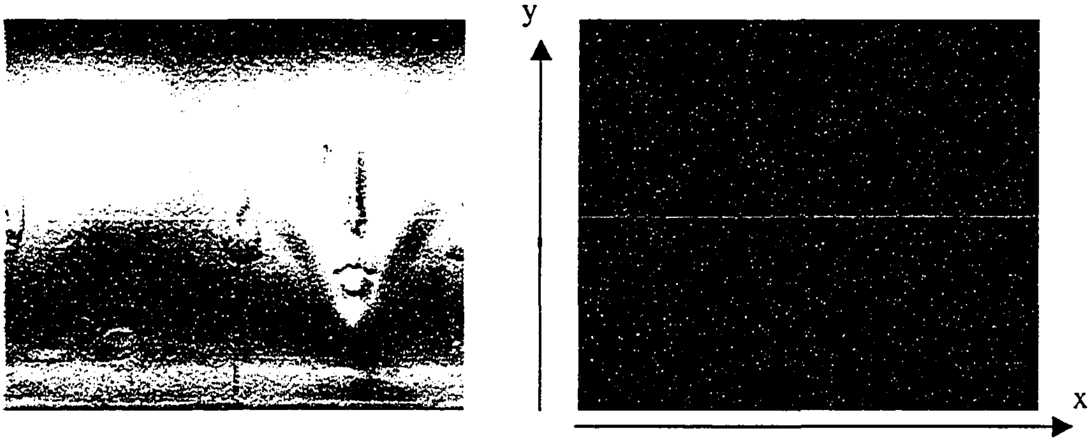


**Fig. 3.2:** *(Left) Texture pattern and (Right) laser signal, of a head model captured using the Zoomage® 3D scanner.*

## 3.2 Fragmentation Approach for Texture Transmission

One characteristics of the TexMesh model is the association of texture reduction with mesh simplification. In a 3D object, with full color real texture, there are two components that can be filtered — the mesh and the texture. The texture quality $q_i(x, y)$ of a fragment (*x*, *y*) is determined based on the associated feature point density $\eta_i(x,y)$, which has a value between zero and one, and is mapped onto a compression scale. For example, in the current implementation, A JPEG quality scale from 0% to 100% is used, where 100% means no compression. While wavelet coding applies to the entire image and is geometry-independent, my approach supports variable quality determined by the density of surface structures (geometry-driven). Note that JPEG is used for convenience and wide support on the web, and in JAVA [JPG05]; however, standards such as JPEG2000 can be used as well in the future to code fragments.

The texture image of a 3D model can be transmitted as one block or a collection of sub-blocks. The advantage of dividing into sub-blocks is to make use of distributed networks and apply variable qualities to different texture blocks as explained below. The main concern is whether the additional headers information will increase the overall volume of data to be transmitted. In this section, I will show that sub-dividing into smaller blocks of optimal dimension does not increase the overall size for high resolution texture images. Instead, the sub-block approach helps to fully utilize the available bandwidth. Feature points extracted at each scale $S_i$ are distributed onto the corresponding sub-block in a fragment map based on the 2D texture coordinates associated with each 3D feature point.

## 3.3 Fragment Map and Variable Texture Quality

The texture image of a 3D model is fragmented into $N_x$x$N_y$ pieces after determining the optimal size of a fragment. To apply JPEG compression efficiently, keeping in mind the size of macroblocks, the optimal dimension of a fragment is chosen as a multiple of 16. The entire texture is also adjusted so that there is no partial fragment. For example, a texture image with dimension 4800*1600 pixels, can be divided into 7,500 fragments of size 32x32 pixels. Similar to the scale map, fragments are arranged in a matrix with $N_y$ rows and $N_x$ columns. Since each 3D vertex is associated with a 2D texel, it is possible to distribute the vertices into the $N_x$x$N_y$ fragments.

### 3.3.1 Comparison between Fragmented and Non-Fragmented Size

We used five texture patterns (Fig. 3.3) to compare the fragmented and non-fragmented sizes for different qualities using the Intel JPEG compression library. Each fragment has a dimension of 16x16 pixels. Experimental results show that the sum of the fragments is significantly less than the size of the non-fragmented JPEG file for images of dimension greater than 256 pixels (Table 3.1). For high resolution images, it is therefore advantageous to transmit individual fragments to the client site for recombining and rendering. The fragmented approach also facilitates image fragment retrieval from multiple repositories, parallel processing and transmission over distributed networks.

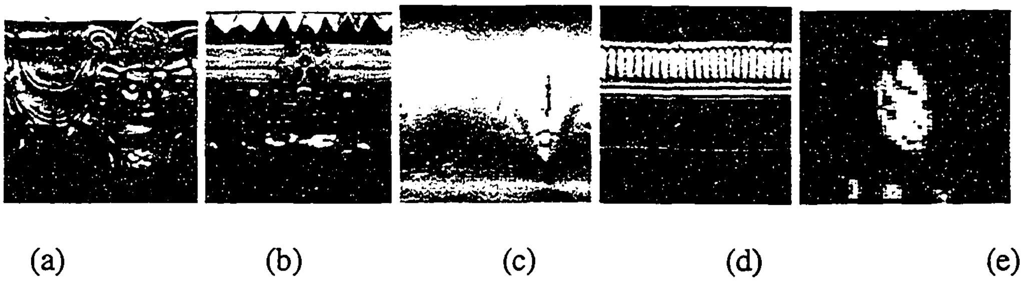


**Fig. 3.3:** *Texture patterns used to compare the total data size required in the fragmented and non-fragmented approach.*

| Model texture | Size (pixels) | JPEG Quality | 1 file (KB) | Sum of fragments |
|---|---|---|---|---|
| (a) | $1024^2$ | 60% | 566 | 293 |
|  | $256^2$ | 20% | 29 | 44 |
| (b) | $1024^2$ | 60% | 305 | 222 |
| (c) | $1024^2$ | 60% | 400 | 253 |
| (d) | $1024^2$ | 60% | 494 | 269 |
| (e) | $1024^2$ | 40% | 383 | 226 |
|  |  | 50% | 423 | 237 |
|  |  | 60% | 458 | 249 |
|  | $512^2$ | 40% | 111 | 62 |
|  |  | 50% | 122 | 66 |
|  |  | 60% | 130 | 70 |
|  | $256^2$ | 20% | 25 | 42 |
|  |  | 60% | 56 | 47 |

**Table 3.1:** *Initial results obtained from the fragmented and non-fragmented approaches.*

To verify the initial result, five different images of Mount Rushmore at resolution $R = 256^2, 512^2, 1024^2, 2048^2$ and $4096^2$ pixels (Fig. 3.4) were used. The resolution is original without interpolation because the analysis on sixteen interpolated texture images showed that the fragmentation approach performed even better if interpolated images are used. Four versions of each Mt. Rushmore image containing number of fragments $n = 2^2, 4^2, 8^2$ and $16^2$ respectively were generated. Each version was then tested using quality $Q$ = 100%, 90%, 80%, 60%, 40%, 20% and 10%. The observation is that, when $n \leq 256$ and each fragment size $\geq 128^2$ pixels, the sum of the fragment files $\Sigma f$ was smaller than the single non-fragmented file $F$ of equal JPEG quality. When $\Sigma f < F$ is true at a quality level, it is also true at lower quality levels.

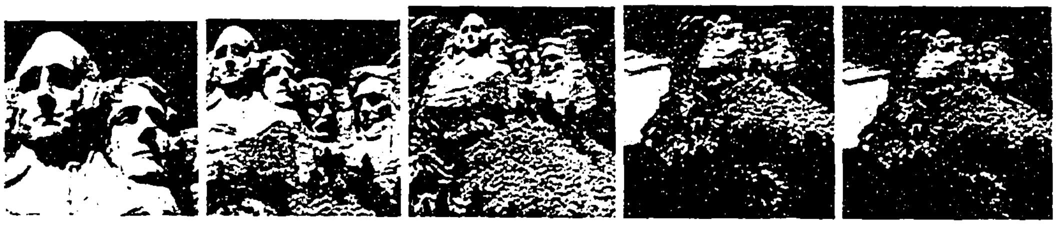


**Fig. 3.4:** *Thumbnails of Mt. Rushmore images with original resolutions at: $256^2$, $512^2$, $1024^2$, $2048^2$ and $4096^2$ pixels respectively.*

In Fig. 3.5 and Fig. 3.6, it is observed that at 100% quality when fragment size $\geq 128^2$ pixels and the number of fragments $\leq 16^2$, fragmentation does not increase the transmitted data. When increasing the number of fragments beyond $16^2$, both the JPEG decoding time and data size are in favor of a single non-fragmented file. When restricting the number of fragments to a maximum of $16^2$, the difference in decoding time is within a second for lower resolution images, and is within 2 seconds for images at a resolution of $2048^2$ and $4096^2$ pixels. This statistics is based on the performance of a Dell desktop computer, using Windows2000 at 1.8 GHz., with 40 GB disk space and 512MB RAM. We confirmed this finding again by using fifteen more images of different resolutions, and by applying various qualities to each image. The results are consistent. Fig. 3.7 and Fig. 3.8 compare data sizes at different qualities. It can be seen that when dividing into fragments of optimal size, the sum of fragments is not bigger than the single non-fragmented file of equal JPEG quality.

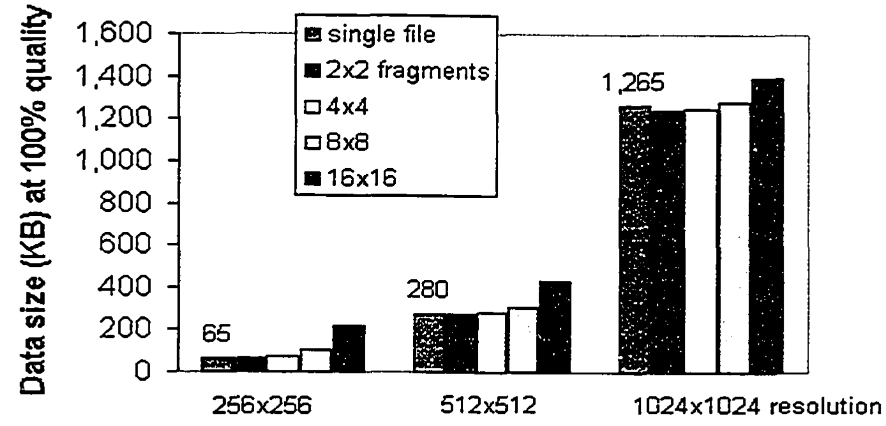


**Fig. 3.5:** *Data size of three Mt. Rushmore images at resolution $256^2$, $512^2$ and $1024^2$ pixels. At each resolution, the single file size (non-fragmented) is compared with the sum of fragment with $n = 2^2$, $4^2$, $8^2$ and $16^2$ respectively.*

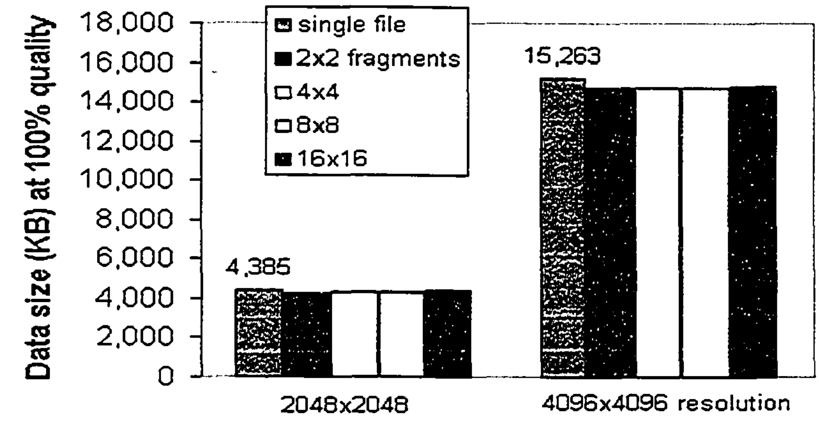


**Fig. 3.6:** *Data size of two Mt. Rushmore images at resolution $2048^2$ and $4096^2$ pixels. At each resolution, the single file size (non-fragmented) is compared with the sum of fragments with $n = 2^2$, $4^2$, $8^2$ and $16^2$ respectively.*

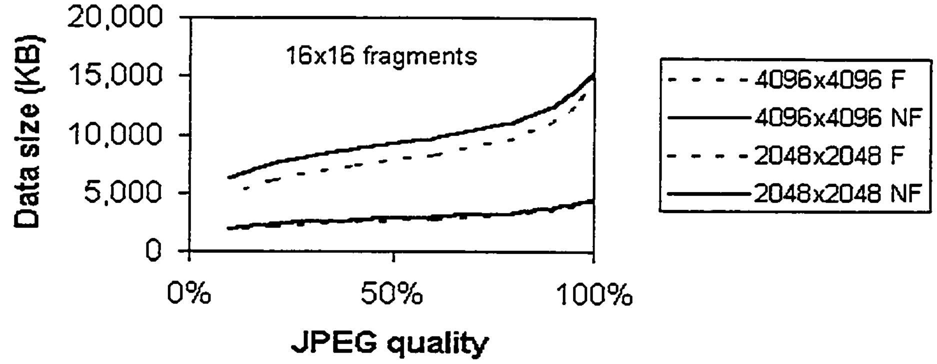


**Fig. 3.7:** *Using images of Mt. Rushmore to compare size of non-fragmented (NF) and fragmented (F) data at resolutions $4096^2$ and $2048^2$ respectively. Number of fragments is $16^2$.*

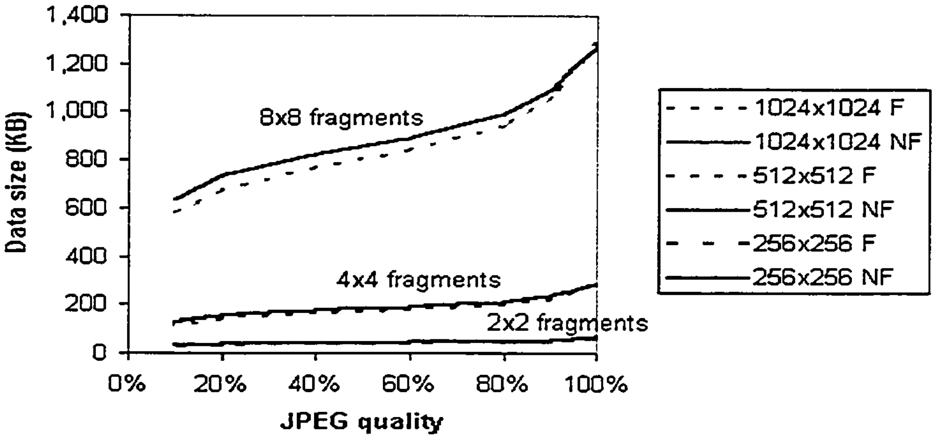


**Fig. 3.8:** *Using images of Mt. Rushmore to compare the size of non-fragmented (NF) and fragmented (F) data at resolutions $1024^2$, $512^2$ and $256^2$. Number of fragments is $8^2$, $4^2$ and $2^2$ respectively.*

Based on these experimental findings, one can keep the number of fragments *n* at $16^2$ or less when applying the adaptive transmission strategy. Note that there is a tradeoff between using $16^2$ fragments or less. The advantage of using fragments is for periodic bandwidth adaptation, so that the time deviation at the end of the transmission can be minimized. For this reason, more fragments are better. However, the JPEG coding/decoding time is proportional to the number of fragments. For example, when an image of 10.97M pixels is divided into $16^2$ fragments, the additional coding/decoding time using the Dell desktop computer is about 5 seconds, which is only 5% of the transmission time if we assume a network speed of 100KB/second. On the other hand, when using $8^2$ fragments the deviation from a given time limit increases from 1% to 2.5%, but the coding/decoding time is reduced to less than 2 seconds. When the number of fragments $n > 16^2$, the cost of fragmentation outweighs its benefit.

### 3.3.2 Comparison between Uniform and Variable Texture Quality

By assigning variable qualities to individual fragments based on their associated feature point density, the overall perception of the 3D objects can be improved. An example is shown in Fig. 3.9 by comparing two dog objects at scale $S_{25}$. Variable texture qualities are applied on the right dog, while a uniform quality of 10% is applied to the left one, maintaining a total texture data size of 41 KB each. Since the eye area of the dog model is associated with a higher feature point density, it is assigned a quality of 50% in the variable approach. It can be seen that the right dog has a better overall visual fidelity, contributed by the higher resolution texture around the eye area.

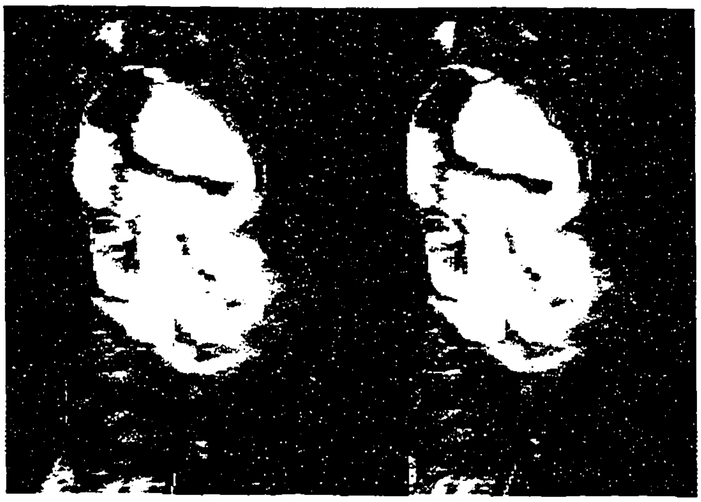


**Fig. 3.9:** *A snapshot of 3D texture mapped dog model showing (right) variable quality and (left) uniform quality.*

Since the HVS is less sensitive to details that subtend a small visual angle, the texture quality $Q_i$ at each scale $S_i$ needs to increase only when $i$ decreases towards a finer model. Given a viewing distance, the corresponding $S_i$ and $Q_i$ are selected. (We will determine how $S_i$ is mapped onto viewing distance by perceptual evaluation experiments in Chapter 6, Chapter 7 and Chapter 8). Instead of applying a uniform quality $Q_i$ to all fragments, a variable approach is used so that texture quality of each fragment varies depending on the feature point density associated with it.

### 3.3.3 Feature Point Density as a Visual Quality Predictor

Let's use the following grenade and nutcracker models to illustrate how different feature point densities affect the perception of 3D objects. The grenade (Fig. 3.10

c & d) has vertical structures on the surface, and therefore the feature point distribution is higher than the back of the nutcracker (Fig. 3.10 a & b), which is comparatively flat. Note that even if the texture quality is reduced by half, there is no significant perceptual degradation on the nutcracker (Fig. 3.10 b). However, the grenade on the right (Fig. 3.10 d) shows noticeably lower perceptual quality.

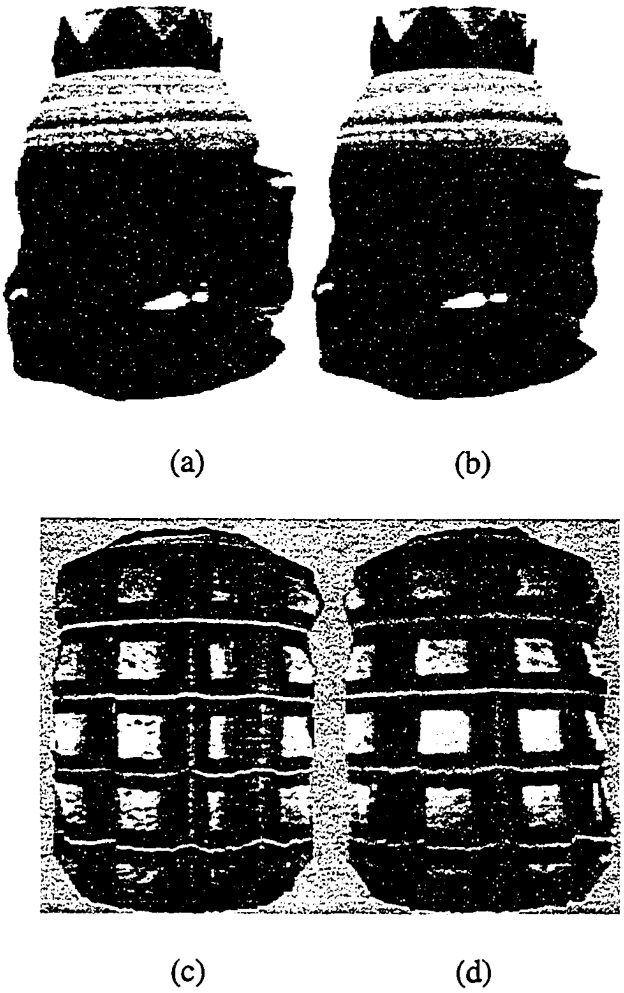


**Fig. 3.10:** *A snapshot of the nutcracker 3D model view from the back (a & b), and the military grenade model (c & d), with original texture quality (a & c), and half of the original texture quality (b & d).*

While the difference in quality of the grenades is obvious, the degraded shiny patch under the belt of the nutcracker (Fig. 3.10 b) is less noticeable. The observation is that compression on surfaces with high feature point density is more obvious to human perception, than on surfaces with low feature point density. The perception of depth, generated by shade and contrast, makes virtual objects look realistic [Nag84]. Higher feature point density surfaces are associated with higher contrast. Thus quality reduction in these areas has more adverse effect on the perception of depth, than reducing quality in low feature point density areas. Fig. 3.11, showing the Armadillo character used in appearance preserving simplification [COM98], is another example illustrating the perceptual impact of high and low feature point density.

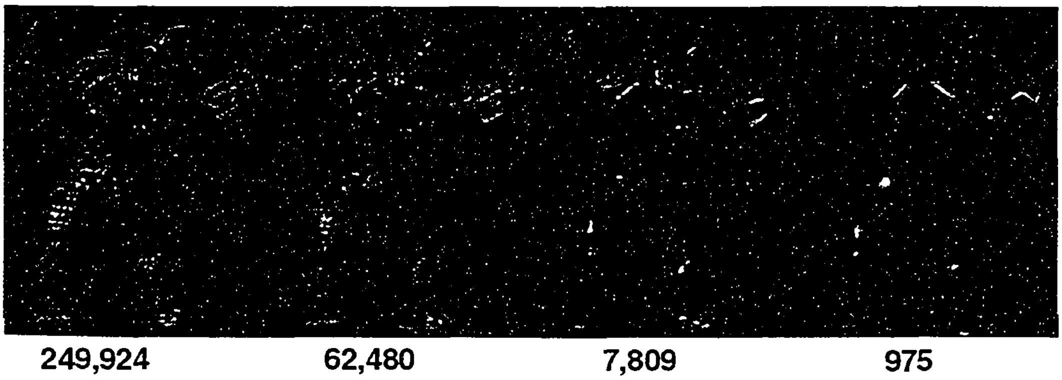


**Fig. 3.11:** *Different LOD of the Armadillo character with the number of triangles indicated at the bottom [COM98].*

Note that the surface detail (visual quality) decreases as the feature point density decreases from left to right.

Degradation can be compensated, to a certain extent, if an appropriate texture pattern is used. This compensation phenomenon is explained by texture masking [FPS*97]. To simplify the initial analysis, both the nutcracker and the grenade models have simple pattern. A more detail analysis of human perception is presented in Chapter 5.

Based on the previous analysis, let's adopt a variable approach by applying different qualities on texture fragments depending on the feature point density, instead of applying a uniform quality to all fragments. Furthermore, the variable qualities are computed adaptively based on the current bandwidth. An adaptive approach is necessary when transmitting data on the Internet because bandwidth fluctuates and can adversely affect the expected quality of experience (QoE).

Before explaining how variable qualities are assigned to different fragments, let's define the terminology used in the adaptive approach:

Let $S_i$ be the scale $i$, *i.e.*, $0 \leq i \leq n$ where $S_0$ is the original signal and $S_n = S_{max}$ is the scale at infinity.

Let $\Delta Q$ be the quality tolerance limit for each scale imposing an upper and a lower bound. Analogous to the depth of field in photography, outside which an object is out of focus, $\Delta Q$ is the tolerance range when displaying 3D objects at a given distance. Given a viewing distance, the HVS finds this range of qualities satisfactory.

Let $Q_i$ be the default texture quality associated with scale $S_i$.

Let $(x,y)$ represents the $x$ and $y$ coordinates in the $N_x$x$N_y$ fragment map.

Let $q_i(x, y)$ be the texture quality of fragment $(x,y)$ at $S_i$.

Let $f_i(x,y)$ be the number of feature points in fragment $(x,y)$ at scale $S_i$.

Let $f_i^{\max}$ be the maximum number of feature points in a fragment at scale $S_i$.

Let $f_i^{\min}$ be the minimum number of feature points in a fragment at scale $S_i$.

Let $\eta_i(x,y)$ be the normalized value of $f_i(x,y)$.

Let $\bar{\eta}_i$ be the mean of the normalized values $\eta_i(x,y)$.

Let $\Gamma$ be the feature point distribution threshold, *i.e.* $0 \leq \Gamma \leq 1$.

Let $d_i(x,y)$ be the data size of fragment $(x,y)$ at $S_i$ with quality $q_i(x, y)$.

Let $D_i$ be the data size of all fragments at $S_i$.

At a given scale $i$, $f_i(x,y)$ is normalized as:

$$\eta_i(x,y) = \frac{f_i(x,y) - f_i^{\min}}{f_i^{\max} - f_i^{\min}}. \qquad (3.1)$$

The texture quality $q_i(x, y)$ of fragment $(x,y)$ at scale $S_i$ is computed as:

$$Q_i + (\eta_i(x,y) - \Gamma)\, \Delta Q. \qquad (3.2)$$

In the current implementation, the threshold $\Gamma$ is set to $\bar{\eta}_i$. Fragments with $\eta_i(x, y) = \Gamma$ are assigned quality $Q_i$. If $\eta_i(x, y) > \Gamma$, the fragment texture quality is higher than $Q_i$. If $\eta_i(x, y) < \Gamma$, texture quality is lower than $Q_i$. $\Delta Q$ controls the deviation (+/-) from $Q_i$ Regions on the 3D model surface with higher feature point density are displayed with higher quality, and lower density regions are displayed with lower quality texture. By increasing or decreasing $\Gamma$, the overall quality of the texture image is decreased or increased accordingly, along with the data size. For each model texture, a lookup table is used to record $D_i$, $d_i(x,y)$ and $q_i(x, y)$ for a range of $\Gamma$. Given a time limit and the current bandwidth, the appropriate $D_i$, and the associated $d_i(x,y)$ are selected as the reference data size to transmit the first fragment. Details of this transmission strategy and bandwidth adaptation will be discussed in Chapter 5.

## 3.4 Conclusion

In Chapter 3, we have introduced the TexMesh model, and described how feature points are distributed onto the texture fragments. Experiments were conducted to show that not only the fragmentation approach does not increase the total transmitted data for high resolution texture image divided into optimal fragment size, but also it improves the overall visual fidelity of the 3D object. Examples were given to illustrate how feature point density is related to shade and contrast, and how it is used as a visual quality predictor to determine texture quality assignment. In Chapter 4, the Harmonic Time Compensation Algorithm (HTCA) will be introduced. We will also discuss how fragmentation and variable quality assignment are used to support bandwidth adaptation and quality of service.

# Chapter 4

## Adaptive Online Transmission of Photo-Realistic Textured Mesh

Efficient bandwidth utilization and optimal quality of service are among the main objectives when transmitting 3D objects. Since mesh data is usually small compared to texture data (as explained in Chapter 3), our focus is to adapt the texture quality to the current bandwidth within a specified, or acceptable, transmission time. A historic average can be used to estimate current bandwidth [CBZ*01], but this approach can cause unacceptable over or under estimation because of bandwidth fluctuations. An optimal bandwidth monitoring approach can provide a more accurate estimation by sacrificing a portion of the transmission time [YCB03]. In this chapter, an adaptive approach is proposed, which does not need to sacrifice transmission time for bandwidth estimation, while efficiently adjusting the quality of the fragments not yet transmitted.

In order to optimize bandwidth, a multi-scale incremental simplification approach is most suitable for online applications. Multi-scale allows a coarse version to be refined incrementally. However, if a simplification process involves relocation of mesh vertices or texture coordinates between scales, then an entirely new version instead of the vertices added to a coarse version, has to be transmitted [BW00]

[GH97] [SG01] [Tur92]. In the progressive meshes method, although the original mesh can be recovered exactly after all the data are received, the edge collapse transformation creates new vertices and the *vsplit* record stream increases network workload [Hop96]. Experimental results in Chapter 6 will show that given a viewing distance two visual stimuli can be quantitatively different, but perceptually similar. For example, the two nutcracker models have similar visual quality, but the mesh resolution on the left is ten times that on the right (Fig. 4.1). Thus it is more beneficial to allocate the available computational and network resources to other related multimedia data, e.g. texture, instead of striving for an exact recovery of the original or denser mesh. Xia's [XEV97] adaptive real-time LOD technique also involves vertex relocation. We apply vertex removal and edge collapse without affecting the 3D point and the associated 2D texture coordinates.

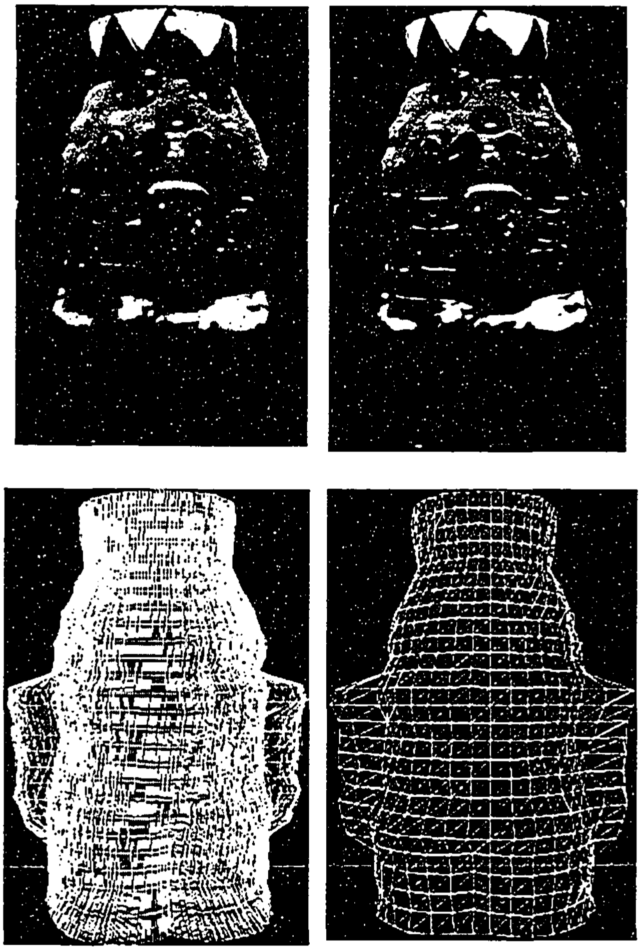


**Fig. 4.1:** *The two nutcracker objects are perceptually similar although the mesh resolution on the left is ten times that on the right (18,720 vs. 1,872 triangles corresponding to 46 and 8 KB in .zip format).*

In recent years, researchers started to incorporate color and texture into their mesh simplification models. When texture is mentioned in the literature, it often refers to synthetic or animated texture [Tur91]. Synthetic texture can be estimated. For example, when walking through an animated scene, the next frame can be predicted based on available neighboring data [CMF99]. The authors demonstrated that this technique has better quality and higher compression factor than MPEG. An image-driven simplification method is used to display textures using images from multiple views [LT00]. However, rendering the entire model for every edge in every viewpoint for different scales is expensive, even with hardware-accelerated rendering. The high-resolution texture used in the TexMesh model is different from the per pixel color stored in each vertex [COM98] [GH98] [SGR96] [SSG01], where interpolation is required when texture resolution is higher than mesh resolution. For certain applications requiring real-life texture, interpolating colors between vertices is not acceptable. The non-interpolated texture used in this thesis has resolution much higher than the mesh. It was observed in perceptual experiments that the human visual system is more sensitive to higher texture resolution after the mesh reaches an optimal density [PCB05]. In general, realism of virtual objects increases as texture quality increases resulting from the perception of depth [Nag84]. The TexMesh model uses photo-realistic texture images, with resolution up to millions of pixels, suitable for displaying on small monitors or high definition screens in reality centers. Photo-texture is used in compressed texture maps [YFM00], but their effort is on recovering geometry from texture patches retrieved from multiple photographs. A distance-based technique is applied to photo-textured terrain [LKH*95]; however, color interpolation between pixels is necessary to avoid blocky appearance of terrain texture. To the best of our knowledge, the TexMesh approach is the first algorithm to introduce geometry driven texture reduction for adaptive online transmission of 3D objects, taking into account visual fidelity. This adaptive approach does not replace any image compression techniques. Instead, it enhances compression by introducing bandwidth adaptation at the application level, in addition to the reduction at the image level.

In [OC00], the joint geometry/texture progressive coding method applies wavelet transform to encode the mesh and texture data for transmission, but the method cannot adapt to fluctuating bandwidth. Wavelets are also used to create space optimized texture maps, which did not require any hardware compression support [BTB02]. In the proposed method, scale-space filtering and zero-crossing detection are applied to extract feature points. Each scale is associated with a default texture quality based on the number of feature points in it, and the quality of each fragment is allowed to deviate within a limit. The quality is later readjusted according to current bandwidth. By relating each scale to a viewing distance, automatic selection of a corresponding simplified textured mesh is possible.

Note that one major difference between wavelets and the scale-space approach is that the former scales up or down by a fixed factor of 2. In the scale-space approach, the scales at which changes occur depend on the surface features of individual objects, and are not fixed beforehand. In this chapter, an extension of the textured mesh (TexMesh) model incorporating a dynamic strategy is proposed, which adapts the current bandwidth when computing texture quality of the next transmitted fragment. A Harmonic Time Compensation Algorithm (HTCA) is applied to ensure optimal use of the time limit and bandwidth. By splitting the texture into fragments, distributed transmission and parallel processing is possible.

## 4.1 Overview of Adaptive Online Transmission Strategy

The strategy for adaptive online transmission of 3D objects has several components, which are shown in Fig. 4.2.

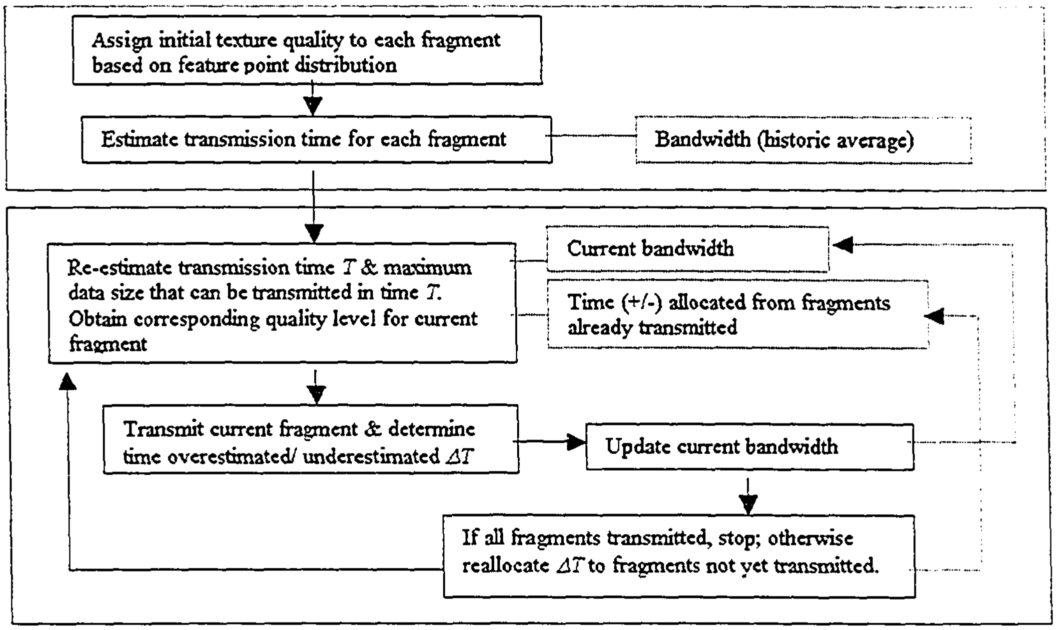

Assign initial texture quality to each fragment based on feature point distribution
Estimate transmission time for each fragment
Bandwidth (historic average)
Re-estimate transmission time $T$ & maximum data size that can be transmitted in time $T$. Obtain corresponding quality level for current fragment
Current bandwidth
Time (+/-) allocated from fragments already transmitted
Transmit current fragment & determine time overestimated/ underestimated $\Delta T$
Update current bandwidth
If all fragments transmitted, stop; otherwise reallocate $\Delta T$ to fragments not yet transmitted.

**Fig. 4.2:** *A block diagram of the adaptive online transmission strategy.*

The components are divided into two groups: preprocessing is described in the upper group, and online process is described in the lower group. The feature point density in each texture fragment is a value between zero and one, which is mapped onto a compression scale. For example, the JPEG compression scale from 0% to 100% is used in the current implementation. While wavelet coding applies to the entire image and is geometry-independent, the approach proposed here supports variable quality determined by the density of surface structures. Note that JPEG [JPG05] is currently used for convenience and wide support on the web and in JAVA; however, standards such as JPEG2000, and other efficient compression algorithms, can be used in the future to code fragments.

## 4.2 Adaptive Bandwidth Monitoring and Texture Quality Determination

Because of bandwidth fluctuations, current bandwidth has to be monitored periodically in order to maintain a good estimate of the data size that can be transmitted in a specified time $T_0$. To minimize the discrepancy, we reallocate the time surplus/deficit to the fragments not yet transmitted.

We adhere to the notation presented in Chapter 3, where $\eta_i(x,y)$ is defined as the normalized feature point density of fragment $(x, y)$.

The $n = N_X \mathrm{x} N_Y$ fragments are pre-sorted in decreasing $\eta_i(x, y)$ values, *i.e.*, from 1 to 0,

$$F_{\text{list}} = \{ F_1, \ldots, \bar{F}, \ldots, F_n \},\ \textit{i.e.},\ \bar{F} \text{ has quality } Q_i.$$

The first fragment to be transmitted is $\bar{F}$ with quality $Q_i$.

Based on a time limit $T_0$ and a historic bandwidth average $\beta_0$, we estimate maximum data size to be transmitted as:

$$D_1 = T_0 * \beta_0$$

Where:

$\beta_k$ is the current bandwidth (KB/sec.) recorded after $k$ fragments are transmitted, *i.e.*, $0 \leq k < n$. $\beta_0$ is the historic average bandwidth before transmission, $T_k$ is the time left after $k$ fragments are transmitted, $T_0$ is the original time limit (seconds) specified, and $D_{k+1}$ is the maximum data size that can be transmitted given $\beta_k$ and $T_k$.

The fragment list $F_{\text{list}}$ (best matching $D_1$) is selected from a lookup table generated during preprocessing. The size of $\bar{F}$, $d_1$, is used to estimate the transmission time of the first fragment:

$$\vartheta_1 = T_0 * \frac{d_1}{D_1}\ \text{, or } \vartheta_1 = \frac{d_1}{\beta_0},\ (4.1)$$

where $d_k$ represents the data size of the $k^{th}$ fragment, and $\vartheta_k$ is the estimated transmission time for fragment $k$

We estimate the transmission time $\vartheta_g$ for all the remaining fragments, *i.e.*, $2 \leq g \leq n$:

$$\vartheta_g = T_0 * \frac{d_g}{D1}. \quad (4.2)$$

After $d_1$ is transmitted, we have the updated bandwidth $\beta_1$ based on the time $\gamma_1$ recorded when transmitting $d_1$:

$$\beta_1 = \frac{d_1}{\gamma_1}, \quad (4.3)$$

where $\gamma_k$ is the actual time needed to transmit the fragment *k*.

The next fragment is selected as follows:

(a) The leftmost fragment in F$_{\text{list}}$ if $\beta_1 \leq \beta_0$, and

(b) The rightmost fragment in F$_{\text{list}}$ if $\beta_1 > \beta_0$

Let $\Delta T_k$ be equal to the difference between estimated and actual transmission time for $k^{th}$ fragment; *i.e.*, $\vartheta_k - \gamma_k$

Let $\Delta t_k$ be the cumulated compensation time (+/-) allocated to the $k^{th}$ fragment from the previous *k*-1 fragments (refer to Algorithm 1 below), and let $wt_f$ be the weight applied to the $f^{th}$ fragment when allocating $\Delta T_{k-1}$, *i.e.*, $k \leq f \leq n$.

The basic adaptive algorithm is the following:

(a) If the actual bandwidth is lower than the estimated one, loss of time $\Delta T_1$ has to be compensated when transmitting the remaining $n$-1 fragments, so that each remaining fragment has to share a portion of $\Delta T_1$. Instead of the initial $\vartheta_2$ computed in Equation (4.2), the $2^{nd}$ fragment has $wt_2 * \Delta T_1$ seconds less, where $wt_2$ is the assigned weight. We regain the time by transmitting the leftmost fragment in $F_{list}$ with reduced quality.

(b) Similarly, if the actual bandwidth is greater than the estimated one, the gained time $\Delta T_1$ is allocated to the remaining $n$-1 fragments, so that each remaining fragment can have additional time. Instead of the initial $\vartheta_2$, the $2^{nd}$ fragment has $\vartheta_2 + wt_2 * \Delta T_1$ seconds. We adjust the time by transmitting the rightmost fragment in $F_{list}$ with increased quality.

Based on the revised $\vartheta_2$, we compute: $d_2 = \beta_1 * \vartheta_2$; and then obtain corresponding quality for the $2^{nd}$ fragment from the lookup table using $d_2$. In general, after $k$-1 fragments are transmitted:

$$\Delta T_{k-1} = \vartheta_{k-1} - \gamma_{k-1} \quad (4.4)$$

$$\Delta t_k = \Delta t_k + \Delta T_{k-1} * wt_k . \quad (4.5)$$

$$\vartheta_k = \vartheta_g + \Delta t_k \quad (4.6)$$

$$\beta_{k-1} = \frac{d_{k-1}}{\gamma_{k-1}} \text{ , and (4.7)}$$

$$d_k = \beta_{k-1} * \vartheta_k \text{ (4.8)}$$

The computation of the weight $wt_k$ in Equation (4.5) is explained in Algorithm 4.1. The quality for the $k^{th}$ fragment is obtained from the lookup table based on $d_k$. Bandwidth fluctuation has a larger impact on the quality if $\Delta T_k$ has to be shared by a smaller number of fragments. Therefore, fragments with quality ≅ $Q_i$ are transmitted last to allow more flexibility for adjustment within the control limit $\Delta Q$. Once the transmission is started, the quality of a fragment is self-adjusted depending on the updated bandwidth.

## 4.3 Harmonic Time Compensation Algorithm

Since all late fragments have to share all preceding allocations, Algorithm 4.1 assigns decreasing weights $(\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, .....)$ from $(k+1)^{th}$ to $n^{th}$ fragments, when reallocating $\Delta T_k$.

**Algorithm 4.1 – Harmonic Time Compensation**

```
After transmitting $k^{th}$ fragment,
    let $\chi$=2;
    compute $\varsigma_k = \sum_{j=2}^{n-k+1} \frac{1}{j} = \ln(n-k+1)$;
    for ($i = k+1$; $i \leq n$; $i$++) {
        $wt_i = \frac{1}{\chi^* \varsigma_k}$;
        $\Delta t_i += \Delta T_k * wt_i$; // allocate to remaining fragments
        $\chi$++;
    }
```

There are two questions we have to address:

(1) How efficient is the algorithm with respect to bandwidth optimization for a given time?

(2) How does the adaptive approach affect the perceptual quality?

To prove the efficiency of the algorithm, we define $\Pi$ as the time surplus/deficiency with respect to the limit $T_0$. $\Pi$ is composed of three errors: estimation error $E_{est}$, allocation error $E_{alloc}$, and compensation error $E_{comp}$. In Theorem 4.1, we establish the upper and lower bound of $\Pi$ (Proof: See Appendix A).

**Theorem 4.1:** $\Pi$ is bounded by:

$$\Delta T_n + (\Delta T_{n-1}/2) + (1.088 + \ln|\ln(n)|)\Lambda$$

Where $\Lambda$ is defined as the average difference between the estimated and actual transmission time for the first $n$-1 fragments,

$$i.e.\ \Lambda = \frac{\sum_{j=1}^{n-1} \Delta T_j}{n-1}.$$

The upper and lower bounds in Theorem 4.1 are verified by experimental results in the next section. We will show that our adaptive approach does not have an adverse effect on perceptual quality for reasonable bandwidth fluctuation.

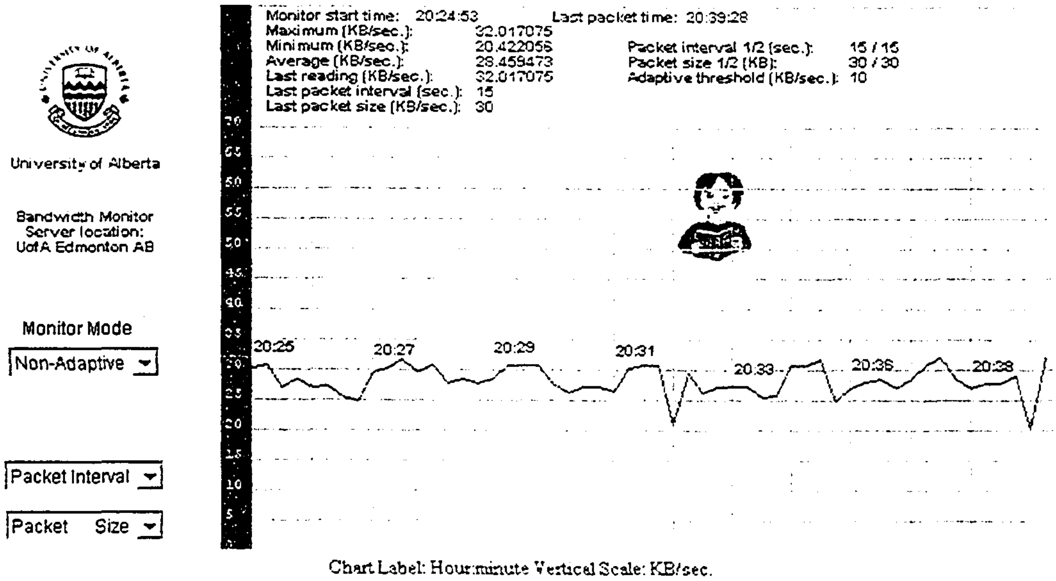
## 4.4 Experimental Results

Let $n$ = 256, applying Theorem 4.1 and substituting 256 into Equation A.3 in Appendix A, we obtain:

$$E_{comp} \leq 2.8\Lambda \text{ if } \Lambda \geq 0, \text{ and } E_{comp} \geq 2.8\Lambda \text{ if } \Lambda < 0$$

Since $\Lambda$ is the average deviation over the entire transmission period, it is expected to be small. The other two components of $\Pi$: estimation error $E_{est}$ (Appendix A Equation A.1) and allocation error $E_{alloc}$ (Appendix A Equation A.2), can be minimized by using sufficiently small data size for the last two fragments.

In order to see how $\Pi$ responds to bandwidth fluctuation, a bandwidth monitor using JAVA and client-server architecture was implemented (Fig. 4.3). When non-adaptive mode was chosen, testing packets of fixed size were transmitted from server to client at regular intervals. The client computed the current bandwidth based on the time to transmit the packet (end-to-end) just received. When adaptive mode was chosen, the packet size and time interval would be adjusted depending on whether the current bandwidth was below or above a predefined threshold.



Monitor start time: 20:24:53
Last packet time: 20:39:28
Maximum (KB/sec.): 32.017075
Minimum (KB/seo.): 20.422056
Average (KB/sec.): 28.459473
Last reading (KB/sec.): 32.017075
Last packet interval (sec.): 15
Last packet size (KB/sec.): 30
Packet interval 1/2 (sec.): 15 / 15
Packet size 1/2 (KB): 30 / 30
Adaptive threshold (KB/sec.): 10

Monitor Mode: Non-Adaptive

**Fig. 4.3:** *An example of bandwidth fluctuation on an Ethernet connection.*

Three sets of bandwidths were extracted from an Ethernet connection on different days and different times (Table 4.1). We then varied the value of $\beta_0$ below and above the average of the sample set within a reasonable range (Table 4.2). The test file was 418KB with 256 fragments.

| Bandwidth sample set | Actual bandwidth average (KB/sec) |
|---|---|
| 1 | 41.68 |
| 2 | 45.64 |
| 3 | 42.07 |

**Table 4.1:** *Three bandwidth sample sets were taken from an Ethernet connection on different days and different times.*

| $\beta_0$ (KB/sec) | $\Lambda$ (sec) | $\Pi$ Surplus/deficit (+/-)% of limit | $T_0$ Time Limit (sec) | Static approach % of limit |
|---|---|---|---|---|
| 20 | 0.038 | 0.399 | 20.9 | 52.0 |
| 23 | 0.031 | 0.346 | 18.17 | 44.8 |
| 26 | 0.023 | 0.280 | 16.07 | 37.6 |
| 29 | 0.015 | 0.195 | 14.41 | 30.4 |
| 32 | 0.011 | 0.142 | 13.06 | 23.2 |
| 35 | 0.003 | 0.010 | 11.94 | 16.0 |
| 38 | -0.001 | 0.010 | 11 | 8.8 |
| 41 | -0.004 | -0.027 | 10.19 | 1.6 |
| 44 | -0.008 | -0.034 | 9.5 | -5.5 |
| 47 | -0.008 | -0.058 | 8.89 | -12.0 |
| 50 | -0.012 | -0.058 | 8.36 | -19.8 |
| 53 | -0.012 | -0.090 | 7.88 | -27.2 |

**Table 4.2:** *Experimental results show that the Harmonic Time Compensation Algorithm had less than 1% deviation for a given time limit. $\beta_0$ was used for initial bandwidth estimation (Sample set 1).*

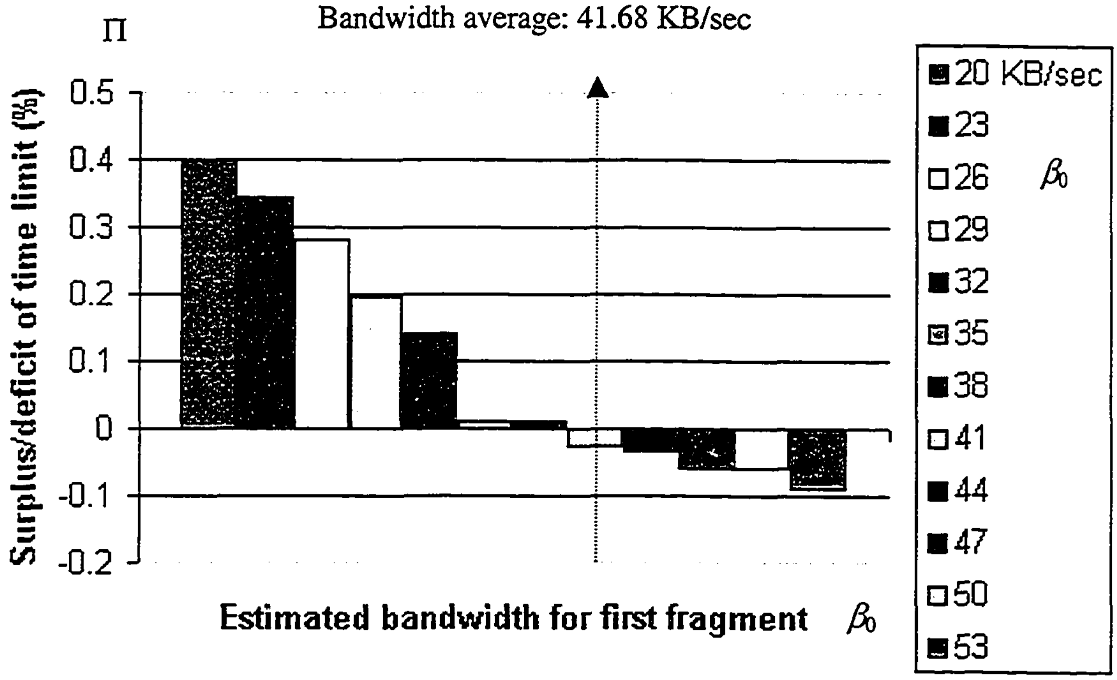


**Fig. 4.4:** *Experimental results show that the overall time surplus/deficit* Π *was close to zero, when the initial estimated bandwidth* $\beta_0$ *was close to the bandwidth average of the transmission period (Sample set 1).*

One can observe that Π is minimum, when Λ is close to zero. That is when the estimated bandwidth $\beta_0$ is close to the bandwidth average (Fig. 4.4). Similar trends were obtained from samples 2 and 3. By keeping the $n^{\text{th}}$ and $(n\text{-}1)^{\text{th}}$ fragments sufficiently small, as in our experiments, the deviation from the time limit is within 1%. For comparison, the last column shows the discrepancy in percentage of time limit, should historic average bandwidth be used in a static approach.

To see how variable quality affects the overall visualization, we used $\beta_0$ = 32 and 50, together with a sample average of 41.68 (KB/sec), and applied to the dog texture. The original texture has quality $Q_i$ equal to 80% and $\Delta Q_i$ is [40%, 100%]. Fig. 4.5 shows that the perceptual quality is maintained, after applying variable qualities to fragments adaptively in order to satisfy the time limit. Given the estimated texture in the middle, actual quality is increased in case of underestimation of actual bandwidth (left), and actual quality is decreased for overestimation (right).

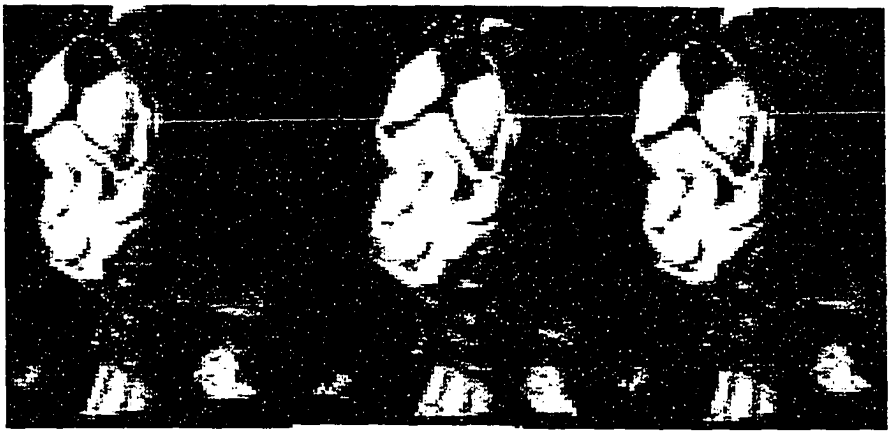


**Fig. 4.5:** *Initial estimated texture (middle), increased quality (left) and decreased quality (right).*

We performed fifteen more simulations using a wider range of bandwidths, including dial-up and high-speed networks. Experimental results are consistent with earlier findings. Results from seven higher speed networks are plotted in Fig. 4.6 and five from lower speed networks are plotted in Fig. 4.7. It can be seen that when the initial estimation is within a reasonable range of the bandwidth average, the overall deviation is less than 1% of the given time limit. The deviation gets smaller when the initial estimation $\beta_0$ approaches the actual average bandwidth of the entire transmission period. Since the initial estimation is based on historic average, it is not expected to be very different from the actual average for a reasonably long transmission period. Also, note that when networks have similar average and if a network is relatively stable, e.g. 3.31 KB/sec. as shown by (a), (b), (c) and (d) in Fig. 4.7, the surplus/deficit curves follow a similar trend. On the other hand, although the bandwidth average of 111.94 and 110.89 KB/sec are close (Fig. 4.6), the surplus/deficit curves have different trends due to unstable bandwidth fluctuations.

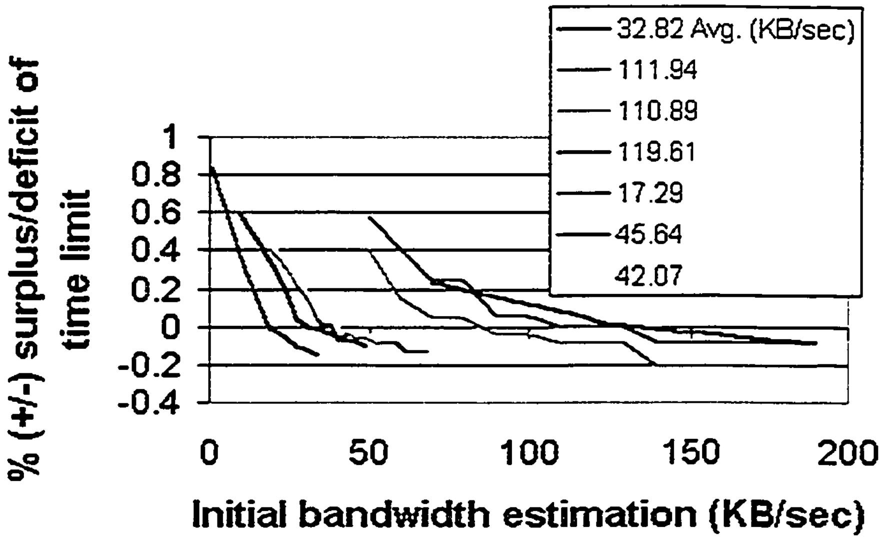


Fig. 4.6: *Simulation results of seven higher speed networks — By applying different initial estimated bandwidth* $B_0$, *the final deviation from a given time limit is with 1%.*

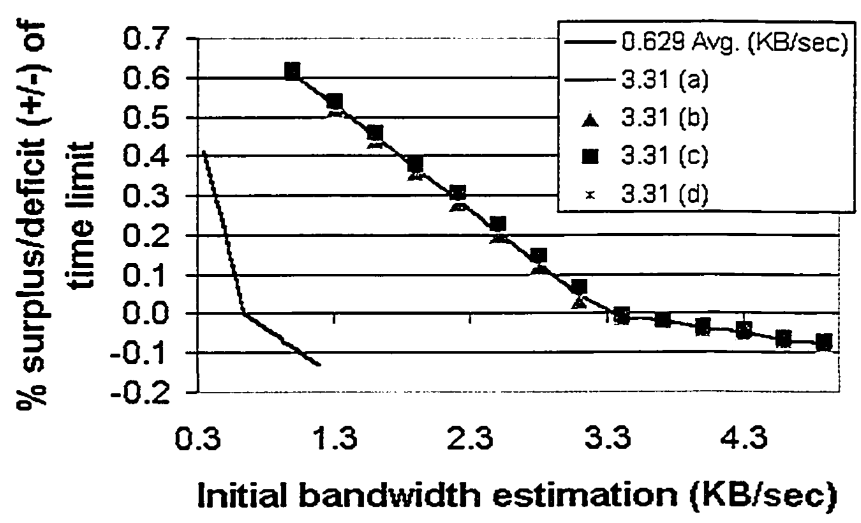


Fig. 4.7: *Simulation results of five lower speed networks —By applying different initial estimated bandwidth* $B_0$, *the final deviation from a given time limit is with 1%.*

The adaptive approach is most suitable for delivering best-effort QoS. It also supports distributed and parallel processing, by storing texture fragments at different sites. Texture data can be transmitted incrementally to refine the previous version. In Chapter 9, an integrated framework will be presented combining feature extraction, geometry simplification associated with texture reduction, bandwidth adaptation and perceptual evaluation.

Although feature point density can be used as a visual quality predictor, geometry driven texture reduction alone is not sufficient to predict the final visual fidelity. In Chapter 8, I will present a VQP model, which includes both geometry and texture driven predictors to enhance the resulting quality.

## 4.5 Conclusion

This chapter shows how texture fragmentation and variable quality assignment can be used to support efficient online transmission. An adaptive Harmonic Time Compensation Algorithm was introduced. Experimental results show that HTCA can adapt to bandwidth fluctuation well and thus provide better QoS. Perceptual quality is a main component in the TexMesh framework. Before looking into different visual quality predictors, in Chapter 5, the various environmental and psycho-visual factors that can affect the visual quality of 3D objects will be reviewed.

# Chapter 5

## Perceptual Issues in a 3D TexMesh Model

Visual fidelity is a fundamental issue in 3D visualization. Different techniques, such as illumination [VMK*00], perception of depth [Nag84], and visual masking [FPS*97], have been proposed to improve visual fidelity and thus enhance the realism of 3D objects in the virtual world. Many measuring criteria were suggested to compare the performance of these techniques. Comparisons were traditionally focused on geometric metrics but, in recent years, perceptual metrics have gained increasing attention [Lue01] [LH01] [PCB05] [RP03] [WFM01] [WLC*03]. While geometric metrics provide clues leading to a decision, perceptual metrics are more reliable when making a final judgement because visual fidelity is ultimately determined by the HVS. Applying a perceptual metric is complex because human judgement is rarely unbiased, often being affected by environmental and psychological factors. Successful perceptual evaluation experiments, therefore, must use a sufficiently large sample size, and preconditions sometimes have to be set. For example, in some experiments judges (subjects) are divided into age groups, or categorized according to academic background. The goal is to obtain the general perceptual behavior of the population, based on the statistics drawn from the sample.

## 5.1 Psycho-Visual Factors Affecting Visual Quality

Geometry and texture are two basic components of a 3D textured mesh object. Visual quality of the geometry and texture may be affected by various environmental and psycho-visual factors [PCB05] [LH01] [Nag84] [Lim79]. The major factors can be summarized as follows:

- Visual threshold [LH01] [Lim79] — Vanishing point at which a stimulus becomes just visible or invisible.
- Luminance of the background in contrast with the stimulus [GW02] [Lim79] — If the background is plain, the visual threshold will change depending on the luminance of the background in contrast to the object. This relationship is described by Weber's Law, which states that the threshold is proportional to the background luminance.
- Spatial visual masking [Lim79] — A large change of luminance may be present across an edge. Such a change reduces the ability of the eye to detect distortions spatially adjacent to the changes. The detection threshold (or the contrast corresponding to the threshold of perception) of a stimulus varies inversely as a function of its distance from the edge. As a target moves across an edge, the threshold on the light side of the edge is increased.
- Temporal visual masking — Post-masking (backward masking) occurs if the perception of a stimulus is affected by a subsequent strong signal. A visual

backward masking model [BT03] suggested that, when two successive stimuli are presented within 0 to 200ms intervals, the recognition of the first stimulus (the target) can be impaired by the second (the mask). Pre-masking occurs if perception is affected by a strong signal beforehand.

- Spatial frequency (cycles per degree of visual arc) — Minimum detectable difference (sensitivity threshold) in luminance between a test spot and a uniform visual field, increased linearly with background luminance at daylight levels. However, virtual scenes and stimuli are not uniform, and contain complex frequency content. Outside a small frequency range, the sensitivity threshold drops off significantly. This phenomenon has led to the study of the perceptibility of contrast gratings (sinusoidal patterns that alternate between two extreme luminance values) and the following terminology:

  a) Threshold contrast at a given spatial frequency is the minimum contrast that can be perceived in a grating.

  b) Contrast sensitivity is the reciprocal of threshold contrast.

  c) Contrast sensitivity function (CSF) plots contrast sensitivity against spatial frequency. It describes the range of perceptible contrast gratings [LH01].

- Feature masking — Presence of features will affect the general impression of the entire stimulus. For example, recognition of the ears of a rabbit outweighs the discrimination of shape (Fig. 5.1). This concept is well understood in the cartoon industry.

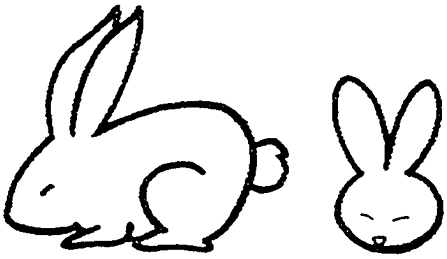


**Fig. 5.1:** *Both stimuli are recognized as a rabbit based on the feature "ear", although their geometric representations are very different.*

- Texture masking — It can be observed that the noise added to a low frequency (or plain) background is much more visible than that added to a high frequency (or flowery) background. A model was developed in computer graphics [FPS*97], which allows the choice of texture pattern to hide the effects of faceting, banding, aliasing, noise, and other visual artifacts. This technique is suitable for applications using synthetic texture; however, for applications displaying intrinsic photo-texture of the object, choice of a different pattern is not an option.
- Short-term memory — The influence of a strong stimulus will last for a short time, thereby imposing a smoothing effect on the distortion measure. It is worth noting that people are more likely to remember bad quality than good quality.

- Visual acuity — The fovea, occupying roughly the central 1° of vision, is the region of highest sensitivity. Visual acuity, measured as the highest perceptible spatial frequency, is lower in the periphery than at the fovea.
- Visual depth — A sensation of reality occurs because of visual depth perception. Depth sensitivity – the ratio of viewing distance to depth discrimination threshold – is directly proportional to the viewing distance. Sharp edge, clear texture, shade, and surface gloss strengthen the sensation of depth.
- Prior knowledge or Expectation — Prior knowledge imposes on the viewer an expected geometry and texture representation. Deviation from expectation degrades visual fidelity. If a stimulus appears in an orientation different from what the viewer is familiar with, the viewer's ability to discriminate decreases.

In video applications, satisfactory frame-rate (and network latency if transmitted over the network) is an important consideration. The TexMesh model is designed for efficient 3D visualization within time and visual constraints. The focus is an optimal overall time, instead of minimizing time intervals between data packets. Spatial and temporal visual masking can have a significant effect on the continuity of a stream of video frames, but are not essential in visualizing 3D objects discussed in this thesis. The focus of the TexMesh model is on the visualization of 3D objects after a specified time is elapsed. Continuity of intermediate data streams is not important as long as the final result is satisfactory. In the next section, the major factors that need to be taken into account when conducting perceptual evaluation experiments on 3D textured meshes will be discussed.

## 5.2 Preparation for Visual Discrimination Experiments

The Human Visual System (HVS) is complex, and many researchers have concluded that the determination of visual fidelity cannot be replaced by simple mathematics such as the MSE (Mean Square Error). Taylor *et al.* [TPA98] conducted an experiment comparing an original image with (a) an image distorted by a slight shift in luminance values, and (b) an image distorted by JPEG quantization. Although image (a) was visually similar to the original image, while image (b) was obviously degraded, the MSE for image (a) is greater than that for image (b). A perceptual metric is believed to be more accurate than a geometric metric in determining visual quality, because the ultimate decision is made by the HVS. As a result, many perceptually driven simplification techniques on 3D models have been suggested in the literature. A common approach for verifying these techniques is to compare the evaluation generated by the algorithm with human judgement. Another approach is to apply perceptual theory to select visual difference predictors (VDP), and incorporate them into the algorithm. VDP was first introduced in 1992 [Dal92], however, previous experiments had focused on static 2D images [TAP98] [TPA98], or a view-dependent discrimination [LH01]. A selected number of optimized views were used to compare stimuli in [WFM01]. Only recently, a 3D view-independent technique was introduced to perform an overall quality evaluation of a 3D object [PCB05]. The criteria described in the following sections are important to achieving accurate results and are taken into account in our perceptual experiments.

### 5.2.1 View-Independence

In a static display or view-dependent approach, judges can compare only a limited number of silhouettes [WFM01] [LH01]. A viewer's gaze tends to focus at the center of the object, thus judgement can be affected by visual acuity, not taking into account details away from the fovea. A complete 360° view of the stimuli can compensate for the falloff of visual acuity in the periphery of the retina, and provide a more accurate comparison. 360° rotating stimuli were therefore used in our evaluation experiments. A major difference between our experiments and those of Pan et al. [PCB05] is that we related scale with viewing distance and depth threshold, while they did not consider viewing distance.

### 5.2.2 Luminance and Highest Visual Sensitivity

Luminance is a crucial factor in perceptual experiments. Taking the worst case scenario, we assume visual sensitivity (ability to discriminate) is the highest when comparing stimuli. In order to enforce this assumption, the experimental environment is set to (1) normal daylight, (2) a background color which has a reasonable contrast with the stimulus's color, and (3) a small spatial frequency just before the Contrast Sensitivity Function falls off sharply [LH01], imposing a condition for high brightness adaptation and visual sensitivity. In addition, we use stimuli with plain or simple texture pattern to avoid texture masking.

### 5.2.3 Judging Technique

The naming time technique [WFM01] is useful in studying visual latency (time between receiving a signal and taking an action), but its accuracy is not clear when evaluating visual fidelity, because judgement is likely to be affected by:

- Prior knowledge of the stimulus.
- Feature masking.

In other words, the time required to recognize and name the stimulus, does not necessarily reflect the quality of the simplified mesh.

The fact that the brain is more efficient in atomic operations can be taken into account in perceptual experiments. Expressing preference by selecting between two or more stimuli, or by assigning a priority rating to a stimulus, have been used in the past and proved to produce satisfactory results [PCB05] [TPA98].

### 5.2.4 Short Term Memory

Classifying stimuli into categories, such as animal and scenery, can divide a perceptual problem into sub-problems; however, presenting similar geometry and texture in consecutive tests can easily confuse judgement. The fact that a strong stimulus lasts for a short time will affect the next discrimination process. Stimuli should be randomly selected from different categories, and displayed in a random order.

## 5.3 Tradeoff between Geometry and Texture

Texture can have the following representations:

- Color per vertex – Each 3D vertex is associated with a color value.
- Synthetic texture – Texture designed for fast interactive rendering, usually with uniform and reproducible patterns [Tur91].
- Photo-realistic texture – Original texture of the object, such as that in museum exhibits and medical images, where alteration is not acceptable.

Both synthetic and realistic texture can be implemented using a texture mapping technique. Cohen found texture mapping less restrictive than color per vertex

[COM98]. A disadvantage of using color per vertex is the limited resolution. An initial color is stored with a vertex, but inter-vertex color has to be interpolated for higher resolution [SGR96]. In addition, the color value must be recomputed if a 3D vertex is shifted during simplification. Synthetic texture can be estimated or duplicated easily. For example, the pattern of the next frame can be estimated based on neighboring data [CMF99]. Synthetic texture file size is usually kept small to facilitate transmission and rendering [COM98]. Although simple colors are often used in game applications for fast interactivity, photo-realistic texture is essential to add realism to a virtual scene. Interesting scenes are often complex and diverse, and cannot be generated by replicating a small patch of pattern. In applications where real life texture is required, every pixel value of the texture has to be recorded, without interpolation. In other words, realistic scenes are likely associated with high resolution texture files, far higher than the density of the underlying mesh. High resolution texture is emphasized in virtual reality systems, where the vertical dimension of each display panel is approximately a thousand pixels in resolution, but the texture image used is often over five thousand pixels vertically in order to provide zoom-in detail. A higher resolution texture can provide the illusion of detail even with a lower resolution mesh [KTL*04].

Experiments were conducted to study under what conditions and what combinations of geometry and texture resolution can be used, without degrading visual quality [RRP00]. The authors used different levels of geometry: smooth. medium, and very simplified. They found that the perceived quality was roughly doubled by doubling the geometry of the smooth sphere, independent of texture. Quality dropped systematically for each decrease in geometry on the medium simplified sphere, but adding texture provided a significant improvement on the perceived quality. On the most simplified sphere, additional texture did little to improve the quality. There are several improvements that can be made in these experiments. First, a sphere was used as the visual stimulus, which would likely lead to a biased judgement due to prior knowledge of the object. The scores may have been different if a more general 3D object, *i.e.*, the armadillo character described in [COM98], had been used as the stimulus. Second, the objects used were of uniform color and surface finish, and therefore the result cannot be applied to 3D objects in general. The spatial frequencies generated from a uniform and a non-uniform color and surface finish, have very different impacts on the HVS. It is necessary to use different types of objects, and conduct additional tests in order to obtain a more objective and accurate conclusion.

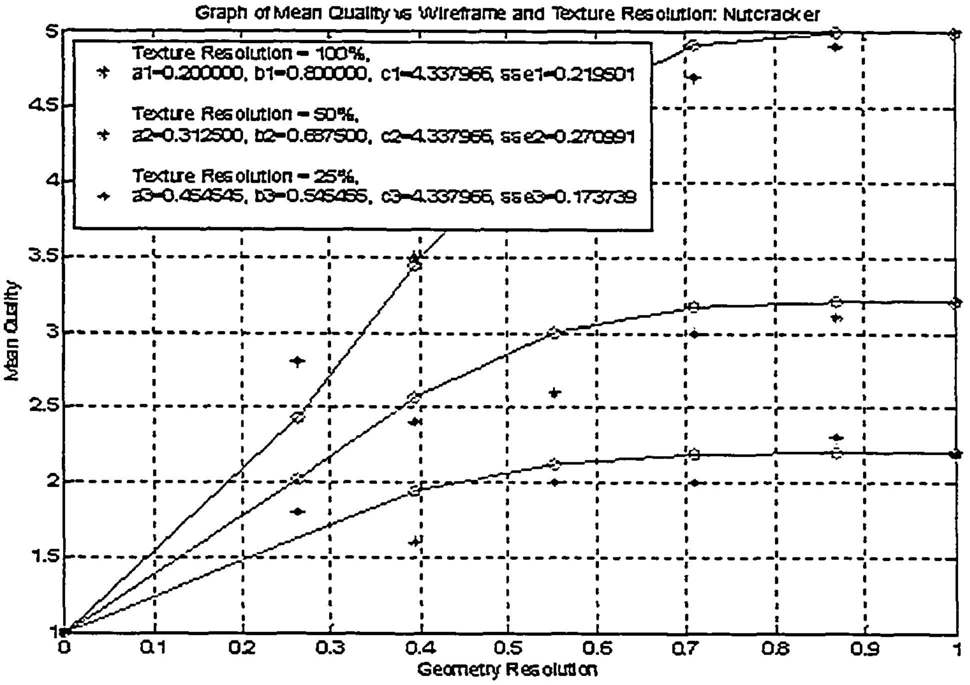


**Fig. 5.2:** *Perceptual experiments show that visual quality relates to geometry resolution exponentially* [PCB05].

Instead of a sphere, three 3D objects were used as visual stimuli, and the number of judges was increased to twenty in [PCB05]. Experimental results show that, after reaching an optimal mesh density, increasing geometry has little effect on visual fidelity (Fig. 5.2). However, additional texture relates linearly to improved quality (Fig. 5.3).

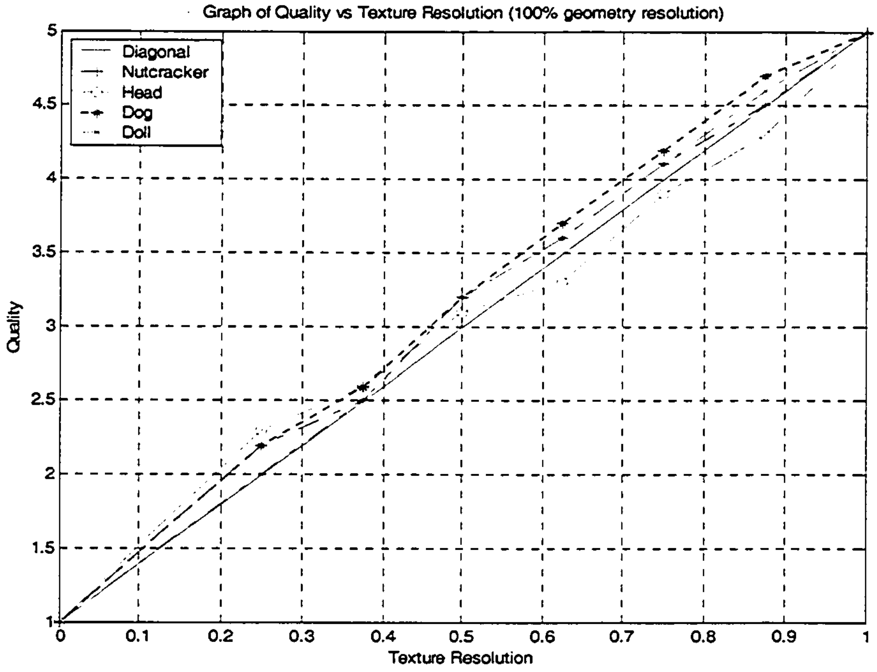


**Fig. 5.3:** *Perceptual experiments show that visual quality relates to texture resolution linearly* [PCB05].

For highly simplified geometry, increased texture resolution had significant impact on perceived quality. Based on these findings, it appears that visual quality can be improved by enhancing texture on all levels of geometry. Although a denser mesh and a higher resolution texture pattern present better fidelity in general, given limited network resources, a trade-off between quality and bandwidth is necessary in online applications. It is beneficial to transmit the mesh, which is small compared to the texture, and focus on reduction of texture.

In Chapter 3, it was discussed how variable texture qualities are assigned to fragments based on feature point density. Given a time limit which is inadequate to transmit the full resolution object, the question becomes which region (fragment with high or low frequency), should be selected for quality reduction, in order to have less impact on visual quality. The technique presented here is motivated by the observation that minor variations in texture can be ignored in relatively smooth regions of a 3D surface, without significantly affecting human perception. This observation is supported by the Contrast Sensitivity Function and the perception of depth, and is discussed in the next section.

## 5.4 Feature Point Density, Perception of Depth and Contrast Sensitivity

One of the many cues for visual depth perception is shade and sharpness, which is represented by changing frequency, on the texture surface [Nag84]. On a surface with high feature point population, the texture has higher frequency due to shade and contrast. Low frequency is less affected than high frequency by quantization in the compression process, resulting in the more obvious degradation on the grenade model (Fig. 3.8 (c) & (d)). Note that the textures of the nutcracker and grenade model are quite simple. If the objects have more complex texture, texture masking will make the degradation less obvious. To optimize the perception of depth given limited bandwidth, higher quality is assigned to higher feature density fragments. Higher qualities in these fragments also provide a cushion effect, to prevent the quality dropping too far below the default quality $Q_i$, thereby causing significant visual degradation.

Good brightness adaptation is achieved for a small value of Weber's ratio ΔI/I, where ΔI is the increment of illumination discriminated 50% of the time, with background illumination I. By plotting the log of Weber's ratio against log I, it is observed that brightness discrimination is poor at low levels of illumination, and improves significantly as background illumination increases [GW02]. The effect of illumination on vision depends not only on the light source, but also on the reflectance of the object. The contrast appearing on the texture is thus a predictor of visual fidelity. Under normal light source, visual discrimination is better on brighter than it is on darker texture patterns. How visual sensitivity relates to contrast and spatial frequency is illustrated in the Contrast Sensitivity Function [LH01].

## 5.5 Visual Quality Prediction (VDP)

In order to achieve visual quality, it is more efficient to employ a technique to determine quality while the textured mesh is generated, instead of relying on assessment and correction after it has been completed. Therefore, visual quality predictors should be available to an algorithm during runtime. Since computing these predictors online is expensive – resulting in lengthy computation time which is not acceptable for interactive applications – the solution is to collect statistics during preprocessing, which can be retrieved in constant time to support online assessment.

Feature point density is a visual quality predictor in the TexMesh model, and contributes to the determination of quality for each texture fragment. It also induces shade and contrast in the texture pattern. In addition to the brightness on the texture surface, as discussed above, texture complexity also contributes an important factor to perception, in terms of texture masking. In Chapter 9, different weights will be assigned to feature point density and texture properties, and these visual predictors will be integrated into the TexMesh model.

## 5.6 Two-Alternatives-Forced-Choice (2AFC)

In the two-alternative-forced-choice (2AFC) procedure, a subject is presented with two stimuli, *A* and *B* (which represent, for example, $x$, $x+\Delta x$). The stimuli may occur in successive intervals, or they may occur in adjacent locations [BKT86 2-39]. For the purpose of this thesis, we take the latter case, where *A* and *B* are placed in adjacent locations. The subject's (judge's) task is to state whether the target occurred in the left or the right location. In the experiments conducted for this thesis, a target is also displayed above the two stimuli so that the subject has a reference of the original object. The subject's decision is recorded, as either correct or incorrect, for each pair of stimuli. Once the judging is completed, the results are summarized. To find the visual threshold, the percentage of correct judgments is plotted as a function of $\Delta x$, which will be illustrated in Chapter 7. To avoid response bias, sufficient evaluations, *e.g.*, 30, should be collected for each $\Delta x$ value. The line of best fit is obtained by regression, and the threshold can be located at the 75% correct performance [Weber05]. When the two stimuli are clearly distinguishable, the score is 100%. If the difference is not apparent to the judge, he/she is forced to guess, and the possibility of picking the correct stimulus is 50%, after a sufficient number of evaluations have been performed.

To prevent a subject from spending too much time on an evaluation, a time limit of 20 seconds is imposed in the experiments. The display is frozen after the time limit and the subject is forced to make a choice, guessing if necessary. It should not be assumed that the subject is unable to distinguish the stimuli, because he/she may have a good idea of which one is better; but, if given unlimited time, the subject often looks for more evidence to support their decision. Forcing the subject to respond will result in their making a decision. If the subject really cannot distinguish the stimuli, he/she will choose one alternative, with 50% correctness (guessing).

## 5.7 Linear and Non-Linear Mechanism in Visual Perception

When performing prediction or statistical inference using perceptual data, we can apply either a linear or a non-linear approach. Although the visual system is demonstrably nonlinear in many circumstances, a linear approach can be used when a nonlinear signal passes through independent detection pathways [BKT86 6-9]. In this case one pathway can be linear, but the overall system is non-linear. When restricting data in a particular pathway, a linear model can be applied. Nonlinearities in the visual system may be well approximated locally by a linear function.

Abbey et al. [AE02] suggested that the linear approach is a useful starting point. Their experimental results showed that the classification image is closely related to a linear observer. Their survey also showed that linear models have a history of use for modeling human-observer performance in noise-limited simple detection and discrimination tasks.

## 5.8 Conclusion

Chapter 5 presents the main environmental and psycho-visual factors that can affect the perceived quality of 3D objects. Some factors, *e.g.*, temporal visual masking and velocity, are more important when dealing with dynamic than with static scenes. Latency is important for video in order to ensure a smooth transition between frames. Since the TexMesh framework is designed for relatively static 3D textured meshes, the focus is on having those factors which are likely to influence the perception of geometry and texture, appear in a spontaneous and not sequential manner. Perceptual experiments will be discussed in Chapter 6, Chapter 7 and Chapter 8. The goal is to understand more about biological vision, in order to integrate that knowledge with the proposed framework.

# Chapter 6

## Perception of Scale with Viewing Distance – A Step Function

In this thesis, viewing distance is defined as the distance between the 3D object and the viewpoint in the virtual world, which is different from the physical distance between the display device and the retina. LOD as perceived by the HVS varies as a function of the viewing distance. In order to support automatic selection of scale given a viewing distance, a scale-distance function has to be established. It is understood that details become less visible and eventually vanish as an object moves towards infinity. In this chapter, results from perceptual experiments are analyzed and explained based on a scale-space filtering approach. In the experiments, a graphical interface was used to display a pair of 3D objects (visual stimuli) at different simplification scales, and the judges were asked to locate a relative position at which the stimuli appear visually indifferent. Experimental results demonstrate that scale relating to distance is not linear, not exponential, but follows a step function.

Perceptual evaluation experimental results can be affected by various environmental and psycho-visual factors. A detailed discussion of these factors was presented in Chapter 5. The experiments were carried out in a laboratory with indoor lighting. The visual stimuli were 360° view-independent texture-mapped 3D objects and the illumination was uniform in the entire scene. Texture resolution was the same on both stimuli. While the position of one stimulus was fixed, the judge was asked to move the other stimulus closer or farther away, in order to locate a viewing distance at which both stimuli appear visually similar. Viewing distance is measured from 0 to –20 with 0 being the closest distance. This approach is different from the rating technique used by Pan et al. [PCB05]. While they fixed the viewing distance to compare the effect of geometry and texture resolution on perceptual quality, the perceptual evaluations discussed in this thesis focus on how scales relate to distances. Since illumination was kept uniform, the contrast sensitivity was the same on both stimuli. Visual acuity is significantly higher at the fovea than in the visual periphery [RV79], and it is an important factor when considering gaze-directed perceptual evaluation. However, in our experiments the 360° rotating 3D objects are view-independent, balancing the visual acuity between the fovea and its periphery after an object has rotated a complete cycle. Until recently [PCB05], perceptual comparisons in the literature have focused on static display, which shows only a limited number of silhouettes. By contrast, using a 360° view allows all silhouettes to be examined. In the following, let us first examine the convergence property of SSF, and see how the convergence property can be used to explain the results of the perceptual experiments.

## 6.1 Convergence Property of SSF

By applying scale-space theory, the number of feature points (structures in the sample space) decreases as scale level increases. This concept best describes how objects are perceived by the HVS when they move from close to farther away. Using SSF, different LOD can be generated by varying the $\sigma$ value in Equation (2.4). However, different 3D models have different surface structures and thus the maximum $\sigma$ value ($\sigma_{max}$) applied to achieve a sufficiently smoothed surface is different. To determine $\sigma_{max}$, the rate of convergence was recorded while performing SSF. A set of 360 sample values was extracted from each of the head and nutcracker models (Fig. 6.1 a & b) to analyze the convergence rates (Fig. 6.2). Every one of ten values was taken from the 360 sample values to generate the 36 sample sets (Fig. 6.1 c & d).

| 3D model | Number of sample values |
|---|---|
| (a) head | 360 |
| (b) nutcracker | 360 |
| (c) head | 36 |
| (d) nutcracker | 36 |

**Fig. 6.1:** *Two sets of samples (360 and 36 scan-points) extracted from each of the head and nutcracker models respectively.*

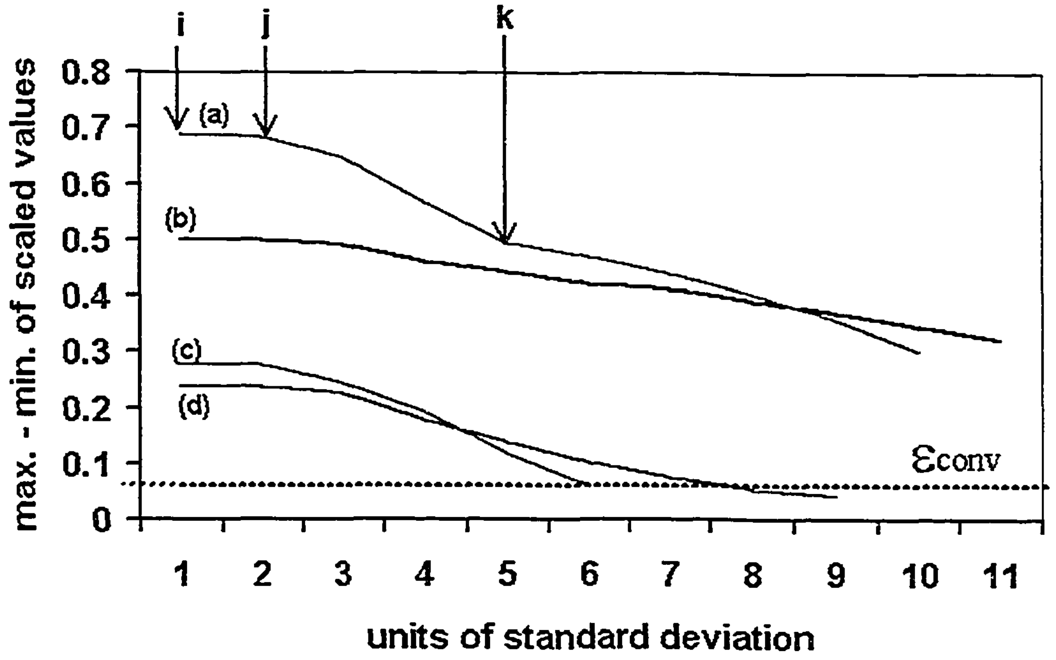


**Fig. 6.2:** *Convergence property of SSF – the difference between the maximum and minimum scaled values diminishes during the smoothing processes as i increases.*

For increasing values of $\sigma$ on the horizontal axis, the vertical axis represents the difference between the maximum and minimum of the scaled values (Fig. 6.2). In other words, a difference of 0 means 100% smoothing. Let us define a convergence threshold $\varepsilon_{conv} > 0$. There are two observations that can be made here:

(1) When the number of sample values increases, $\sigma_{max}$ also increases. Samples (a)

and (b) contain 360 values and require a higher $\sigma_{max}$ to reach $\varepsilon_{conv}$ than samples (c) and (d).

(2) Different models have different surface structures and require different $\sigma_{max}$ to reach $\varepsilon_{conv}$.

When displaying 3D objects in the virtual world, viewing distance is defined from 0 to infinity where an object vanishes. For each object, the scales generated by value $\sigma \in [0, \sigma_{max}]$ are then matched with a corresponding distance in the range $[0,\infty]$. In the following implementation, we used 20 scales in addition to the original signal to cover this range, where $S_0$ corresponds to $\sigma = 0$ and $S_{max}$ $(S_{20})$ corresponds to $\sigma_{max}$.

## 6.2 Perceptual Evaluation Experiments and Analysis of Results based on the HVS

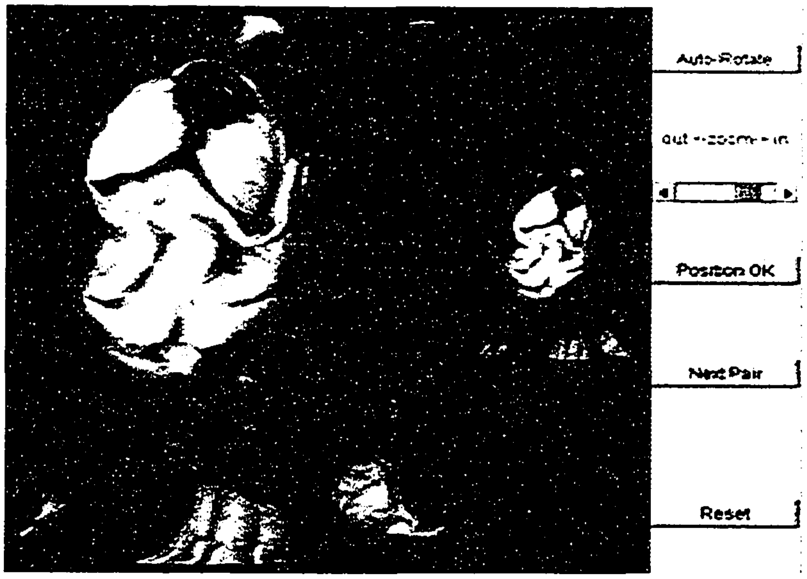


**Fig. 6.3:** *Rotating visual stimuli are used to conduct perceptual experiments relating scale to viewing distance.*

In general, a simplified version of a 3D object needs to be refined when it gets closer to the viewpoint, in order to preserve visual fidelity. In this section, I will use perceptual experiments, fixing the texture resolution and illumination on the stimuli, to analyze how the HVS responds to changing scale – whether the function relating scale to viewing distance follows a linear, exponential or a more complex pattern.

In these experiments, the judges had to decide whether the more simplified version (Fig. 6.3 right) was perceptually similar to the less simplified version (Fig. 6.3 left). If not, they moved the right stimulus farther away until it appeared visually similar. If yes, the judge decided on the closest distance that the right stimulus could be placed without noticeable degradation. Both stimuli were rotating slowly in synchronization, so that the judges could compare a complete 360° view. For each of the five models (nutcracker, head, dog, grenade and vase), a number of simplified versions were generated in a randomized fashion. Preliminary results suggest that, for each model, the relationship between distance and scale follows a step function. In Fig. 6.4, the *x*-axis is the scale and *y*-axis is the distance with zero being the closest distance, which increases in the negative direction.

The step function is divided into alternate red (perceptible) and green (imperceptible) zones (Fig. 6.4), with the red zone corresponding to the slope and the green zone corresponding to the comparatively flat portion. The convergence property suggests that the surface of a 3D object gets smoother as more feature points are removed. In the red zone, removal of perceptually important feature points from a mesh causes a noticeable degradation in visual quality. While in the green zone, feature points eliminated do not have significant perceptual impact. For example, at distance D, decreasing the scale from B to A does not improve perceptual quality, while increasing from scale B to C creates a much more significant impact. For each model, the step function is unique, containing different number of steps depending on the surface structures of the model. The step function suggests that perceptually less important data can be bypassed during simplification or refinement, so that we can focus on the major scales (all those in the red zone excluding the green vertical line) when selecting LOD.

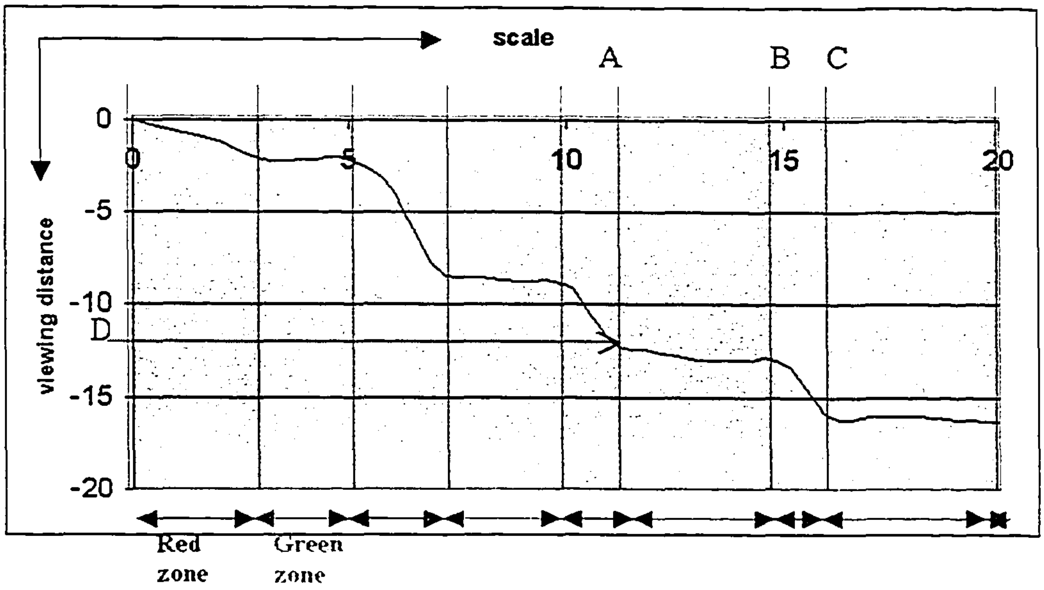


**Fig. 6.4:** *Preliminary results show that perception, relating scale to viewing distance, is close to a step function.*

Given a viewing distance *d*, we locate the corresponding zone and extract the simplified version of the object based on the major scale. If *d* falls into a green zone, the rightmost scale is selected, because using other scales (minor scales or denser mesh) in the zone does not improve visual quality. If *d* falls into a red zone, the corresponding scale is selected because each scale in the zone is a major scale.

The step function can be explained by the convergence property of SSF shown in Fig. 6.2. We already noted that objects with more scan-points (samples a & b) take longer to converge to a predefined value $\varepsilon_{conv}$, but more importantly, for each sample, the convergence rate is not constant. Since the human visual system is insensitive to minute changes below a certain threshold, such as the convergence rate between $i$ and $j$ (Fig. 6.2 a), refining the scale from 2 to 1 will not have significant impact on the visual quality. However, the convergence rate between $j$ and $k$ indicates a significant improvement on visual quality when refining the scale from 5 to 2.

Based on these preliminary findings, more precise perceptual experiments were conducted, which will be discussed in Chapter 7. The number of judges and models were increased to further verify the step function and the convergence property of SSF. A perceptual metric will be introduced to determine the step function of each 3D object. Another issue that will be addressed is how to take advantage of complex texture patterns to compensate for low geometry, and whether the compensation can reduce bandwidth usage. A visual quality prediction (VQP) model, analyzing both 3D and 2D surface properties, will be discussed in Chapter 8.

## 6.3 Conclusion

In this chapter, we discuss the convergence property of SSF. Perceptual experiments were performed to relate scale to viewing distance. Experimental results show that the relationship follows a step function, which can be explained by the convergence rates when applying SSF on 3D surfaces. Since different 3D objects have very different surface structures, a perceptual metric is required to determine the step function associated with each object in order to support automatic selection of scale at a given viewing distance and mesh refinement. How to formulate this perceptual metric will be discussed in Chapter 7.

# Chapter 7

## A Perceptual Metric for Mesh Refinement based on Just-Noticeable-Difference (JND)

When transmitting 3D textured mesh (TexMesh) over a shared network, limited resources such as bandwidth has to be allocated between both mesh and texture data. Progressive refinement strategies [Hop96] [KSS03] often assume that visual quality improves as the mesh resolution increases, ignoring the experimental finding that texture resolution has more significant impact on quality after the mesh resolution has reached a certain threshold [PCB05] [RRP00]. Geometric metrics were commonly used in previous simplification techniques [HH93] [GH98]. However, perceptual metrics [OHM*04] have been gaining increased attention among researchers for two main reasons: First, visual quality is ultimately determined by the Human Visual System (HVS), and thus using perceptual metrics is expected to be more accurate. Second, assessment relying on geometric criteria, such as mean square error (MSE) or quadric error [GH98] is not sufficient because geometrically different objects can be visually indistinguishable to the HVS. Transmitting redundant mesh data without improving visual quality is a waste of resources [CB04].

In this chapter, we present a mathematical model to measure the perceptual values associated with 3D vertices, which will be used to predict the benefit to visual quality when refining a coarse mesh to a denser version. In order to maximize the overall quality, the server decides, based on the statistics gathered during preprocessing, whether mesh refinement should terminate, allocating the remaining bandwidth to increase texture resolution. This can be achieved by locating a perceptual threshold (Just-Noticeable-Difference or JND), where the HVS can just distinguish the difference between two levels-of-detail (LOD). In the following, the JND was determined and verified by conducting perceptual evaluation experiments with texture mapped on to the mesh. Texture mapping was used for two reasons: (a) it is easy to visually identify differences in mesh only, and (b) one of the goals of this thesis is to optimize the perceptual quality of photo-realistic 3D objects given bandwidth limitations. Online transmission of 3D TexMesh can then be more efficient, by suppressing imperceptible geometric data, which have dimension below the JND.

Scale-Space Filtering (SSF) is used to extract 3D features (Chapters 2 & 3). Traversal between the different scales is achieved by varying the standard deviation ($\sigma$) value of the Gaussian filter; the higher the value of $\sigma$ the more is the smoothing [Wit83] [KF01]. Decimation and refinement are performed using edge collapse and vertex split operations. A detailed discussion of various mesh simplification approaches can be found in [Lue01]. There are two main differences between my edge collapse/vertex split and that used in progressive meshes

[Hop96]: (1) There is no vertex relocation between different LOD in the TexMesh; all vertices at a coarse level is a subset of those at a finer level. (2) In progressive meshes, the minimum energy cost, recalculated each time a new vertex is introduced in an edge collapse operation, affects the choice of the next collapsing edge. In the TexMesh model, the order of collapsing edges follows the priority predetermined by applying SSF on the original 3D surface. A vertex is removed by integrating it with its closest neighbor, collapsing the edges associated with it.

In related work, Reddy approximates the contrast sensitivity function (CSF) in dynamic scenes to optimize the amount of detail removed from the scene without the user noticing [Red01]. The velocity of the object is taken into consideration. By contrast, the method proposed in this thesis is designed for comparatively static 3D objects. CSF is also used to derive perceptual metrics in order to measure the perceptibility of visual stimuli [LH01]. In their approach, only simplification operations inducing imperceptible contrast and spatial frequency are performed. However, the technique is not adequate to suppress perceptually redundant data. Williams *et al.* [WLC*03] improves prior approaches by accounting for textures and dynamic lighting. However, the above techniques are view-dependent, while the TexMesh approach is view-independent. In addition to the reduced navigation costs associated with view-independent algorithms, the proposed perceptual model provides a systematic way, instead of heuristics, to predict visual fidelity. The JND defined in the mathematical model follows the same spirit as Weber's Law on contrast, computed as *the change relative to the original value*. Perceptual experimental results described later confirm that JND in the 3D TexMesh framework is a constant and is independent of the viewing distance.

## 7.1 Weber's Law and JND

The JND is the minimum amount by which the stimulus intensity must be changed in order to be noticeable to human sensation or perception [Weber05]. When a stimulus value $x$ is examined in the TexMesh framework, we are interested in the smallest change $\Delta x$ such that $x + \Delta x$ is "just detectable" by a subject or judge. A 19th century psychologist, Ernst Weber, observed that the ratio between the JND and the original stimulus value appeared to be a constant, which is known ever since as Weber's Law. Weber's Law is often associated with psycho-visual experiments on contrast sensitivity. For example, if two light sources of 100 units each are presented to an observer. Then one of the light intensity is randomly chosen and increased gradually. At each increment the observer is asked to identify the brighter source. Suppose at the point when the observer can just identify the brighter source is after an increment of 10 unit, the JND would be 10/100 = 0.1. Based on the JND and given the original source of 1000 units, the increment required to generate a noticeable perception is 1000 x 0.1 = 100 units.

Weber's Law can be applied to a variety of sensory and perceptual aspects, including brightness, loudness, mass, line length, etc. Fig. 7.1 shows how the perceptual impact of a change is relative to the original stimulus' magnitude for line length.

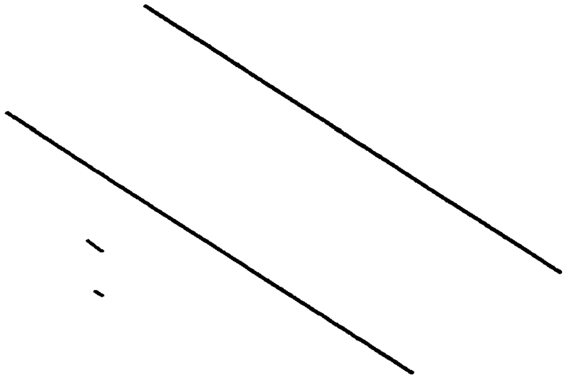


**Fig. 7.1:** *An example of perception on line length. The absolute difference in the two lines in each pair is the same.*

A group of 60 judges were asked which of the two black lines in Fig. 7.1 (right) is longer. One third of the judges chose the upper one, another one third selected the lower one and the rest said they were equal. When asking the same question on the pair of blue lines (Fig. 7.1 left), every judge identified the upper one as longer than the lower one. Notice that the absolute difference for both pairs of lines is the same. However, the relative difference is much bigger in the blue pair than in the black pair, which makes it perceptually more difficult to compare the black line lengths. It is interesting to note that the viewing angle relative to the stimulus plays an important role in perceived similarity [SW04]. Let $\theta$ be the angle between the line direction and the line of sight. When $\theta = 0°$ both lines appear as a dot to the viewer and discrimination of line length is impossible. Suppose $\varepsilon$ is the difference in line length. As $\theta$ increases, the projection of $\varepsilon$ onto the retina also increases and it generates a maximum projection when θ = 90°. Since the TexMesh framework is designed for view-independent manipulation of 3D objects, maximum projection (perceptual impact) is considered.

Weber's Law was also tested for surface curvature discrimination [Joh94]. In their experiments, curvature discrimination thresholds were measured as a function of the curvature of the main body of the sphere, which was computed as the inverse of the radius. There was an approximately linear increase in discrimination threshold with curvature for all three subjects. The JNDs were in the range 0.08 to 0.17 with a mean value of 0.11, which compares well with the JND of around 0.1 found by the author in 1991 for a curvature discrimination task in which cylinders defined by binocular disparity were used [Joh91]. In the surface curvature discrimination experiments, Johnston used the slope of the regression line to locate the JND and found it relatively constant as a function of curvature. He concluded that from a JND of 0.11, it can be calculated that when the physical distance between the display and the retina is 75 cm (29.6 in), the HVS can discriminate a change of 3.7 mm in the radius of curvature of a surface patch on a 7.5 cm diameter sphere.

The outcome from visual discrimination experiments based on Weber's Law means that stimuli of decreasing magnitude relative to the original stimulus of a fixed dimension are difficult to discriminate. This is consistent with the perceptual experimental results [PCB05] showing that after the mesh has reached a minimum required resolution, further increase in mesh resolution does not have significant perceptual impact. This is because the refined triangles are getting smaller and smaller. In any given viewing direction when a 3D object is projected onto a display device, it is visualized as a 2D shape. The silhouettes define visible surfaces which generate different degrees of impact on the HVS during mesh refinement. A perceptual value can be computed by comparing the visible surfaces between two scales of detail. An image-based edge cost approach was introduced for determining the visual similarity between an original and a simplified model [LT00]. Their edge cost measure is based on the mean square error (MSE) between the two projected images. Since complex visual shapes are represented in terms of distributed collections of parts which are processed independently in visual search [AS04], computing the cumulated MSE without taking part segments of a 3D object into consideration is not consistent with the HVS.

In the next section, I will extend Weber's Law to perceived similarity on 3D TexMesh, segregating an object into parts in visual search as processed by the HVS.

## 7.2 Perceptual Value and JND

When a 3D object moves closer to the viewpoint in a virtual scene, the mesh needs to be refined only if the resulting mesh improves visual quality. To determine whether mesh refinement should be performed requires measuring perceptual impact on the HVS. Adding or deleting a vertex or surface structure from a mesh generates a stimulus to human vision. To compare the perceptual impacts of these stimuli, the dimension of a structure is used as a visual cue in the model. Lets follow the argument that humans naturally describe an object as consisting of parts and infer the projected 2D shapes of these parts [AS04] [ZN99], and segment the object into corresponding parts — skeletonization [PK04] [PSB*01] [SZZ01] [Skel05]. Skeletonization involves a thinning or boundary peeling algorithm which literatively peels off the boundary layer by layer by identifying and removing the simple voxels with additional conditions (for example the end point condition) [PK04]. Detail on skeletonization is available in the literature and will not be discussed further.

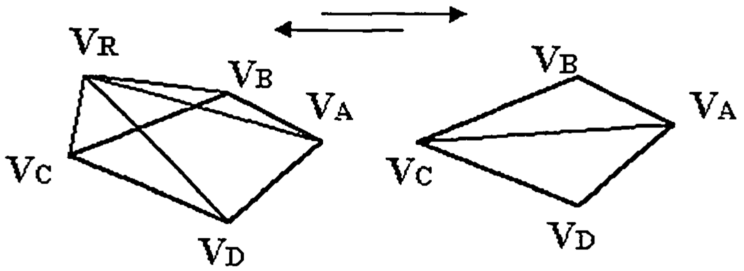
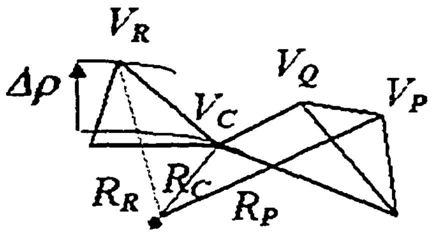


**Fig. 7.2:** *An example of vertex removal: (Left) denser version before $V_R$ is removed and (Right) coarser version after removing $V_R$.*

In each edge collapse operation during preprocessing, when a vertex $V_R$ is removed and integrated with its closest neighbor $V_C$ (Fig. 7.2), we record the surface change as the difference $\Delta\rho$ between $R_R$ and $R_C$. $R_v$ is the shortest distance between vertex $v$ and the skeleton. For a spherical object, the skeleton is represented by the center of the object (Fig. 7.3). $\rho_R = (R_R-R_C)/R_C$ is defined as the *perceptual value* of $V_R$. If edge $V_QV_C$ collapses after $V_PV_Q$, the perceptual value of the combined operation is $(R_P - R_C)/R_C$. Our model is designed for view-independent simplification. In a given view, when a 3D object is projected onto a 2D display, the stimulus can be interpreted by Weber's fraction on shape. Also, note that visual impact of a stimulus is dictated locally by the closest adjacent vertex and the closest distance to the skeleton. For example, collapsing $V_RV_C$ has higher impact than collapsing $V_QV_C$, and we can disregard the overall shape and dimension of the object.



**● Center of 3D object.**

**$R_R$, $R_C$ & $R_P$ are radii of vertex $V_R$, $V_C$ & $V_P$ respectively.**

**$\rho_R = (R_R - R_C) / R_C$.**

**$\rho_P = (R_P - R_C) / R_C$.**

**Fig. 7.3:** *$V_R$ and $V_P$ have perceptual values $\rho_R$ and $\rho_P$ respectively.*

Let $\Delta\wp$ be the change when removing $V_R$ and $\wp$ be the distance of $V_C$ from the skeleton. When viewed on the display device, the difference $\Delta\wp$ generates a stimulus to the retina (Fig. 7.4). The Just-Noticeable-Difference (JND) is the minimum change in perceptual value in order to produce a noticeable variation in visual experience. Weber's Law [GW02] states that at the JND threshold,

$$\frac{\Delta \wp}{\wp} = K \qquad (7.1)$$

where $K$ is a constant. A value greater than $K$ generates a significant perceptual impact on the HVS. In this thesis, we extend Weber's Law to evaluate perceived similarity on 3D TexMesh. Instead of representing the stimulus linearly, an alternative is to use the area of the quadric error generated by removing $V_R$. Experimental results show that this perceptual metric predicts visual quality well, closely following human perception.

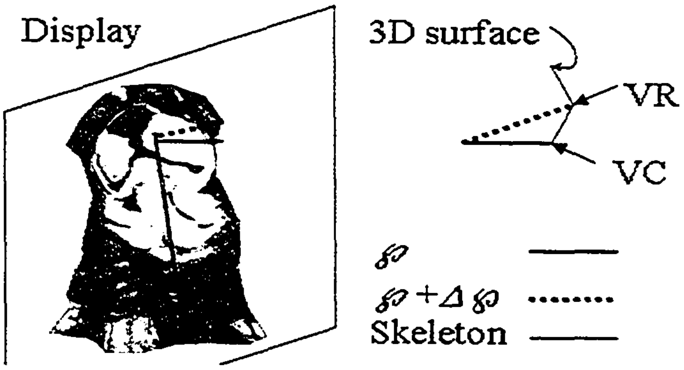


**Fig. 7.4:** *An example of perceptual impact generated by the removal of vertex* $V_R$.

Notice that the discussion so far assumes that the viewer is working with a desktop or laptop computer, and the entire 3D object is displayed within the fovea region. If a partial object is displayed, the original shape is clipped and the skeleton needs to be adjusted accordingly using the boundary of the display device to form the shape of the object (Fig. 7.5 left). In this case the skeleton is closer to the visible surface. Stimuli which are not obvious before may become noticeable.

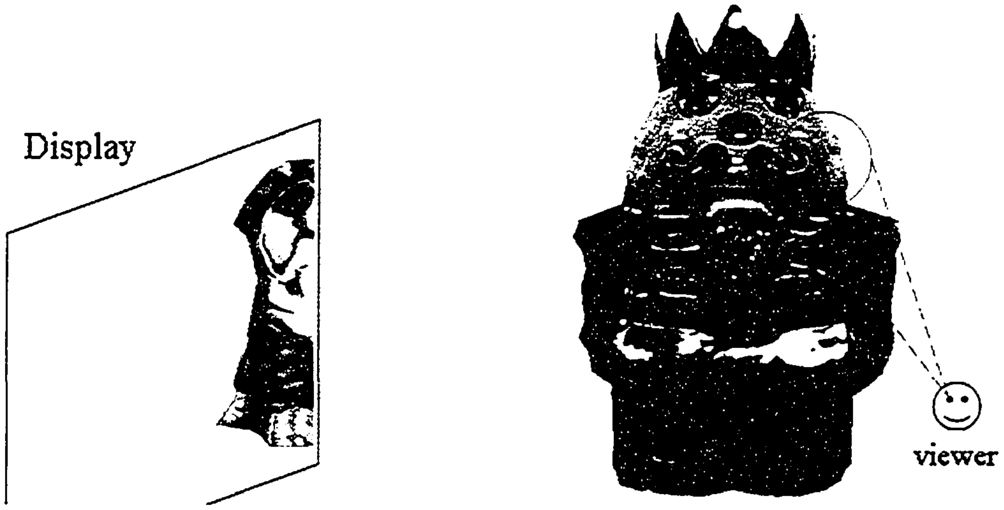


**Fig. 7.5:** *(Left) The boundary of the display device is used to form the shape if only a partial object is displayed, and (right) if the displayed object is too big the visible surface for computing perceptual values is restricted by the fovea region.*

Suppose the working distance in front of a desktop or laptop is 18 to 30 inches. If an object is displayed 10 times bigger on a bigger screen, the working distance is expected to be at least 180 inches in order to use the original skeleton. If the viewer is too close to the screen (Fig 7.5 right) the fovea can only cover a part of the object. In this case the skeleton should be generated corresponding to the fovea region instead of the entire object.

Online applications have benefited from the high speed communication infrastructure in recent years. Meshes with a few hundred triangles do not create much problem in term of latency and frame rate. The focus is therefore on suppressing redundant data from high resolution meshes composed of thousands of triangles. Perceptual impacts can be generated between change of scales or when individual vertices are inserted. In either case, the perceptual value of each vertex inserted into the refined scale is recorded. Based on SSF discussed in Chapters 2 and 3, the dimensions of the inserted 3D structures decrease towards finer scales, but more importantly structures of similar size are grouped between adjacent scales. If we take a conservative approach, the maximum perceptual value among these structures should be used assuming the viewer can detect the maximum impact. An aggressive approach would be to take the minimum impact. In the experiments, an intermediate strategy was adopted using the average value. The average perceptual value is a good estimate provided that the standard deviation is small.

To verify the perceptual metric, SSF was performed on the nutcracker object with 1260 faces at $S_0$. For each scale change, the perceptual values of the vertices removed were recorded. At each scale the average value was used to represent the perceptual impact when refining from $S_i$ to $S_{i-1}$ (Table 7.1). The cumulated perceptual values were also computed and stored in a lookup table (LUT) so that the perceptual impact between $S_\mathrm{i}$ and $S_\mathrm{j}$ can be retrieved (Table 7.2).

| Scale | # of faces | Perceptual value | |
|---|---|---|---|
| $S_{i-1}$ - $S_i$ | | Avg. | Std. |
| 0-1 | 1162 | 0.0410 | 0.0308 |
| 1-2 | 1118 | 0.0412 | 0.0294 |
| 2-3 | 1074 | 0.0478 | 0.0390 |
| 3-4 | 1040 | 0.0468 | 0.0288 |
| 4-5 | 1002 | 0.0678 | 0.0491 |

**Table 7.1:** *$\Delta\wp / \wp$ of the nutcracker mesh between adjacent scales.*

Previous refinement techniques assume that visual quality is proportional to the number of vertices. Our preliminary finding shows that not every set of vertices has significant impact on visual quality [CB04]. Note that the average perceptual value column in the tables indicates that change of scale generates stimuli of different magnitudes. Notice that the average value increases towards higher scales except from $S_3$ to $S_4$. This can happen if a large number of smaller values are present in the data set lowering the overall average. Remember that the HVS is insensitive to stimulus below a certain dimension denoted by the JND. In the next section, I will use perceptual experiments to locate and verify the JND for mesh refinement.

| From scale | To scale | Perceptual value |
|---|---|---|
| 0 | 1 | 0.0410 |
| 0 | 2 | 0.0616 |
| 0 | 3 | 0.0677 |
| 0 | 4 | 0.0759 |
| 0 | 5 | 0.1080 |

**Table 7.2:** *An example of cumulative perceptual values.*

## 7.3 Perceptual Evaluation Experiments to Estimate JND

Psycho-visual experiments were conducted to establish some thresholds for human sensitivity [ODG*03], but they explore the factors that affect the perception of dynamic events, in this thesis we will focus on relatively static objects. In the initial set of experiments, an 8" x 11" monitor of resolution 768 x 1024 pixels was used, with indoor incandescent lighting. 360° rotating objects were the visual stimuli. By using automatically rotating objects, the judges were able to examine all silhouettes, which is more accurate than selecting a limited number of views [WFM01].

Since the goal is to evaluate the visual impact resulting from geometry change, the same texture was mapped onto both stimuli under comparison. The type of texture used may affect the silhouette information. For example, checkerboards, randomly oriented lines and elongated texture elements could present additional cues to discriminate perceived similarity [Joh94] [SW04]. Thus, limited choice of band-limited random-noise textures were used, which were regular and non-oriented. Four sets of experiments were conducted. Randomly generated ellipsoids of different dimensions were used in the first two. Irregular quadrics were used in the third, and a 3D object was used in the fourth. 3D surfaces can be approximated by ellipsoids [KT96] defined by the polynomial equation with parameters *a*, *b* and *c*.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \qquad (7.2)$$

By altering the parameter values, one could easily control the dimension of the stimuli generated. Since each 3D object has its unique surface property and thus perceptual values, not every perceptual range can be found in a 3D object. For an initial estimation, it is easier to apply scaling factors on ellipsoids to narrow down the range where the JND is located.

### 7.3.1 Experiment 1 – An Initial Estimation of JND

In each test, a pair of ellipsoids (original and scaled versions) was displayed to a judge. The stimuli could be zoomed into and out of, and rotated in any direction in a synchronized manner for examination. The original version was generated by randomly selecting the parameter set $\{a, b, c\}$. The scaled version is defined by the parameters $\{fa, fb, fc\}$ where $f$ is a scaling factor in the range [0.7, 1.3]. One, two or none of the three parameters $a$, $b$ and $c$ were randomly exempted from scaling. The left and right positions were randomly assigned to the ellipsoids. The two-alternative forced-choice (2AFC) strategy [Weber05] was adopted, and judges were asked to choose the larger ellipsoid. After 34 tests with one judge, the scaling factors which could be recognized correctly 100% of the time was eliminated. We also eliminated the scaling factors for which judges relied on guessing (correctly judged approximately 50% of the time). The range was then refined to [0.9, 1.1].

### 7.3.2 Experiment 2 – Locating JND for Regular Ellipsoids

Experiment 1 was repeated within the refined range [0.9, 1.1]. Each correct or wrong answer was recorded under the ten sub-ranges $\Re_j$ ($j \in [1,10]$), corresponding to the set of values $0 < \chi_1 \leq 0.01, \ldots, 0.09 < \chi_{10} \leq 0.10$, with $f = 1 \pm \chi_j$. 439 tests were conducted with two judges. For each sub-range the percentage $\Omega_j$ for which the judge had chosen the correct ellipsoid was computed. It was noticed that in a low sub-range, $\Omega_j$ is also low, implying that it was more difficult to distinguish among the ellipsoids. Experimental results show that in the sub-range $\Re_4$, the judge could choose the correct answer 75% of the time. Thus, 0.04 was determined as the JND for discriminating ellipsoids.

### 7.3.3 Experiment 3 – JND for Irregular Quadric Surfaces

In the virtual world, 3D objects are often more complex than a smooth ellipsoid. To verify the JND for more general 3D shapes, Experiment 2 was repeated but the ellipsoids were randomly distorted to generate irregular quadrics of random dimensions (Fig. 7.6).

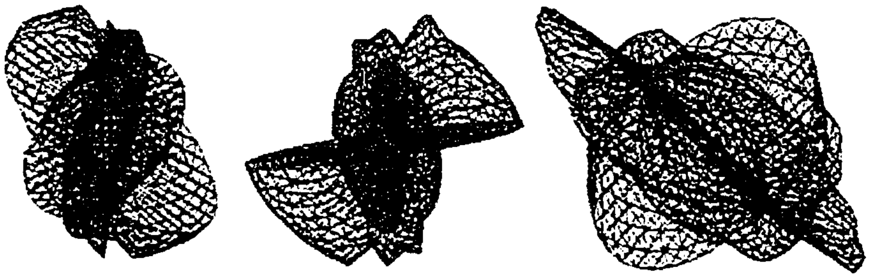


**Fig. 7.6:** *Examples of randomly generated irregular quadric.*

For each evaluation, a texture was selected randomly from six different patterns to avoid possible texture masking effect, but the same texture was mapped onto both stimuli in each pair, and indoor incandescent lighting was used in the experimental environment. Prior knowledge and familiarity are compelling factors affecting how the HVS perceive, and thus it is likely that the irregular quadrics could be more difficult than the regular ellipsoids for the HVS to discriminate. To accommodate this factor, a broader range of [0.8, 1.2] and 20 sub-ranges were used. One thousand tests were assigned to seven judges, and each judging session did not exceed 3 minutes to avoid fatigue. To ensure unbiased result. at least 30 tests were completed in each sub-range. The line of best fit was solved by regression (Fig. 7.7). A JND of 0.10 (sub-range 10) was located as the threshold where the judgment was correct 75% of the time. The experimental points were fitted by a regression line instead of a psychometric curve because in the selected interval the function appears to be approximately linear [BKT86 1-24].

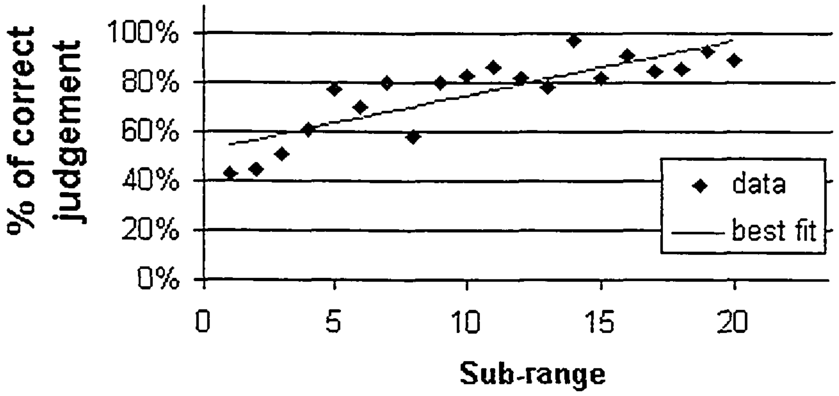


**Fig. 7.7:** *JND for quadrics based on data obtained from 1000 tests.*

Note that the JND is higher for irregular quadrics than regular ellipsoids. Since the appearance of 3D objects are close to quadrics then ellipsoids, 0.10 was used as the benchmark in Experiment 4 to evaluate the perceptual impact when refining the nutcracker object from a coarser to a denser version.

### 7.3.4 Experiment 4 – Verifying JND with 3D TexMesh

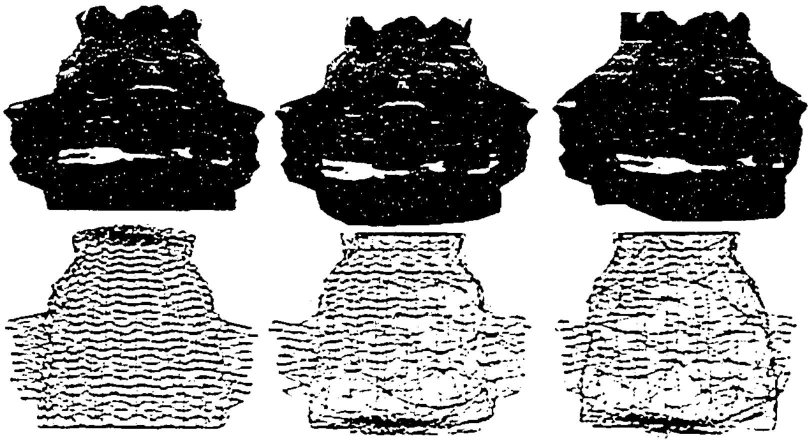


**Fig. 7.8:** *An example of different scales of the nutcracker object, $S_0$, $S_6$ and $S_8$ from left to right.*

In this experiment, the JND obtained from irregular quadric surfaces was verified by testing pairs of simplified meshes randomly selected from $S_0$ to $S_{20}$ of the nutcracker object (Fig. 7.8). The original mesh $S_0$ was displayed as a reference in the upper part of the interface. Two stimuli were displayed side by side in the bottom part. I followed the 2AFC with reference strategy, and a judge was asked to decide which one (left or right) was a finer version closer to the original. The perceptual values in the LUT (as shown in Table 1 and Table 2) were grouped into 10 sub-ranges. 361 tests were conducted by twenty judges on three monitors of different dimension and resolution, and the percentage of correct judgement in each sub-range was recorded. A threshold of 0.10 (Fig. 7.9) was obtained by locating the 75% correct judgement on the best-fit psychometric curve (sigmoid curve). The threshold is consistent with the value obtained in Experiment 3, showing that the JND perceptual metric is consistent with the HVS.

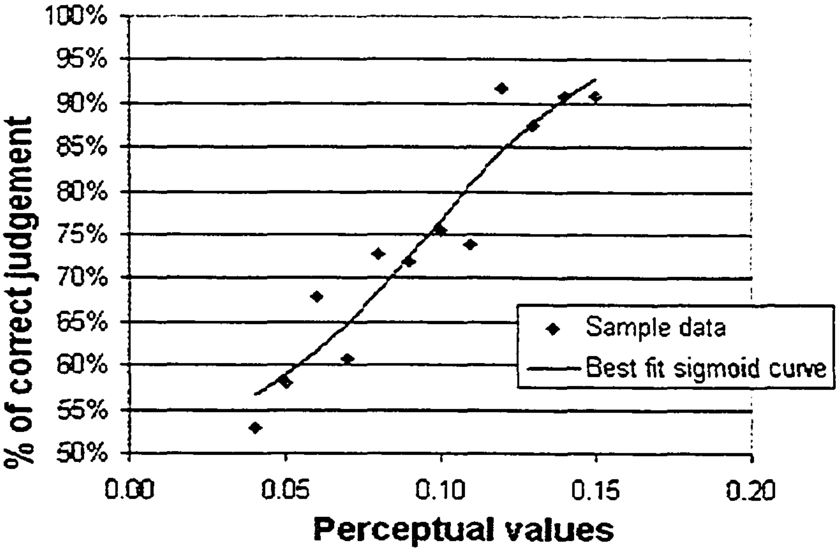


**Fig. 7.9:** *Verification of JND using the nutcracker object.*

## 7.4 Mathematical Analysis of the Perceptual Metric

In order to apply the JND perceptual metric, the following criteria have to be satisfied:

(1) Coarse meshes composed of a few hundreds of vertices are not costly to transmit and render considering the high-speed networks and advanced rendering hardware technology available. The focus is thus on reducing redundant data from dense meshes composed of at least a thousand vertices. As discussed by Pan et al. [PCB05] the visual quality benefited

from increasing mesh resolution diminishes towards higher density, and no quality improvement is detected after a certain visual threshold. It can be seen that in Fig. 7.10, (a) contains 18,720 triangles which is 10 times that of (d), corresponding to 42 and 8 KB respectively in zip format. However, when the same texture is mapped onto (a) and (d), the resulting 3D objects (b) and (c) are visually similar. The JND perceptual metric is used to determine whether further mesh refinement would improve visual quality. If not, the remaining computational and network resources should be allocated to other related multimedia data.

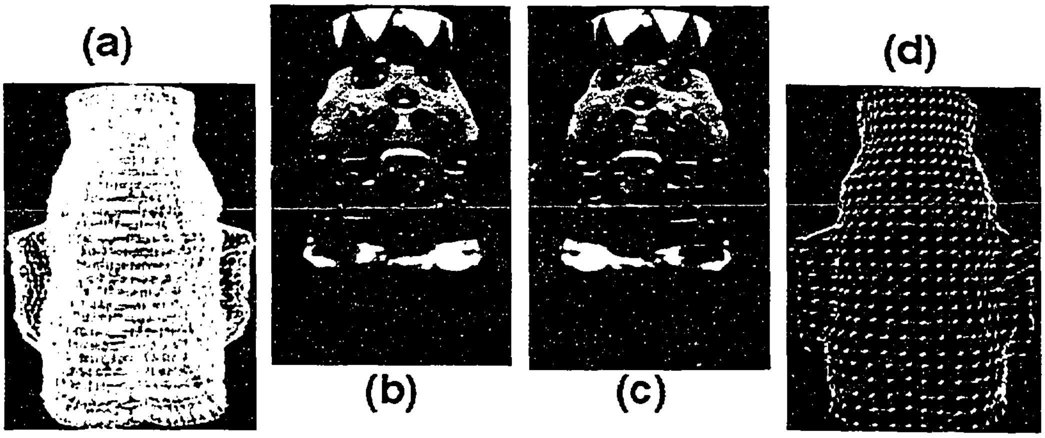


**Fig. 7.10:** *(a) and (d) show the mesh resolution of (b) and (c) respectively.*

(2) An important parameter in the JND model is the shortest (perpendicular) distance $D_R$ from the inserted vertex $V_R$ to the skeleton (Fig. 7.11). Since we are considering dense meshes, the difference between $D_C$ and $D_Q$ of two adjacent vertices $V_C$ and $V_Q$ is not expected to be large.

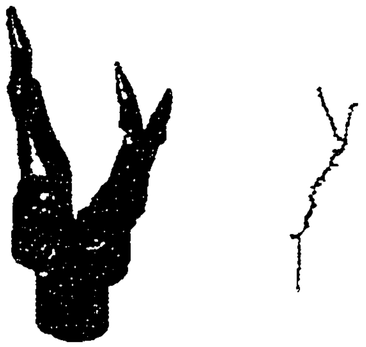


**Fig. 7.11:** An *example of the skeleton of a simulated 3D object, with different branches marked by different colors [SZZ01].*

Bearing in mind the two criteria, we can illustrate the structural change (or stimulus generated) when a vertex is inserted onto a mesh surface. Fig. 7.12 shows the original surface $V_C V_Q$ of a 3D object (cross-section) and $V_R$ is inserted during a refinement operation. Let $V_C$ be the closest neighbour of $V_R$ which is integrated with $V_C$ during a simplification operation as described in Section 7.2. $V_Q$ can be any adjacent vertex of $V_R$ before simplification. $D_C$ and $D_Q$ are the distances to the skeleton from vertex $V_C$ and $V_Q$ respectively. The visual impact measured by the fraction $\frac{(h+\varepsilon)}{H}$ is the highest at vertex $V_R$. Criteria (1) and (2) say that in a dense mesh, $D_C$ and $D_Q$ are of similar length, and thus the value $\Delta d$ and $\varepsilon$ are small. In the experiment using the nutcracker object, the approximation $\frac{h}{H}$ was used to compute the perceptual value. Let us see how significant the value $\varepsilon$ is in affecting visual perception based on the JND of 0.10 obtained from the experiment.

Let $D_R = h + \varepsilon + H$ be the shortest distance to the skeleton. From Equation (7.1) we know that if the perceptual value $\frac{\Delta\rho}{\rho} = \frac{(h+\varepsilon)}{H} < 0.10$, then the structural change is not detectable.

$$\frac{h+\varepsilon}{H} < 0.10 \Rightarrow \frac{h}{H} < 0.10 \Rightarrow \frac{h}{H+\varepsilon} < 0.10 \quad (7.3)$$

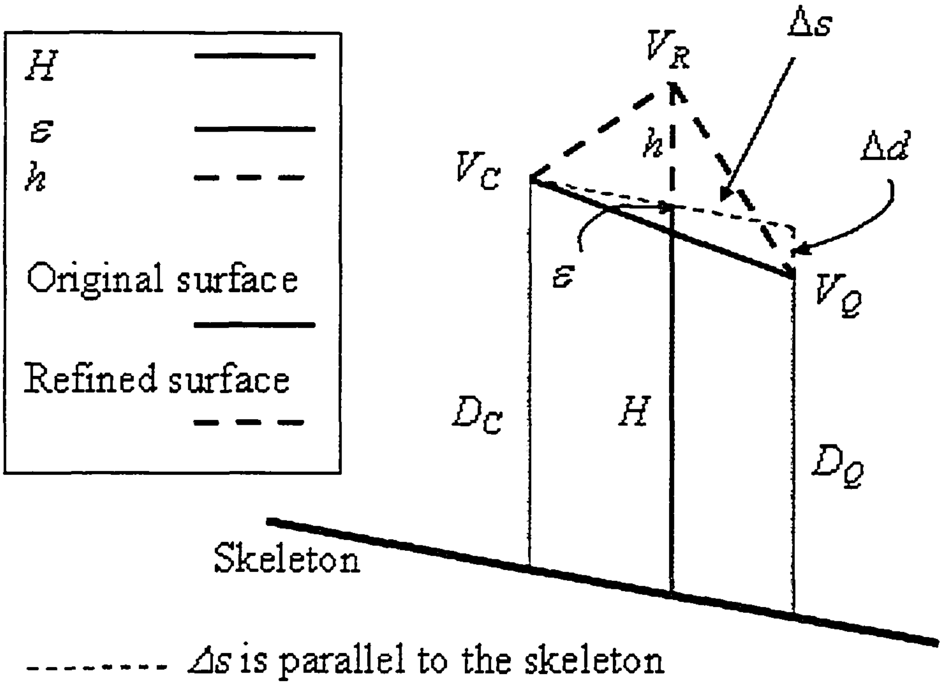


**Fig. 7.12:** *An analysis of the perceptual impact during vertex insertion.*

From Equation (7.3) we know that if $\frac{h+\varepsilon}{H}$ is not detectable $\frac{h}{H+\varepsilon}$ is also not detectable. Notice that:

$$\frac{h}{H+\varepsilon}=\frac{D_R-D_C}{D_C} \qquad (7.4)$$

Equation (7.4) is easy to compute and is used to estimate the perceptual value. This is a conservative approach because values greater than $\frac{h}{H+\varepsilon}$ and less than $\frac{h+\varepsilon}{H}$ are taken as detectable. However, this conservative approach accompanied by the intermediate strategy (discussed in Section 7.2), using the average perceptual value at a scale, works well and produces good result on the nutcracker object.

## 7.5 Efficient Mesh Refinement

Based on the knowledge of JND (0.10), the scales of the nutcracker can be divided into tiers as shown in Fig. 7.13. For simplicity, we assume an application using 20 mesh scales in a distance range of 20 units. We define viewing distance as the distance between the object and the viewing platform in the virtual world. By contrast, physical distance is the distance between the display device and the retina. Instead of a linear relationship (pink Δ), the JND indicates that scales relate to viewing distance following a step function (blue ◊). The concept of a step function is established through perceptual experiments in Chapter 6. In Fig. 7.13 for example, at distance unit 1, $S_4$ is used instead of $S_1$ because it requires a smaller number of vertices and has perceived similarity (Table 7.2). The scale where the pink and blue symbols meet is defined as a major scale and the others are minor scales. Only changes from one major scale to another adjacent major scale have significant impact on visual perception.

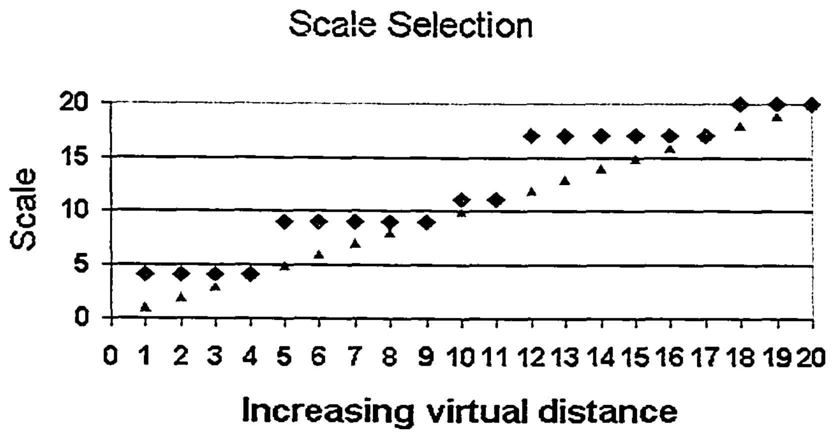


**Fig. 7.13:** *Perceptual function of the nutcracker object, relating scale to virtual distance.*

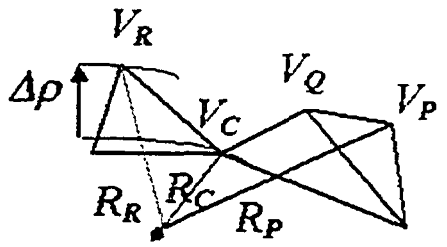
Since geometrically different objects can be perceptually similar [CB04], it is important during online transmission to suppress redundant mesh data, which do not improve visual quality. Major scales can be identified from the LUT during runtime to perform this task. For the nutcracker object the major scales are 4, 9, 11, 17 and 20 (Fig. 7.13).

## 7.6 The Screen Pixel Constraint and Automatic Scale Selection

The JND perceptual metric supports not only efficient mesh refinement, but also automatic selection of scale based on the displayed size of a 3D object. Visual quality increases as fine features appear on the visual surface of a 3D object. During the refinement process, vertices are inserted into the coarser mesh. The screen pixel constraint (distance between grid-points on a display device) has been used in the literature to define the smallest inter vertex distance [COM98] [Red01]. Before reaching this limit, further vertex insertion is assumed to improve visual quality. By applying the JND perceptual metric, it can be shown that visual quality is restricted by the JND and not the screen pixel constraint. Limited computational and network resources can therefore be utilized efficiently by stopping mesh refinement before reaching the threshold imposed by the display device resolution.

### 7.6.1 JND and the Displayed Size of an Object

In the previous sections, we discussed how vertex insertions generate visual stimuli to the retina, and how to compute the perceptual impacts of the stimuli based on Just-Noticeable-Difference. Based on the perceptual evaluation experimental results, 0.10 was established as the JND for 3D TexMesh. Since the surface properties of 3D objects are different, each object is associated with its own step-function. The mesh simplification process is performed during pre-processing and the relative change on the surface structure during an edge collapse operation is recorded in a lookup table (LUT), which can be used at runtime to determine whether further mesh refinement, given limited resources, e.g. bandwidth, will benefit the visual fidelity of the object. This perceptual approach based on JND reduces redundant mesh data which cannot be suppressed by traditional approaches relying solely on a geometric metric. Another issue that needs to be considered is how to automatically select a scale giving the distance between the object and the viewing platform in the virtual world.



● Center of 3D object.
$R_R$, $R_C$ & $R_P$ are radii of vertex $V_R$, $V_C$ & $V_P$ respectively.
$\rho_R = (R_R - R_C) / R_C$.
$\rho_P = (R_P - R_C) / R_C$.

**Fig. 7.14:** *$V_R$ and $V_P$ have perceptual values $\rho_R$ and $\rho_P$ respectively.*

Suppose $R_C$ (Fig. 7.14) is 170 pixels. Based on the JND value (0.10) determined earlier, a stimulus less than 17 (170x0.10) pixels does not have significant perceptual impact. In other words, it is not necessary to insert $V_R$ if it does not generate a stimulus of dimension greater than 17 pixels. Similarly if $R_C$ is 90 pixels, a stimulus of dimension less than 9 pixels will not have significant perceptual impact. In both cases, it can be seen that mesh refinement can be terminated before reaching the resolution of the display device. The dimensions and resolutions of the monitors used in the perceptual experiments are given in Table 7.3.

| Monitor | Dimension (inches / pixels) |
|---|---|
| 1 | 16x12 / 1280x1024 |
| 2 | 11x8 / 1024x768 |
| 3 | 12x9 / 1024x768 |
| 4 | 14x11 / 1280x1024 |

**Table 7.3:** *Resolutions of display devices used in perceptual experiments.*

Screen-space projection was used instead of object-space length when considering a particular view [XEV97]. Since my approach is view-independent, object-space length is used in order to cover all possible viewing directions. Let *D* pixels be the vertical dimension of an object on a screen. $E_S$ and $O_V$ are respectively the length of the shortest edge and vertical length of the object. If $E_S$ occupies 1 pixel, the object will span $O_V/E_S$ pixels vertically. Based on this estimation, the application can select the LOD which has an $O_V/E_S$ value most matching the desired *D* value. Since fine details (shorter edges) tend to diminish at lower scales leaving global structures (longer edges) at higher scales [CB05], a coarser mesh will be automatically selected for an object displayed farther away. Given limited resources, if refining the mesh from (b) to (a) (Fig. 7.15) does not benefit visual quality, the strategy will be to allocate the resources to enhance other multimedia data, such as texture, which may improve visual quality.

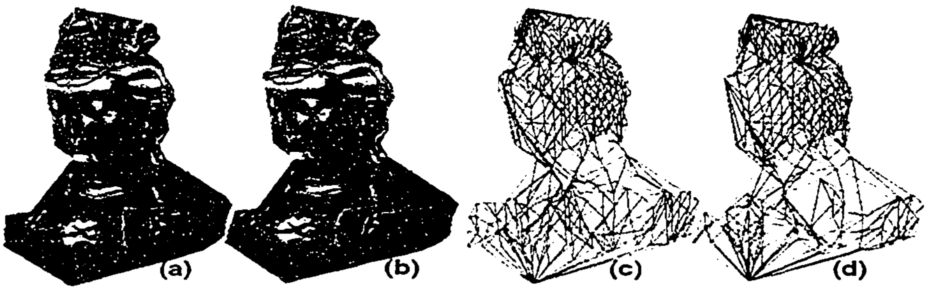


**Fig. 7.15:** *Perceptually similar textured meshes (a & b) composed of 774 and 630 triangles respectively (c & d).*

# 7.7 Conclusion

In this chapter, using perceptual value as a metric was proposed for efficient online visualization of 3D TexMesh. The JND and Weber's fraction were used to evaluate the perceptual impact on the HVS resulting from mesh refinement. The approach is view-independent. Differing from previous techniques, which measure the spatial frequency generated by the stimulus affecting the visual field, this technique is independent of viewing distance. While an *absolute difference* (MSE) between two projected images was used in [LT00], a perceptual value computed as the change *relative to the original stimulus* is used as a perceptual metric, the performance of which is supported by user perceptual evaluations. The novelty of this approach lies on integrating perceptual and geometric metrics to determine at which stage mesh refinement should be terminated, allocating the remaining processing and network resources to other multimedia data. A scale can be selected automatically based on the displayed size of a 3D object. The JND perceptual metric can determine whether vertex insertion can stop before reaching the threshold imposed by the display device resolution.

In the TexMesh  framework, both mesh and texture data need to be considered for efficient online visualization. In Chapter 8, we will discuss how the visual quality of 3D texture can be optimized.

# CHAPTER 8

## A Visual Quality Prediction (VQP) Model for 3D Texture

Differing from previous approaches which focus mainly on 2D and grayscale images, in this Chapter a visual quality prediction (VQP) model which takes both 3D and 2D properties of color texture into consideration will be discussed. Since high-resolution color texture images are much larger than the mesh data, I focus on supporting bandwidth adaptation using texture reduction, which can be associated with an efficient level-of-detail (LOD) algorithm. In order to achieve satisfactory interactivity, applications such as online games use synthetic texture which is often simple and easy to duplicate, so that the transmitted data are kept small. This model is designed for applications such as displaying museum exhibits and medical images, where high-resolution real texture is required.

Visual quality models have been discussed in the literature [OHM*04], but to the best of my knowledge, none of them addresses visual fidelity in a systematic manner, taking bandwidth limitation and fluctuation, together with 3D and 2D texture properties into consideration. A perceptual approach was used, by approximating the Contrast Sensitivity Function (CSF), to simplify details in a scene [Red01]. The metric derived from the CSF was used to perform simplification operations [LH01]. Prior approaches were improved by accounting for textures and dynamic lighting [WLC*03]; their focus was on mesh and not on texture simplification. This view-independent approach analyzes the intrinsic property of the texture image, independent of viewing direction. An image fidelity assessor was discussed in [TPA98]; their technique accepts two grayscale images as input and outputs a distortion value, while this technique is applicable to color as well as grayscale images, and predicts the overall visual fidelity of 3D objects. Two visual fidelity algorithms for mesh simplification were discussed in [WFM01]. A visual difference predictor was used to select the appropriate global illumination algorithm [VMK*00]. The visibility of differences between two images was used to determine whether a particular area of a synthetic scene needed refinement [BM98]. Unlike the proposed approach, these techniques are not designed to guide real texture reduction at multiple scales.

## 8.1. Texture Reduction Driven by 3D and 2D Properties

In addition to the 2D properties of a texture image, there are other factors which can affect the visual quality of 3D objects. These factors are classified as geometry driven and texture driven visual predictors.

### 8.1.1 Geometry Driven Visual Predictors

Past psychophysical experiments show that contrast induced by the 3D surface properties is an important visual cue to predict the resulting quality [Nag84]. In order to represent three-dimensional real world objects on a two-dimensional display device, it is essential to impose the perception of depth and contrast on the HVS. A rough surface requires more contrast then a flat surface in order to highlight the surface. The smoothness of a mesh surface is dictated by its underlying geometry, which can be estimated by the feature point distribution generated by a LOD technique [CB05] [GH98] [Hop96] [HH93]. If the same texture quality is assigned to the entire surface without taking depth and contrast into consideration, a plain surface may have excessive quality, leaving insufficient bandwidth to more complex surfaces, thereby degrading the overall visual quality. An example of how feature point distribution can affect human perception is illustrated in Fig. 8.1. The grenade has vertical structures on the surface, and therefore the feature point distribution is higher than the back of the nutcracker, which is relatively flat. Note that even if the texture quality is reduced to half, there is no significant perceptual degradation on the shiny patch under the belt of the nutcracker. However, the grenade on the right (Fig.8.1d) shows noticeably lower perceptual quality.

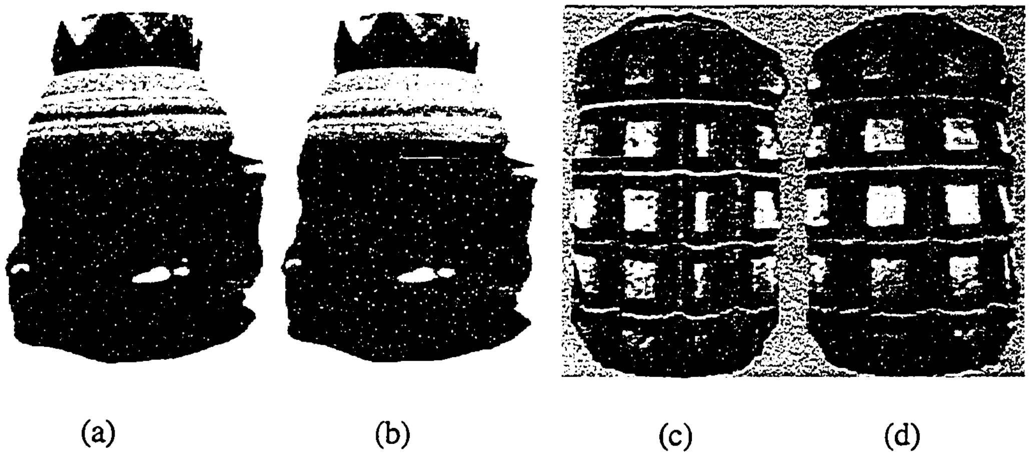


**Fig. 8.1:** *A snap shot of the nutcracker 3D model (a and b), and the military grenade model (c and d), with original texture quality (a and c), and half of the original texture quality (b and d).*

Feature point distribution is therefore identified as one of the geometry driven visual quality predictors for 3D texture. A high resolution texture image is divided into fragments of optimal dimensions. Higher quality is associated with higher feature point distribution to preserve the surface property and to allow better perception of depth and contrast. In online applications, texture data is made available to heterogeneous client displays of different resolutions. It is a waste of resources if excessive resolution is transmitted and cannot be displayed. A solution is to request the display resolution from the client before transmission. Based on the display resolution, the texture of corresponding resolution is selected and fragmented into optimal dimension.

Although feature point distribution is a visual quality predictor, geometry driven texture reduction alone is not sufficient to predict the overall visual quality. We will discuss how texture masking and other 2D image properties affect the resulting quality.

### 8.1.2 Texture Driven Predictors

In addition to geometry, texture complexity also influences how the HVS perceives quality. Visual masking was suggested in the literature [FPS*97]. Psychophysical experiments showed that light intensity and contrast affect human perception [LH01] [Red01] [WB01] [MC95]. The light source can have constant and evenly spread illumination, but the reflective properties vary for different texture patterns. We use the range required for color quantization $Z$ (RGB color model), the texture intensity component $I$ (HSI color model), and the degree of visual masking $M$ induced by the pattern complexity, as our texture driven visual predictors. I assess the performance of these predictors based on their impact on human perception.

#### The *ZIM* Predictors

The texture size can be reduced, by lowering its quality, which means using a smaller quantization range instead of a true color range of 256 values for the red, green and blue components. By analyzing the relative sizes of the color range used to represent different texture patterns, the relative impact of reducing qualities on these patterns can be predicted. For a texture with $n$ pixels, $Z$ is computed as follows:

$$Z = \sigma_R + \sigma_G + \sigma_B \qquad (8.1)$$

$\sigma_R$, $\sigma_G$ and $\sigma_B$ are the standard deviation for red ($R$), green ($G$) and blue ($B$) color values of the texture respectively.

$I$ is computed as the average intensity (brightness) of the texture, using the standard formula to calculate the intensity of each pixel:

$$I = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{r_i + g_i + b_i}{3}\right) \qquad (8.2)$$

$r_i$, $g_i$ and $b_i$ are the red, green and blue color values of each pixel.

The $M$ predictor has a counter effect on the $Z$ and $I$ predictors, and is defined based on two observations: (i) The HVS can discriminate better on brighter surfaces. (ii) Irregular, dense and mixed color patterns on the surface tend to lower the discriminating capability, while the HVS can discriminate better on a plain color surface [FPS*97]. We define $M$ based on these two observations.

$$M = (1 - (\omega_I(1 - I) + \omega_\zeta \zeta)) \quad (8.3)$$

$\omega_I$ and $\omega_\zeta$ are the weights for $I$ and $\zeta$ respectively. The value $\zeta$ is the gradient count, computed as follows: The count for two adjacent pixels is either zero or one. When the difference between two adjacent color values, either the red, green or blue component, exceeds a predefined threshold the count is one. Let us define

$\eta_{east}$ and $\eta_{south}$ as the east and south neighbors of a pixel. Given a threshold $\Gamma$ and a counter starting from zero, we start from the top left pixel in the texture. If the difference of the red, green or blue value between the pixel and $\eta_{east} \geq \Gamma$, the counter is incremented by one. If the difference between the pixel and $\eta_{south} \geq \Gamma$, the counter is incremented again. The process is repeated for all the $(n\text{-}1)$x$(n\ \text{-}1)$ pixels of the image. Increasing light intensity has a counter effect on $\zeta$. Surfaces with higher $M$ values (lower masking effect) are assigned higher qualities.

The *ZIM* predictors are normalized in the range [0,1]. We analyzed the performance of the *ZIM* predictors using 24 different texture patterns (Fig. 8.2). Starting from the bottom left and moving towards the top right, we notice that the texture patterns change from plain to relatively complex. It is observed that patterns (b) and (c) have the same *ZIM* values, although (b) has a higher *Z* value (more affected by color quantization) than (c). This is because texture (c) has higher *I* and *M* values (brighter and less visual masking effect). Both texture (i) and (j) have irregular patterns, but (i) is darker and can mask better, and thus has a slightly lower *ZIM* value than (j). The sharper edges and brighter background in texture (s) result in a higher *ZIM* value compared with the edges in (l), (n), (p) and (q). Texture (n) and (p) are extracted from the grenade surface (Fig. 8.1). (p) is slightly brighter than (n) and thus the *ZIM* value is higher. Fig. 8.3 highlights the $\zeta$ effect of each texture pattern.

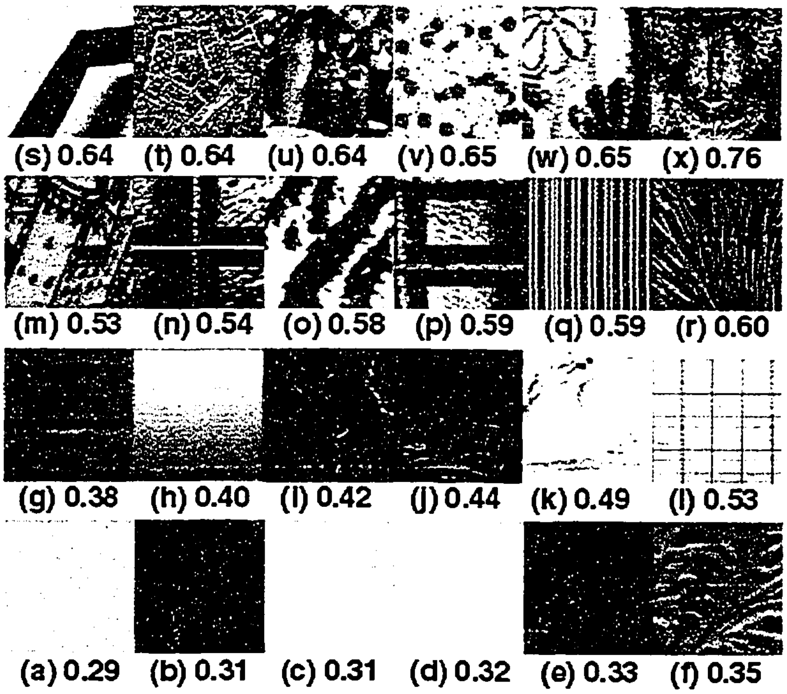


**Fig. 8.2:** *24 texture patterns with corresponding ZIM values used in the experiments.*

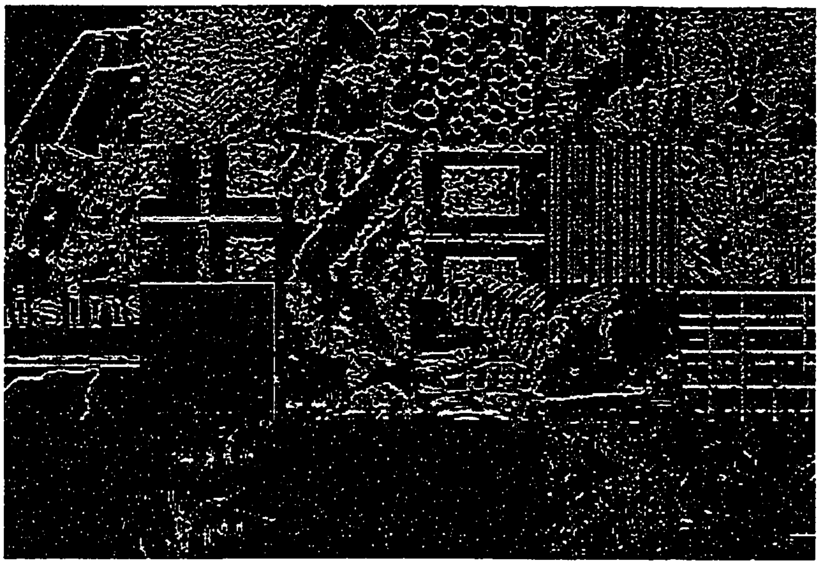


**Fig. 8.3:** *Highlight of the $\zeta$ effect by assigning a grayscale value of 200 to pixels, which have gradient count one, and a value of 0 to other pixels.*

## 8.2 The VQP Computation Model

Visual quality is an important consideration when representing high resolution real texture in online 3D applications. In a constrained environment, there is a tradeoff when distributing limited resources, e.g. bandwidth, among different multimedia data. Earlier perceptual experimental results show that after mesh data has reached a certain minimum resolution, allocating resources to increase texture resolution is more beneficial to the overall visual fidelity, than sharing the remaining resources between texture and mesh data [PCB05] [RRP00]. This concept is used when transmitting museum data [KTL*04]. In this paper, we introduce a VQP computation model, taking both geometry and texture properties into account, to predict the overall resulting quality.

In the current implementation, the predictors are applied equal weights. *ZIM* together with the feature point distribution predictor, generated by a LOD technique, are substituted into Equation (8.4), to predict the overall texture quality of the resulting 3D object,

$$\wp = \sum_{k=1}^{a} \omega_k \rho_k = \sum_{g=1}^{b} \omega_g \rho_g + \sum_{t=1}^{c} \omega_t \rho_t \qquad (8.4)$$

with $a = b + c$, where

$\wp$ — The overall visual quality predictor.

$\rho_k$ — A visual quality predictor.

$\omega_k$ —The weight applied to $\rho_k$.

$\rho_g$ — A geometry driven visual predictor.

$\omega_g$ —The weight applied to $\rho_g$.

$\rho_t$ — A texture driven visual predictor.

$\omega_t$ —The weight applied to $\rho_t$.

$a$ —Total number of predictors.

$b$ —Number of geometry driven predictors.

$c$ —Number of texture driven predictors

The model can be extended to accommodate more geometry and texture driven visual quality predictors.

## 8.3 Online Visualization

As discussed in Chapter 4, texture fragments are used to absorb bandwidth fluctuations. Instead of applying a uniform quality, each fragment is assigned a different quality based on the associated visual quality predictor $\wp$. The VQP model is designed to support the fragmentation approach. A smaller area of the texture has a more uniform pattern, while the entire texture image taken as a whole is likely to contain surfaces of diverse patterns, making the $\wp$ value more of a global average. In this thesis, the JPEG quality scale 0% to 100% is used because JPEG images are widely supported on the web and in JAVA applets. However, the VQP concept can be applied to other compression schemes, such as JPEG2000. The Intel JPEG encoder and decoder were used in the current implementation. Each LOD generated is assigned a default quality scale $Q_i$ based on viewing distance [CB04]. Given a visual quality predictor $\wp$, the data size $S_i$ corresponding to a quality scale $Q_i$ is given by:

$$S_i = A(100\wp)^{BQ_i} \qquad (8.5)$$

$Q_i$ is normalized in the range [0, 1]. Values $A$ and $B$ are constants for a given $\wp$ value. Note that $P_i$ is maximum $(100\wp^2)$ when $Q_i = 1$. We choose a polynomial function for the mapping because of the polynomial characteristic of the JPEG quality *vs.* data size curve (Fig. 8.4).

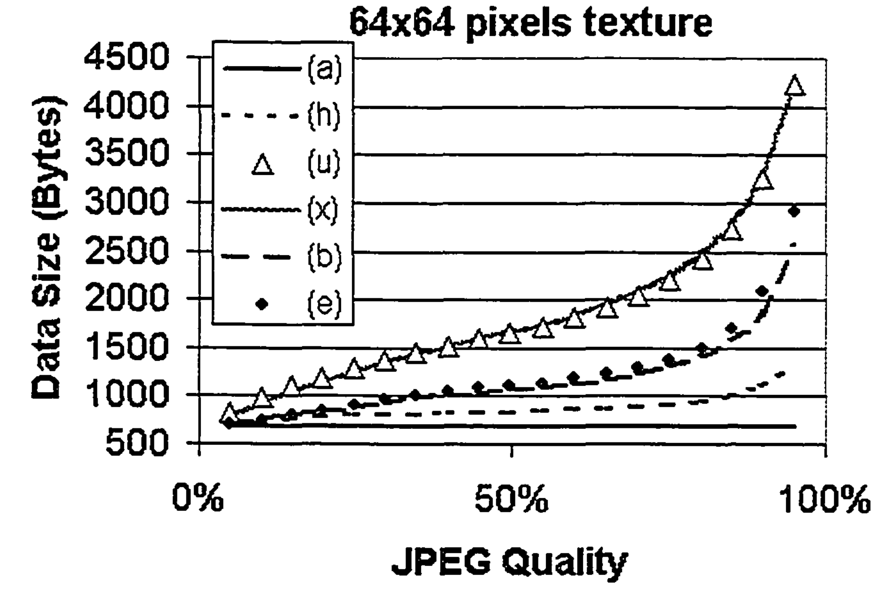


**Fig. 8.4:** *JPEG data size relates to quality following an exponential function.*

During online transmission, the estimated data size of the next fragment to be transmitted is computed based on the current bandwidth. By selecting several (quality, data size) pairs, constants $A$ and $B$ in Equation (8.5) can be derived by the curve of the best-fit method. For example, texture (x) has $A$ = 825.45 and $B$ = 0.34. In the quality range [5%, 100%], texture (x) has the best-fit curve with correlation coefficient of 0.98. To compute constants $A$ and $B$, we denote $R$ = $(100\wp)^B$ and establish the equation:

$$S_i = AR^{Q_i} \,. \qquad (8.6)$$

Constants $A$ and $R$ are solved by the regression method using Equation (8.7).

$$\log S_i = \log A + Q_i \log R \,. \qquad (8.7)$$

Constant $B$ is solved by combining Equation (8.5) and Equation (8.6), and taking the logarithm on both sides:

$$B = \frac{\log R}{\log(100\wp)}. \qquad (8.8)$$

From Equation (8.7) and Equation (8.8):

$$Q_i = \frac{\log S_i - \log A}{B \log(100\wp)}. \qquad (8.9)$$

Equation (8.9) can then be used to compute the quality based on a given data size. An alternate way to obtain $Q_i$ and $S_i$ is to generate a lookup table during preprocessing and store the (quality, data size) pairs. The most matching value is selected during runtime. Fig. 8.5 illustrates the result of incorporating *ZIM* in the variable quality approach. The texture data is divided into 4 fragments. Suppose that the available bandwidth can support 4492 bytes of data. Applying a fixed quality to every fragment, each one can have 30% quality (Fig. 8.5 I). If the variable quality approach is applied, texture (x), (q), (j) and (c) can have 44%, 34%, 27% and 26% quality respectively maintaining a total data size of 4492 bytes (Fig. 8.5 III). The qualities of texture fragments (c) and (j) are lower than 30% but there is no significant degradation in the image. Texture (x) is upgraded from 30% to 44%. Note that a satisfactory quality of the eyes and nose of the baboon is maintained, closer to the original image (Fig. 8.5 II).

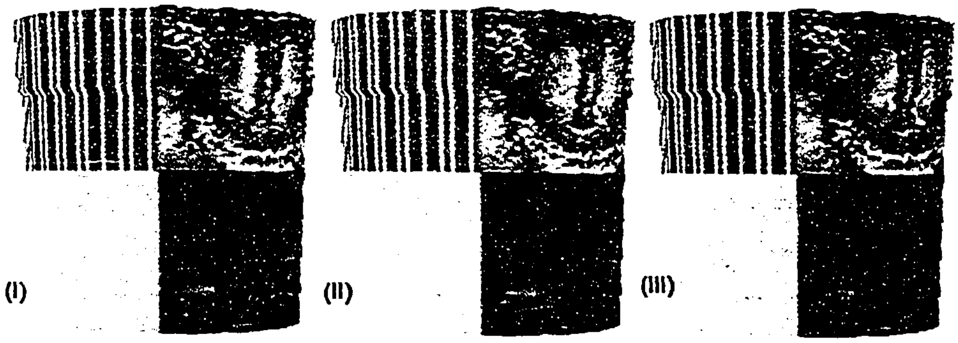


**Fig. 8.5:** *An example of variable quality assignment based on ZIM: Texture (x) with quality at (II) 100%, (I) 30% and (III) 44% are mapped onto a pot 3D object.*

## 8.4 Conclusion

In this Chapter, a VQP model is introduced, which supports an adaptive fragmented texture transmission approach for 3D objects. Differing from other visual discrimination models in the literature, geometry driven as well as texture driven visual predictors are incorporated into the model, taking bandwidth fluctuation into account to best predict the quality of the resulting 3D object. Although the JPEG compression is used in the current implementation, the model can be applied to other compression schemes. The model is extensible to incorporate more predictors if required.

After discussing feature extraction, mesh and texture reduction, bandwidth adaptation and perceptual evaluation in previous chapters, we will discuss how these components are integrated in the TexMesh framework in Chapter 9.

# Chapter 9

# Integrating Feature Extraction, Simplification at Multiple Scales, Bandwidth Adaptation and Perceptual Evaluation into the TexMesh Framework

The TexMesh model presented here applies an adaptive approach, which does not need to sacrifice transmission time [YCB03]; at the same time, bandwidth is efficiently utilized by adjusting the quality of the fragments not yet transmitted. The entire texture image is divided into fragments of optimal size. Given a viewing distance and a historic average bandwidth, the matching scale and a default quality for texture are selected. The first fragment is transmitted with the default quality. The idea is to re-estimate the current bandwidth based on the transmission time of the previous fragment, as well as the previous fragment size. The updated bandwidth is then used to compute the quality of the next fragment to be transmitted. After transmitting each fragment, the over- or under- estimated time is distributed to the remaining fragments, using the Harmonic Time Compensation Algorithm. This will ensure that, when the transmission is completed, the overall time discrepancy (compared to the predefined time limit) is minimized.

In addition to bandwidth adaptation, the TexMesh model is unique in its feature- and perceptually-driven characteristics. Surface property represented by feature points is extracted using SSF. Triangulated meshes are generated based on feature points at multiple scales. The variable quality assigned to each texture fragment is based on the feature point density and other texture properties associated with the fragment. The premise of assigning higher quality to regions with higher feature point density, and lower quality to regions with lower feature point density, is based on perceptual evaluation and the convergence property of SSF.

In [PCB05], the authors found that improving mesh resolution improves perceptual quality following an exponential curve, whereas enhancing texture improves perceptual quality linearly (see detailed discussion in Chapter 5). Based on this finding, and the fact that the texture component in a 3D image requires greater storage or bandwidth for communication, the TexMesh model uses SSF analysis to integrate texture reduction with mesh simplification. The mesh data is transmitted first and the remaining time is allocated to texture data. The approach can be summarized as follows:

## 9.1 Preprocessing

Step 1 — Perform a SSF analysis of the 3D object and identify regions of strong persistent structures *vs*. regions of small surface variations, at different scales. Generate a priority list of feature points from strong to weak persistence.

Step 2 — Compute the perceptual value for each edge collapse operation, as well as the cumulative perceptual values at each scale. These values are stored in a lookup table for online retrieval.

Step 3 — Relate perception of scale to viewing distance. Establish a step function and identify major and minor scales for each 3D object.

Step 4 — Divide the model texture into fragments of optimal size. Generate a fragment map for each scale $S_i$, by distributing the feature points onto corresponding texture fragments.

Step 5 — Compute the *ZIM* predictors for the texture fragments and derive the mapping function between a texture quality level and the corresponding data size.

Step 6 — For each scale $S_i$, use a lookup table to record the individual and total fragment size.

## 9.2 Runtime Processing

Step 1 — Given the displayed size of a 3D object, refer to its step-function and perceptual values. Select the mesh scale $S_i$, using the JND perceptual metric.

Step 2 — Based on the mesh scale $S_i$, choose the default quality and use the initial bandwidth (historic average) and $T_0$ to locate the closest matching set of texture fragments in the lookup table, taking visual quality prediction into consideration. $T_0$ is the given time limit, minus the time required to transmit the geometric data (mesh).

Step 3 — Adjust the texture quality of each fragment using the adaptive approach and transmit the texture fragments.

Step 4 — The client site recombines fragments and renders the texture-mapped 3D model.

# Chapter 10

## Conclusions and Future Research

The TexMesh model proposed in this thesis integrates mesh simplification and texture reduction, based on scale-space analysis. Laplacian of Gaussian (LoG) is used to detect zero-crossings at each scale and generate statistics, including a scale map and fragment map, during preprocessing. These statistics are used during runtime to ensure efficient extraction and transmission of 3D data. Feature points are transmitted only if they contribute to visual quality. Variable qualities applied to texture fragments are readjusted during transmission to adapt to fluctuations in bandwidth. Experimental results show that the Harmonic Time Compensation adaptive approach utilizes bandwidth efficiently, and provides satisfactory control on QoS.

Packet loss during transmission is a common problem. When no other data is competing for bandwidth, sending redundant mesh data can ensure that the rendered quality will not be significantly affected by missing vertices. Since the TexMesh method is designed for transmitting a 3D textured mesh, each 3D vertex is associated with a 2D texel. By transmitting the vertex and texel coordinates in separate packets. the (x,y) coordinates of a vertex, if lost, can be estimated based on the corresponding texel coordinates received in a separate packet. Recall that feature points in the priority queue represent decreasing structure sizes. Therefore, the depth component, z, can be estimated by taking the average of the previous and next feature points. However, this packet loss strategy works only when the application can tolerate a certain degree of estimation error. For applications which require high precision, retransmission is more appropriate.

The step function relating scales to a given distance, which is consistent with the convergence property of scale-space filtering, suggests an efficient way to suppress redundant data during mesh refinement. Simplification techniques in the literature often use a number of vertices, or triangles, or a quantitative metric to measure efficiency. The experiments presented here show that perceptual quality is a more reliable measure, because the ultimate judgement on visual quality is made by the Human Visual System. An interesting outcome of this thesis is the distinction between major and minor scales in level-of-detail, which applies to all simplification techniques (geometrically- or perceptually-based). Inserting vertices into a mesh has major impact only if there is a significant change to the HVS. In other words, two mathematical models can generate meshes of equal visual fidelity. How perception relates to surface property is not fully understood. **To the best of my knowledge, associating perceptual quality of 3D surfaces with the convergence property of filtering is novel.**

Mesh refinement techniques in the literature assume that inter-grid length on the display imposes a limit for further refinement. In this thesis the JND perceptual metric is applied, and it is demonstrated that mesh refinement can be terminated before reaching the threshold imposed by the display device resolution (without affecting visual quality) – allocating the remaining resources to texture data. In order to optimize texture quality given limited bandwidth, a visual quality prediction model based on human perception is incorporated into the framework.

View-dependent rendering, when utilized in a user collaboration environment, requires the server to keep track of a large number of rendered views, and transmit different sets of edited data based on each viewpoint. There is a trade-off between rendering the complete object at the outset and on-demand. For example, when the user wishes to view high-resolution medical data, museum exhibits, and E-commerce products, reserving time at the beginning to download the complete object enables swift subsequent navigation. The alternative is to wait for an incremental update after each change of view. The TexMesh framework is designed for applications requiring high resolution 3D objects and fast navigation, trading off some download time at the outset.

**The main contribution of the TexMesh model is the introduction of an integrated framework, combining: feature extraction; mesh simplification associated with texture reduction, based on feature filtering and distribution; bandwidth adaptation; and perceptual evaluation**. A complete framework integrating related issues from different directions is missing in the literature.

More importantly, perceptually adaptive graphics has been recognized as an essential element in bringing realism into the virtual world. The perceptual metric introduced in this thesis is an example of research effort in this direction.

## 10.1 Thesis Publications to Date and Future Research

The TexMesh 3D visualization framework lays the foundation for future analysis and research. Selected work has been accepted in two IEEE journals and seven international conferences:

(1) I. Cheng and P. Boulanger, "Feature Extraction on 3D TexMesh Using Scale-space Analysis and Perceptual Evaluation," IEEE Transactions on Circuits and Systems for Video Technology Special Issue 2005 (in press), 10 transactions pages.

(2) I. Cheng and P. Boulanger, "Adaptive Online Transmission of 3D TexMesh Using Scale-space and Visual Perception analysis," IEEE Transactions on Multimedia (to appear), 12 transactions pages, 2005.

(3) I. Cheng and P. Boulanger, "A 3D Perceptual Metric using Just-Noticeable-Difference," EUROGRAPHICS 2005 Dublin, Short Paper, 4 pages.

(4) I. Cheng and P. Boulanger, "A Visual Quality Prediction Model for 3D Texture," EUROGRAPHICS 2005 Dublin, Short Paper, 4 pages.

(5) I. Cheng and P. Boulanger, "Automatic Selection of Level-of-Detail based on Just-Noticeable-Difference," SIGGRAPH 2005 Poster Session Los Angeles, 1-page Abstract, Poster and Presentation (ACM SRC finalist).

(6) I. Cheng and P. Boulanger, "Scale-space 3D TexMesh Simplification" IEEE ICME 2004 Taipei, 4 pages.

(7) I. Cheng and P. Boulanger, "Perception of scale with distance in 3D visualization" SIGGRAPH 2004 Poster Session Los Angeles, 1-page Abstract, Poster and Presentation (ACM SRC contestant).

(8) I. Cheng and P. Boulanger, "Adaptive online transmission of 3D TexMesh using scale-space analysis" 3DPVT 2004 Thessaloniki, 8 pages.

(9) I. Cheng, "Efficient 3D objects simplification & fragmented texture scaling for online visualization" IEEE ICME 2003 Baltimore, 4 pages.

Each research topic in this thesis can be extended and integrated with other research areas in Computer Science. Future directions will focus on:

1. Extending perceptual evaluations on texture quality, comparing the results with other visual discrimination models and assessing the best combination of different weights associated with the visual quality predictors.

2. Assessing the real-time impacts when integrating the Harmonic Time Compensation Algorithm, the JND perceptual metric and the VQP model, taking packet loss into consideration.

3. Incorporating view-dependent perceptual considerations which may improve the performance of the view-independent JND technique if only part of the mesh vertices need to be displayed.

4. Extending the TexMesh framework to 3D dynamic scenes, containing complex shapes and different illumination conditions. Complex shapes, including both man-made and natural objects, can be segmented into simple parts before the scale-space filtering technique is applied. Multiple objects, the moving velocity of an object and the viewer's area of interest are also factors to be considered in a dynamic scene.

5. Incorporating 3D data storage, retrieval and distribution into the TexMesh framework.

6. Extending the framework to applications using mobile and wireless computing.

In order to accomplish these ambitious tasks, collaborations with different research groups and other disciplines (*i.e.* Psychology and Art) are required. This thesis provides the foundation for future research, focusing on extending and enhancing the TexMesh framework for efficient online 3D visualization.

# Bibliography

[AE02] Abbey C. and Eckstein M., "Classification image analysis: Estimation and statistical inference for two-altemative forced-choice experiments," Joumal of Vision, 2: 66-78, 2002.

[AS04] Arguin M. and Saumier D., "Independent processing of parts and of their spatial organization in complex visual objects," Psychological Science, American Psychological Society, 15(9): 2004.

[BDD*03] Baudisch P., DeCarlo D., Duchowski A. and Geisler W., "Focusing on the essential: considering attention in display design," ACM Communication, vol. 46, no. 3, pp. 60-66, March 2003.

[BG95] Bronnimann H. and Goodrich M., "Almost optimal set covers in finite VC-dimension," Discrete and Computational Geometry, vol. 14, pp. 463-479, 1995.

[BJB02] Boulanger P., Jokinen O. and Beraldin A., "Intrinsic filtering of range images using a physically based noise model," In Proc. Vision Interface, pp. 320-330, Calgary, 2002.

[BKT86] Boff K., Kaufman L. and Thomas J., "Handbook of perception and human performance, Sensory Processes and Perception," A Wiley—Interscience Publication, vol. I, 1986.

[BM98] Bolin M. and Meyer G., "A Perceptually based adaptive sampling algorithm," in Proc. SIGGRAPH 1998, pp. 299-309.

[BP02] Bauer D. and Peikert R., "Vortex tracking in scale-space," in Proc. IEEE Symposium on Visualization Eurgraphics- TCVG, pp. 140-147, 2002.

[BT03] Bugmann G. and Taylor J., "A model of visual backward masking," Submitted to Biosystems – 6.12.2003, University of Plymouth and Kings College London 2003.

[BTB02] Balmelli L., Taubin G. and Bernardini F., "Space-optimized texture maps," in Proc. Eurographics 2002, pp. 411-420, Saarbrucken, Germany.

[BW00] Brodsky D. and Watson B., "Model simplification through refinement," in Proc. of Graphics Interface, pp. 221-228, 2000.

[CB04] Cheng I. and Boulanger P., "Perception of scale with distance in 3D visualization," in Proc. SIGGRAPH Poster Session (1 page abstract) 2004.

[CB05] Cheng I. and Boulanger P., "Feature Extraction on 3D TexMesh Using Scale-space Analysis and Perceptual Evaluation," IEEE Trans. on CSVT Special Issue, vol. 15, no. 10 (11 pages), October 2005.

[CBZ*01] Cheng I., Basu A., Zhang Y. and Tripathi S., "QoS specification and adaptive bandwidth monitoring for multimedia delivery," in Proc. IEEE EUROCON, pp. 485-488, Slovakia, 2001.

[Cla76] Clark J., "Hierarchical geometric models for visible surface algorithms," ACM Commumication, 19(10): 547-554, 1976.

[CMF99] Cohen-Or D., Mann Y., Fleishman S., "Deep compression for streaming texture intensive animations," in Proc. SIGGRAPH 1999, pp. 261-267.

[CMS98] Cignoni P., Montani C., Scopigno R., "A comparison of mesh simplification algorithms," Computer and Graphics, 22(1): 37-54, 1998.

[COM98] Cohen J., Olano M. and Manocha D., "Appearance-preserving simplification," in Proc. SIGGRAPH 1998, pp. 115-122.

[CPS97] P. Cignoni, E. Puppo, R. Scopigno, "Representation and visualization of terrain surfaces at variable resolution," The Visual Computer, 13(5): 199-217, 1997.

[CVM*96] Cohen J., Varshney A., Manocha D., Turk G. and Weber H., "Simplification envelopes," in Proc. SIGGRAPH 1996, pp. 119-128.

[Dal92] Daly S., "Visible differences predictor: An algorithm for the assessment of image fidelity," Proc. of the SPIE 1992, vol. 1616, pp. 2-15.

[DG95] Das G. and Goodrich M., "On the complexity of approximating and illuminating three-dimensional convex polyhedra," 4th Workshop Algorithms Data Structure, Lecture Notes in Computer Science, Springer-Verlag, 1995.

[DP73] Douglas D. and Peucker T., "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," The Canadian Cartographer, 10(2):112–122, 1973.

[EDD*95] Eck M., DeRose T., Duchamp T., Hoppe H., Lounsbery M. and Stuetzle W., "Multiresolution analysis of arbitrary meshes," Computer Graphics (Proc. SIGGRAPH 1995), ACM Press, 29: 173-182, 1995.

[FL79] Fowler R. and Little J., "Automatic extraction of irregular network digital terrain models," in Proc. SIGGRAPH 1979, pp. 199-207.

[FMP00] Floriani L., Magillo P. and Puppo E., "Variant: A system for terrain modeling at variable resolution," GeoInformatica, Kluwer Academic Publishers, 4(3): 287-315, 2000.

[FPS*97] Ferwerda J., Pattanaik S., Shirley P. and Greenberg D., "A model of visual masking for computer graphics," in Proc. SIGGRAPH 1997, pp. 143-152.

[GH97] Garland M. and Heckbert P., "Simplification using quadric error metrics," in Proc. SIGGRAPH 1997, pp. 209-216.

[GH98] Garland M. and Heckbert P., "Simplifying surfaces with color and texture using quadric error metrics," in Proc. IEEE Visualization, pp. 263-269, 1998, North Carolina, USA.

[GW02] Gonzalez R. and Woods R., "Digital Image Processing," second edition, Prentice Hall, pp. 39-42, 2002.

[GWM01] Gumhold S., Wang X. and Macleod R., "Feature extraction from point clouds," in Proc. 10th Int'l Meshing Roundtable, pp. 293-305, 2001.

[HG97] Heckbert P., Garland M., "Survey of polygonal surface simplification algorithms," Conf. Course: Multiresolution Surface Modeling, SIGGRAPH 1997.

[HH93] Hinker P. and Hansen C., "Geometric optimization," in Proc. Visualization 1993, pp. 189–195.

[HHK*95] He T., Hong L., Kaufman A., Varshney A. and Wang S., "Voxel-based object simplification," in Proc. Visualization 1995, pp. 296-303.

[Hop96] Hoppe H., "Progressive meshes" in Proc. SIGGRAPH 1996, pp. 99-108

[Joh91] Johnston E., “Systematic distortions of shape from stereopsis,” Vision Research, 31: 1351-1360, 1991.

[Joh94] www.psychol.ucl.ac.uk/vision/publications/Johnston-1994-Perception-23-169.pdf.

[JPG05] http://www.jpeg.org/jpeg/

[KBS*01] Kobbelt L., Botsch M., Schwanecks U. and Seidel H., “Feature sensitive surface extraction from volume data,” in Proc. SIGGRAPH 2001, pp. 57-66.

[KCH*91] Kalvin A., Cutting C., Haddad B. and Noz M., “Constructing topologically connected surfaces for the comprehensive analysis of 3D medical structures,” SPIE 14: 247–259, 1991.

[KF02] Kuijper A. and Florack L., “Logical filtering in scale space,” Dept. Comput. Sci., Utrecht University, Technical Report, UU-CS-2002-018, 2002.

[KG03] Kho Y. and Garland M., “User-guided simplification,” ACM Symposium on Interactive 3D Graphics 2003, pp. 123-126.

[Koe84] Koenderink J., “The structure of images,” Biological Cybernetics, Vol. 50, pp. 363-370, 1984.

[KSS03] Khodakovsky A., Schroder P. and Sweldens W., “Progressive geometry compression,” in Proc. SIGGRAPH 2000, pp. 271-278.

[KT96] Kalvin A. and Taylor R., “Superfaces: polygonal mesh simplication with bounded error,” IEEE Computer Graphics and Applications, 16(3): 1996.

[KTL*04] Koller D., Turitzin M., Levoy M., Tarini M., Croccia G., Cignoni P. and Scopigno R., "Protected interactive 3D graphics via remote rendering," ACM Trans. on Graphics, vol. 23, no. 3, pp. 695-703, Aug. 2004.

[LE97] Luebke D. and Erikson C., "View-dependent simplification of arbitrary polygonal environments," in Proc. SIGGRAPH 1997, pp. 199-208.

[LH01] Luebke D. and Hallen B., "Perceptually driven simplification for interactive rendering," in Proc. 12th Eurographics Workshop on Rendering Techniques, pp. 223-224, London, UK 2001.

[Lim79] Limb J., "Distortion criteria of the human viewer," IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-9(12): 1979.

[Lin00] Lindstrom P, "Out-of-core simplification of large polygonal models," in Proc. SIGGRAPH 2000, pp. 259-262.

[LKH*95] Lindstrom P., Koller D., Hodges L., Ribarsky W., Faust N. and Turner G., "Level of Detail Management for Real-Time Rendering of Phototextured Terrain," TR-95-06, Graphics, GeorgiaTech, 1995.

[Lindberg95] Lindberg T., "A scale selection principle for estimating image deformations," ICCV, Cambridge, MA, USA, 1995, pp. 134-141.

[Lindberg98] Lindberg T., "Feature detection with automatic scale selection," International Journal of Computer Vision, vol. 30, no. 2, pp. 77-116, 1998.

[LT00] Lindstrom P. and Turk G., "Image-driven simplification," ACM Trans. on Graphics, vol. 19, no. 3, pp. 204-241, Jul. 2000.

[LT97] Low K. and Tan T., "Model simplification using vertex clustering," in Proc. Symposium Interactive 3D Graphics 1997, pp. 75-82.

[Lue01] Luebke D., "A Developer's survey of polygonal simplification algorithms," IEEE Computer Graphics and Applications, vol. 21, no. 3, pp. 24-35, May/June 2001.

[MCT*00] McNamara A., Chalmers A., Troscianko T. and Gilchrist I., "Comparing real & synthetic scenes using human judgements of lightness," in Proc. Eurographics Workshop on Rendering, pp. 207-218, 2000.

[Mcn00] McNamara A., "Visual perception in realistic image synthesis," STAR - State of The Art Report, Eurographics 2000.

[Nad86] Nadler E., "Piecewise linear best $L_2$ approximation on triangulations," In: C.K.Chui et al., editors, Approximation Theory V, Academic Press, pp. 499-502, Boston, 1986.

[Nag84] Nagata S., "How to reinforce perception of depth in single two-dimensional pictures," in Proc. of the SID, 35(3): 1984.

[NU05] http://www.ncsa.uiuc.edu/VR/VRHomePage.html

[OC00] Okuda M. and Chen T., "Joint geometry/texture progressive coding of 3D models," in Proc. IEEE Int'l Conf. on Image Processing (ICIP), vol. 3, pp. 632-635, Vancouver, 2000.

[ODG*03] O'Sullivan C., Dingliana J., Giang T. and Kaiser M., "Evaluating the visual fidelity of physically based animations," ACM Trans. on Graphics, vol. 22, no. 3, pp. 527-536, July 2003.

[OHM*04] O'Sullivan C., Howlett S., McDonnell R., Morvan Y. and O'Conor K., "Perceptually adaptive graphics," STAR - State of The Art Report, Eurographics 2004.

[PCB05] Pan Y., Cheng I., and Basu A., "Quantitative metric for estimating perceptual quality of 3D objects," IEEE Trans. on Multimedia, vol. 7, no. 2, pp. 269-279, Apr. 2005.

[PGK02] Pauly M., Gross M. and Kobbelt L., "Efficient simplification of point-sampled surfaces," in Proc. IEEE Visualization, pp. 163-170, 2002.

[PK04] Pan M. and Klette G., "A revision of a 3D skeletonization algorithm," http://www.citr.auckland.ac.nz/techreports/2004/CITR-TR-143.pdf.

[PKG03] Pauly M., Keiser R. and Gross M., "Multi-scale feature extraction on point-sampled surfaces," in Proc. Eurographics 2003, pp. 281-289, Granada, Spain.

[PS03] Pojar E. and Schmalstieg D., "User-controlled creation of multiresolution meshes," ACM Symposium on Interactive 3D Graphics, pp. 127-130, 2003.

[PSB*01] Palagyi K., Sorantin E., Balogh E. and Kuba A., "A sequential 3D thinning algorithm and its medical applications," http://www.cisst.org/~emese/publications/IPMI2001_palagyi.pdf.

[RB93] Rossignac J. and Borrel P., "Multi-resolution 3D approximations for rendering complex scenes," Geometric Modeling in Computer Graphics, Springer-Verlag 1993, pp. 455-465.

[Red97] Reddy M., "Perceptually modulated level of detail for virtual environments," Ph.D. Thesis, University of Edinburgh, 1997.

[Red01] Reddy M., "Perceptually optimized 3D graphics," Applied Perception, vol. 21, pp. 68-75, September/October 2001.

[RP03] Reitsma P. and Polland N., "Perceptual metrics for character animation: sensitivity to errors in ballistic motion," ACM Transaction on Graphics, 22(3): pp. 537-542, 2003.

[RRP00] Rushmeier H., Rogowitz B. and Piatko C., "Perceptual issues in substituting texture for geometry," in Proc. of SPIE Human Vis. Electronic V, vol. 3959, pp. 372-383, 2000.

[RV79] Rovamo J. and Virsu V., "An estimation and application of the human cortical magnification factor," Experimental Brain Research, vol. 37, pp. 495-510, 1979.

[SC91] Schmitt F. and Chen X., "Fast segmentation of range images into planar regions," in Proc. CVPR, pp. 710-711, 1991.

[SG01] Shaffer E. and Garland M., "Efficient adaptive simplification of massive meshes," in Proc. IEEE Visualization, pp. 127-134, 2001.

[SGR96] Soucy M., Godin G. and Rioux M., "A texture-mapping approach for the compression of colored 3D triangulations," The Visual Computer, 12: pp. 503-514, 1996.

[Skel05] http://www.inf.u-szeged.hu/~palagyi/skel/skel.html

[Sou91] Southard D., "Piecewise planar surface models from sampled data," in N. M. Patrikalakis, editor, Scientific Visualization of Physical Phenomena, Springer-Verlag, pp. 667-680, 1991.

[SP92] Scarlatos L. and Pavlidis T., "Optimizing triangulations by curvature equalization," in Proc. Visualization 1992, pp. 333-339.

[SSG*01] Sander P., Snyder J., Gortler S. and Hoppe H., "Texture mapping progressive meshes," in Proc. SIGGRAPH 2001, pp. 409-416.

[SW04] Sweet G. and Ware C., "View direction, surface orientation and texture orientation for perception of surface shape," in Proc. Graphics Interface, pp. 97-106, 2004.

[SZL92] Schroeder W., Zarge J. and Lorensen W., "Decimation of triangle meshes," ACM Trans. on Computer Graphics (Proc. SIGGRAPH 1992), 26(2): pp. 65-70, 1992.

[SZZ01] Shahrokni A., H. Zadeh H. and Zoroofi R., "Fast skeletonization algorithm for 3-D elongated objects," http://www.radiologyresearch.org/SPIE01-M14322-34.pdf.

[TAP98] Taylor C., Allebach J. and Pizlo Z., "The image fidelity assessor," in Proc. of IS&T Image Processing, Image Quality and Image Capture Systems Conf., pp. 237-241, 1998.

[TPA98] Taylor C., Pizlo Z. and Allebach J., "Perceptually relevant image fidelity," in Proc. of IS&T/SPIE Int'l Symposium on Electronic Imaging Science and Technology, pp. 110-118, 1998.

[Tur91] Turk G., "Generating texture on arbitrary surfaces using reaction-diffusion," in Proc. SIGGRAPH 1991, pp. 289-298.

[Tur92] Turk G., "Re-tiling polygonal surfaces," in Proc. SIGGRAPH 1992, pp. 55-64.

[UA05] http://web.cs.ualberta.ca/~graphics/introlab.html

[US05] http://www.uni-stuttgart.de/index.en.html

[VMK*00] Volevich V., Myszkowski K., Khodulev A. and Kopylov E., "Using the visual differences predictor to improve performance of progressive global illumination computation," ACM Trans. on Graphics, 19(1): 122-161, 2000.

[WB00] Wang Z. and Bovik A., "A human visual system-based objective video distortion measurement system," in Proc. Multimedia Processing and Systems (4 pages), August 2000.

[Weber05] www.usd.edu/psyc301/WebersLaw.htm

[WFM01] Watson B., Friedman A. and Mcgaffey A., "Measuring and predicting visual fidelity," in Proc. SIGGRAPH 2001, pp. 213-220.

[Wit83] Witkin A., "Scale-space filtering," International Joint Conference on AI, 1983, pp. 1019-1022.

[WLC*03] Williams N., Luebke D., Cohen J., Kelley M. and Schubert B., "Perceptually guided simplification of lit, textured meshes," in Proc. SIGGRAPH 2003, pp. 113-121.

[XEV97] Xia J., El-Sana J. and Varshney A., "Adaptive real-time level-of-detail-based rendering for polygonal models" IEEE Trans. on Visualization and CG, vol. 3, no. 2, pp. 171-183, April 1997.

[YFM00] Yu Y., Ferencz A. and Malik J., "Compressing texture maps for large real environments," in Proc. SIGGRAPH Sketches and Applications (1 page) 2000.

[YCB03] Yu Y., Cheng I. and Basu A., "Optimal adaptive bandwidth monitoring," IEEE Trans. on Multimedia, vol. 5, no. 3, pp. 466-472, September 2003.

[ZN99] Zerroug M. and Nevatia R., "Part-based 3D descriptions of complex objects from a single image," IEEE Trans. on PAMI, vol. 21, no. 9, pp. 835-848, September 1999.

# Appendix A

## Proof of Theorm 4.1

After $1^{st}$ fragment is transmitted, $\Delta T_1$ is allocated to the remaining $n$-1 fragments as follows:

$\Delta T_1 = \Delta T_1 * (1/(2*\varsigma_1) + 1/(3*\varsigma_1) + \ldots + 1/(n*\varsigma_1))$.

After $2^{nd}$ fragment, $\Delta T_2$ is allocated to the remaining $n$-2 fragments as follows:

$\Delta T_2 = \Delta T_2 * (1/(2*\varsigma_2) + 1/(3*\varsigma_2) + \ldots + 1/((n-2)*\varsigma_2) + 1/((n-1)*\varsigma_2))$,

.....

In the last two allocations,

$\Delta T_{n-2} = \Delta T_{n-2} * (1/(2*\varsigma_{n-2}) + 1/(3*\varsigma_{n-2})$; allocated to $(n-1)^{th}$ and $n^{th}$ fragments,

$\Delta T_{n-1} = \Delta T_{n-1} * (1/(2*\varsigma_{n-1}) + 1/(2*\varsigma_{n-1})$; allocated to $n^{th}$ fragment.

Since there is no other fragment after $n$, the $n^{th}$ fragment has to share 100% of $\Delta T_{n-1}$.

The estimated time $\vartheta_n$ is revised by adding the cumulated compensating time $\Delta t_n$ applied to the $n^{th}$ fragment, and we compute the overall surplus/deficit $\Pi$ as follows:

$\Pi = \vartheta_n + \Delta t_n - \gamma_n$, where $\gamma_n$ is the actual transmission time for $n^{th}$ fragment.

$\Pi = d_n / \beta_0 + (\Delta T_1/(n*\varsigma_1) + \Delta T_2/((n-1)*\varsigma_2) + \ldots + \Delta T_{n-2}/(3*\varsigma_{n-2}) + \Delta T_{n-1}) - d_n / \beta_n$.

$\Pi = (d_n / \beta_0 - d_n / \beta_n) + (\Delta T_1/(n^*\varsigma_1) + \Delta T_2/((n\text{-}1)^*\varsigma_2) + \ldots + \Delta T_{n-2}/(3^*\varsigma_{n\text{-}2}) +$

$\Delta T_{n-1}/(2^*\varsigma_{n\text{-}1})) + (\Delta T_{n-1}/2)$ .

We define $\Pi = E_{est} + E_{comp} + E_{alloc}$ where

$E_{est}$ – Estimation Error $(d_n / \beta_0 - d_n/\beta_n)$ : Equation (A.1)

$E_{est}$ is caused by the discrepancy between the historic average and the actual bandwidth for $n^{th}$ fragment. Note that the $1^{st}$ $d_n$ is before adjustment of $\Delta t_n$, and the $2^{nd}$ is after,

$E_{comp}$ – Compensation Error :

$(\Delta T_1/(n^*\varsigma_1) + \Delta T_2/((n\text{-}1)^*\varsigma_2) + \ldots + \Delta T_{n-2}/(3^*\varsigma_{n\text{-}2}) + \Delta T_{n-1}/(2^*\varsigma_{n\text{-}1}))$

$E_{comp}$ is the time surplus/deficit allocated from fragments 1 to $n$-1, shared by the $n^{th}$ fragment,

$E_{alloc}$ – Allocation Error $(\Delta T_{n-1}/2) = (d_{n-1} / \beta_0 - d_{n-1}/\beta_{n-1})/2$ : Equation (A.2)

$E_{alloc}$ is the time incapable of allocating further.

The time deviation $\Delta T_k$ caused by the $k^{th}$ fragment can be expressed as $\Lambda + \varepsilon_k$ where $\Lambda$ is the average deviation. Let $\varsigma_k$ be $\ln(n-k+1)$, as defined in Algorithm 1, $E_{comp}$ can be further analyzed by splitting $\Delta T_i$ into $\varepsilon_i$ and $\Lambda$:

$E_{comp} = (\varepsilon_1/(n^*\varsigma_1) + \varepsilon_2/((n\text{-}1)^*\varsigma_2) + \ldots + \varepsilon_{n\text{-}2}/(3^*\varsigma_{n\text{-}2}) + \varepsilon_{n\text{-}1}/(2^*\varsigma_{n\text{-}1})) + \Lambda^*(1/(n^*\varsigma_1)$

$+ 1/((n\text{-}1)^*\varsigma_2) + \ldots + 1/(3^*\varsigma_{n\text{-}2}) + 1/(2^*\varsigma_{n\text{-}1}))$, and thus

$E_{comp} \le (\varepsilon_1/(2*\varsigma_{n-1}) + \varepsilon_2/(2*\varsigma_{n-1}) + \ldots + \varepsilon_{n-2}/(2*\varsigma_{n-1}) + \varepsilon_{n-1}/(2*\varsigma_{n-1})) + \Lambda*(1/(n*\varsigma_1) + 1/((n-1)*\varsigma_2) + \ldots + 1/(3*\varsigma_{n-2}) + 1/(2*\varsigma_{n-1}))$, which means that

$E_{comp} = \Lambda(1/(n*\varsigma_1) + 1/((n-1)*\varsigma_2) + \ldots + 1/(3*\varsigma_{n-2}) + 1/(2*\varsigma_{n-1}))$ because

$$\sum_{i=1}^{n-1} \varepsilon_i = 0.$$

Note that: $E_{comp} \ge (\varepsilon_1/(n*\varsigma_n) + \varepsilon_2/(n*\varsigma_n) + \ldots + \varepsilon_{n-2}/(n*\varsigma_n) + \varepsilon_{n-1}/(n*\varsigma_n)) + \Lambda*(1/(n*\varsigma_1) + 1/((n-1)*\varsigma_2) + \ldots + 1/(3*\varsigma_{n-2}) + 1/(2*\varsigma_{n-1}))$.

So we establish: $E_{comp} = \Lambda(1/(n*\ln(n)) + 1/((n-1)*\ln(n-1)) + \ldots + 1/(3*\ln 3) + 1/(2*\ln 2))$.

Since $\frac{1}{x\ln(x)}$ is a continuous decreasing function, the sum can be bounded using integration:

$$\int_3^{n+1} \frac{1}{x\ln(x)} dx \le \sum_{i=3}^{n} \frac{1}{i\ln(i)} \le \int_2^{n} \frac{1}{x\ln(x)} dx.$$

Thus if $\Lambda \ge 0$,

$(\ln|\ln(n+1)| - \ln|\ln(3)| + 1/2\ln(2))\Lambda \le E_{comp} \le (\ln|\ln(n)| - \ln|\ln(2)| + 1/(2\ln(2)))\Lambda$

and if $\Lambda < 0$,

$(\ln|\ln(n+1)| - \ln|\ln(3)| + 1/2\ln(2))\Lambda \ge E_{comp} \ge (\ln|\ln(n)| - \ln|\ln(2)| + 1/(2\ln(2)))\Lambda$

So:

$$E_{comp} \leq |\,(1.088 + \ln|\ln(n)|)\Lambda\,|, \text{ and} \qquad \text{Equation (A.3)}$$

$$\Pi \leq \Delta T_n + (\Delta T_{n-1}/2) + |\,(1.088 + \ln|\ln(n)|)\Lambda\,| \qquad \text{Equation (A.4)}$$

Therefore, we have proved the upper and lower bound of Π in Theorem 4.1.