Optimal Bit Allocation in 3D Compression

Davis King and Jarek Rossignac
GVU Center
Georgia Institute of Technology
kingd, jarek @cc.gatech.edu

Abstract: *To use 3D models on the Internet or in other bandwidth-limited applications, it is often necessary to compress their triangle mesh representations. We consider the problem of balancing two forms of lossy mesh compression: reduction of the number of vertices by simplification, and reduction of the number of bits of resolution used per vertex coordinate via quantization. Let A be a triangle mesh approximation for an original model O. Suppose that A has V vertices, each represented using B bits per coordinate. Given a file size F for A, what are the optimal values of B and V? Given a desired error level E, what are estimates of B and V, and how many total bits are needed? We develop answers to these questions by using a shape complexity measure K that allows us to express the optimal value of B for a general model in terms of V and K alone. We give formulas linking B,V,F,E, and K, and we provide a simple algorithm for estimating the optimal B and V for an existing triangle mesh with a given file size F.*

1. Introduction

1.1 Motivation

Triangle meshes are commonly used for interactive graphic and network applications which involve computer models of 3D objects, manufacturing assemblies, construction sites, geographical or geological datasets, or virtual environments for commercial or entertainment applications. Although other representations exist, triangle meshes are popular because they are supported by many data exchange standards and rendering systems.

A triangle mesh in 3D may be represented by a table or list of its V vertices and a table or list of its T triangles. Each vertex may be represented by its three coordinates. Each triangle may be represented by three vertex-references and is the convex hull of these three vertices. A simple data structure that stores the vertices and the triangles independently would require 3VB bits for the vertices, where B is the number of bits used for each vertex coordinate, and $3T\log_2(V)$ bits for the triangles.

Many applications require that 3D models be accessed over network or telephone connections. In such cases, the naive representation significantly limits the complexity of 3D models that can be downloaded during interactive sessions. A large number of 3D compression techniques have been developed recently [23] to address this problem.

Numerous techniques have been proposed to compress the entire triangle table, instead of representing the triangles independently of each other. Many of these techniques encode the triangles in a specific order to exploit triangle-triangle adjacency. For example, several techniques [14,30,34,8,22] construct a spiraling triangle spanning tree and simply encode the irregularities of that tree.

The vertices are labeled using the order in which they are first encountered in the triangle tree and encoded in the order of increasing label. Predictive techniques, which encode the corrective vector between the actual vertex location and an estimate based on previously decoded neighboring vertices, yield short corrective vectors, if the estimates are good [30,34]. The coordinates of these corrective vectors are compressed using variable length encoding techniques [17]. To further compress the representation of a shape, one can reduce the number B of bits used to represent each vertex coordinate or replace the mesh by a simpler one, which has fewer vertices and triangles.

B may be reduced by quantization [3] as follows. Compute the smallest axis-aligned box that contains the model and define a new coordinate system with a vertex of that box for origin. Chose the units of the new coordinate system, so that each vertex coordinate lies between 0 and 2^B-1, where B is the desired number of bits. Then express each vertex in the new coordinate system and round off their coordinates to the nearest integer. This process amounts to subdividing the box into a regular grid of 2^Bx2^Bx2^B cells and snapping each vertex to the center of the nearest corner of its cell.

The simplest way of constructing a simplified version of a triangle mesh O is to perform the quantization described above and to remove degenerate triangles, which have at least two coincident vertices [20]. Variations on this approach and more complex, yet more effective, simplification techniques are reviewed in [23]. Many of these techniques [12,18,5,6] simplify the model incrementally by collapsing one edge at a time and by discarding the triangles that become degenerate.

1.2 Problem

Let O be a 3D model whose use requires a compressed representation and let A be a triangle mesh approximation for O. A may be compressed both by reducing the number V of its vertices and by quantizing the vertex coordinates to B bits. We investigate how to best balance these two forms of lossy compression and answer the following questions:

1. Given a bound on the file size, F, which values of B and V minimize the error?
2. Given a bound on the geometric error, E, which values of B and V minimize the total number of bits?
3. How does the relationship between B, V, F, and E depend on model shape?

We do not consider the issues involved in choosing the best mesh approximation for a given number of vertices or in choosing a lossless compression in addition to the lossy reduction of V and B. Instead, we wish to understand the relationship between B, V, E, F, and model shape in a way that generalizes to any simplification and compression schemes.

1.3 Our Contribution

The research contribution reported here may be articulated as follows.

Consider the error E(V,B) between some original shape O an its triangle mesh approximation with V vertices, whose coordinates are represented using B vertices each. We assume that the approximation was computed so that the error is roughly uniform over the entire surface (i.e., there are no portions where the approximation is significantly more accurate than at others) and that the vertex quantization to B bits is done using a normalized coordinate system derived from the minimum axis-aligned box.

The error function E(V,B) may be viewed as an elevation over a V-B domain: to each pair of (V,B) coordinates in a given domain corresponds one elevation, E(V,B). The storage cost F is a linear function of VB. It may be viewed as a function F(V,B) over the same domain. We are interested in all points on that surface for which E is minimized for a given F. These points lie along the ridges of the E surface, which follow the steepest descent path. If we are able to identify these ridges and compute their intersections with constant F curves, we can compute optimal V and B values for a given file size or for a given error.

We first solve this problem for a sphere. We identify the ridges B(V) and provide formulae for predicting how they intersect iso-error and iso-size curves. We formulate the error as a Hausdorff distance between a sphere and its triangle-mesh approximation and study how it varies with V and B.

Then we generalize it to arbitrary surfaces through a mapping K, which specifies the number $K(V_S)$ of vertices that would be required to approximate the surface with an error equal to the Hausdorff distance between a sphere of radius one and its triangular approximation with V_S vertices.

We approximate $K(V_S)$ with KV_S and use a simple algorithm to compute K as an integral of the local curvature over the entire surface. When the original surface is a fine tessellation, we estimate its curvature at the mid-point of each edge by fitting a sphere through the four vertices of the incident triangles. We multiply the curvature of the sphere by one third of the area of both triangles and sum up the result for all the edges.

This approximation may be refined by using more accurate curvature estimators, which for example distinguish between the maximal and minimal curvatures at each point. Furthermore, the linear behavior of $K(V_S)$ is only valid when the error is smaller than the surface features or undulations.

We present formulas for B and V in terms of K, F, and E, answering the questions above. We validate these formulas empirically by presenting numerical computations of the error of a tessellation of the sphere, and by exploring quantized versions of standard 3D models.

1.4. Organization

The paper is organized as follows. We compare our analysis to prior work on vertex quantization and mesh error in the next section. In section 3, we outline our definition of error and present the general form of the equations for error and file size that we will use in our subsequent optimization. Section 4 discusses shape complexity and introduces the parameter K. Section 5 presents the analysis of the sphere, and Section 6 gives the actual formulas for computing B, V, E, and F. Section 7 outlines the algorithm used to estimate K, and Section 8 presents our empirical results.

2. Prior Art

2.1. Optimal bit allocation

Bit quantization was used for mesh simplification [20] and for compression [3,12,30,8,34], but the number of bits used for compression was selected by the operator, through visual criteria and trial-and-error. Chow [2] provides an algorithm for selecting the quantization level, based on testing each triangle in a model against the size of the coordinate grid.

Li and Li [15] have focused on a progressive transmission of triangle meshes and have suggested that vertices be represented with fewer bits in the initial stages of the model than in the final stages. They provide a formula for deciding whether the next batch of bits should be used to refine the triangulation or to further constrain the vertex locations. Their formula ,derived for their specific progressive transmission model, depends on information from the vertex reduction stage. We have generalized this principle to arbitrary compression techniques and to non-progressive compression techniques and have provided analytic relations between B, V, E that depend only on the shape factor K instead of on data generated during the simplification.

2.2. Shape Complexity

Several other authors have considered the relationship between V, E, and model shape. [35] identifies the local curvature as a main factor in influencing the distribution of vertices in an optimized mesh; and [1] uses the distance to the medial axis as a local measure that incorporates both curvature and the thinness of model sections, although they do not analyze how V and E relate to these shape measures. Garland, in his PhD thesis [6], relates a quadric error estimate to the local curvature. Our shape factor K incorporates the effects of curvature and other shape characteristics in a way that can be quantified, estimated, and analyzed both locally and globally. Garland and Heckbert study how curvature influences the shape of the approximating triangles [9].

Nadler [17] provides a theoretical analysis of the asymptotic behavior of piecewise linear function approximations. His work leads to an integral similar to the one from which we derive our shape characteristic. He expresses the asymptotic limit of the product of the approximation error and the number of vertices as an integral of the Hessian determinant of the function, although he does not comment on the integral's potential applications. Apart from the mathematical differences between functional approximation and 3D modeling, the main difference between our results and Nadler's is that our formulation of K allows it to be computed in practice from a mesh even if the original model is unknown. K also allows us to consider how the relationships among V, E, and K behave for meshes with small V and for meshes generated by practical optimization algorithms.

3. Error and Filesize Equations

In our analysis of the B-V tradeoff, we assume the existence of an original model, O, that we wish to compress and of a family of approximating meshes for O, each with different numbers of vertices and each generated by a process minimizing the tessellation error before quantization. Such a family of triangle meshes at different levels of detail may be called a uniform multi-resolution model.

3.1 Total Storage Cost of a Compressed Model

The total size in bits of a compressed mesh may be written as a function of:

- V, the number of vertices
- B, the number of bits per vertex coordinate
- T, the number of bits per vertex needed to encode the triangle connectivity
- and α , a compression factor for vertex coordinates

We write:

Equation 1 $F = (3\alpha B + T)V$

To assign specific values to these parameters, we assume for simplicity that T=≈2V, which holds for meshes that are the boundary of a manifold solid with relatively few handles and holes. The triangle table may be compressed [22] to less than 2 bits per triangle, or equivalently to 4 bits per vertex. For typical applications, B is an integer between 6 and 14. For complex but regular models, α varies from 1 for small meshes and 1/3 for large meshes. The total storage cost is: 3α BV+4V bits and hence varies between 10V bits and 46V bits, depending on the desired vertex accuracy and compression ratio.

3.2 Approximation error due to surface tessellation

Consider a surface O that may be smooth or finely tessellated. Let A be a triangle mesh of V vertices that approximates O. The discrepancy between O and A is called the tessellation error and will be denoted E_T. It may be measured in several ways.

For example, one may be interested in the volume of the symmetric difference between the solid bounded by A and the solid bounded by O [6,16]. Because a small volume in the symmetric difference does not guarantee a small deviation between one surface and the other, we prefer to formulate the deviation as the Hausdorff distance H(O,A), which is usually defined as the maximum of the distance between a point on one of these two surfaces and the other surface.

Note that, in general, H(O,A) may not be computed by only considering distances between vertices and edges of one set and the other set. To illustrate this point, consider the equivalent definition of H(O,A) as the minimum r for which $O \subset A\uparrow r$ and $A \subset O\uparrow r$, where $X\uparrow r$ is the offset of the set X by a distance r [20] or equivalently is the Minkowski sum of X with a ball of radius r centered at the origin 25. The maximum deviation may happen at a point c in the interior of a face of O, such that the open ball of center c and radius r does not intersect A. Thus, it is necessary to consider quadruples of faces when computing H(O,A).

Because the exact Hausdorff distance is expensive to compute, upper error bounds or least-square estimators have been used. Hoppe [13] used a set of sampling points on each face, Ronfard and Rossignac [18] used deviations from supporting planes, Gueziec [6] used bounding spheres, Klein and Strasser [28] used a geometric bound, Heckbert and Garland [5] estimate the error using a least square distance to the supporting planes of Ronfard and Rossignac.

Typically, the approximating mesh, A, is generated through a mesh simplification or curved surface tessellation process, which attempts to remove all those vertices that can be removed without exceeding the prescribed bound on the tessellation error. Note that some simplification algorithms remove vertices by collapsing them onto other vertices, while other approaches [13] optimize the final location of the retained vertices.

We will not assume that these meshes are optimal, but only that they are 'optimized' well enough that their overall error will be increased by rounding off the vertex coordinates and by removing any vertices through an additional simplification.

Clearly, increasing the maximum error reduces the number of vertices in A. The precise nature of this relation depends on the shape of O and has been studied empirically on several examples [16]. Such experimental results could be used to compare different simplification techniques [16,3]. E-V plots produced by optimized simplification techniques [13] for a set of benchmark objects could provide an absolute reference against which new simplification techniques could be measured.

Instead, we use analytical relations for E and V. An equation for E and V may be written explicitly for a uniform tessellation of a sphere. For more complex shapes, we use the shape factor K. This simplification allows us to use analytic optimization methods to derive equations for B and V.

3.3 Quantization error

As discussed above, popular compression techniques rely on vertex quantization. We define the quantization error, Eq, to be the increase in error due to the truncation of the vertex coordinates to B bits each, after they have been mapped into the normalized coordinate system. For an optimized mesh as defined above, Eq will never be negative, since quantization does not in general decrease the error.

Equation 2 $E_Q \leq S\dfrac{2^{-B}\sqrt{3}}{2}$**, where S is the size of the coordinate grid in units of length**

As noted above, computing the Hausdorff error is not trivial and quantization may change both the errors at each point and which point in the mesh is closest to which point in the model. At each point, however, the maximum possible increase in error may be bounded by the distance by which quantization displaces the point. The total error E, that we define as H(O,A), is therefore bounded by E_Q+E_T.

Equation 3 $E \leq E_Q + E_T$

3.4 Keeping the quantization and tessellation errors in synch

One could naively set E_Q=E_T and use the formulae mentioned above to establish V and B for A, given O and E. Such a choice may be motivated by the desire to balance both errors: there is no point over-specifying the vertex coordinates when the tessellation error dominates and a simplification is sub-optimal when the quantization error dominates. Knowing B and V suffices to estimate the total number of bits needed for A.

Often, the overall objective of compression is to minimize the total error E, without exceeding a total bit-count chosen based on the network capacity or storage space available. The solution to this optimization problem often differs from the above naive guess, and it is easier to apply to arbitrary shapes. We derive an analytic expression for computing optimal B and V, given K and F.

In other situations, it may be important to bound the allowed error and to find the values for B and V that minimize the file size F. Again, we provide analytical formulae for extracting such values.

Figure 1 Plot of Isoerror and Isofilesize contours

The two optimization questions described above are equivalent to a simple two-dimensional optimization procedure based on the gradient of error. This approach may be understood by considering an error surface for E as a two-dimensional function of B and V. In the three-dimensional space defined by B, V, and E, each value of F defines an isosurface that intersects the error surface in a curve of (B,V) pairs. The locally optimal (B,V) values for each F occur where the gradient of E is orthogonal to the constant-F curve. At any point where the gradient of E is not orthogonal to the iso-F curve, one may find a lower error for the same filesize and a different (B,V) pair by moving in the direction of the projection of the gradient onto the isocurve. To find an algebraic relation for the optimal (B,V) pairs where the gradient is orthogonal to the isosurface, one may simply set the slope of the gradient vector to equal the negative inverse of the slope of the tangent to the constant-F curve:

$$\frac{\frac{\partial E}{\partial B}}{\frac{\partial E}{\partial V}} = -\frac{1}{slope} = -\frac{\frac{\partial F}{\partial V}}{\frac{\partial F}{\partial B}} = \frac{daV}{daB+T}$$

Equation 4 $$\frac{\partial E}{\partial V}V = \frac{\partial E}{\partial B}\left(B + \frac{T}{da}\right)$$

This is equivalent to solving for B and V such that the marginal contribution of a bit of additional B or a bit of additional V is equal; this is the criterion [14] suggests. Note that it does not mean that the tessellation and quantization errors are equal, but that the increases in error are equal. Depending on the shape of E, this equation may be satisfied at more than one locally optimal (B,V) pair for a given value of F. In such a case, the pair with the lowest value of E is the global optimum.

4 Shape Complexity

$\frac{\partial E}{\partial V}V$, the marginal effect of a vertex addition, depends on the details of the model's shape. Consider a 3D model of a Russian doll, with several shells of increasingly smaller, inner offset surfaces. Clearly, its representation will require more vertices for the same error and for the same bits of coordinate resolution than a model of the outer shell alone. Also, the error function may not be differentiable in a strict mathematical sense, since the addition of only a single new vertex to one of a model's connected components may not decrease the error of the model as a whole. These same considerations apply to multiple regions of a connected surface, on which different curvature may affect the requirements for V.

4.1 The Shape Factor K

Our approach is to investigate the error on small regions of the original model O, and then to apply a shape function $K(V_S)$ which measures the global shape complexity. $K(V_S)$ relates the number of vertices $V=K(V_S)$ used to approximate O to the number of vertices V_S needed to represent a sphere with the same error as the approximation of O. We derive an expression for $K(V_S)$ by stating a uniformity condition relating the error in each local region of O to the total error, and by approximating each such region as a portion of a sphere.

Suppose that O is a piecewise spherical surface composed of spherical patches Si. Let Ai be the surface area of patch Si and let ri be its radius. Let $V_s(E)$ be the number of vertices needed to approximate a sphere of radius 1 by a triangulation with a uniformly distributed tesselation error E. The number of vertices needed to approximate a sphere of radius ri is $V_s(E/Ri)$. The number of vertices, Vi, needed to approximate a portion of such a sphere that has area Ai is $A_i \Big/ \left(4\pi r_i^2 V_s\left(E/r_i\right)\right)$. Let A be a triangular mesh approximating O such that the tessellation error is uniform throughout the surface and significantly lower than all ri. Assume that patch Si is associated with Vi vertices of A. The total number of vertices of A is the sum of Vi. Using the above formula for Vi yields:

Equation 5 $$V = \sum_O \left(\frac{A_i}{4\pi r^2{}_i} V_S \left(\frac{E}{r_i} \right) \right)$$

Nadler [18] proves that in the limit as V approaches infinity, VE converges to an integral over the surface of O. For a sphere, a taylor series expansion of the formula for tessellation error on a sphere (Equation 10) gives a formula which may be approximated as Vs=X*ri/E for ri>>E. Using this result, we may factor ri out of the above equation to get:

Equation 6 $$V = \sum_O \frac{A_i X}{4\pi r_i E}$$

We factor X/E out and define K to be sum(Ai/Ri) The following formulas may be used to modify equations derived on the sphere into equations applicable to arbitrary shapes, where S is a scale factor in units of length.

Equation 7 $$V_S = \frac{4\pi}{K} V = \frac{4\pi S}{\iint_O \frac{1}{r_i} dSA} V$$

Equation 8 $$K = \frac{1}{S} \iint_O \frac{1}{r_i} dSA$$

Our piecewise spherical approximation of the original shape may be easily derived by fitting spheres to all pairs of adjacent triangles and by considering that the associated patch has an area equal to a third of the total area of both triangles. This approximation of the model as a collection of spherical patches is not intended to give an accurate prediction of the true model's surface, but to give a first-order estimate of the behavior of the many possible true surfaces of O. Indeed, more accurate methods are available for predicting a surfaces from a triangle mesh [31]. For the purposes of analyzing quantization, spheres are the simplest structures that allow an analysis of local curvature. Since they are a first-order approximation to the error, they are likely to capture the effects of the largest factors affecting local error. However, this approximation does not capture the precise relation between V and E in areas where the surface has uneven principal curvatures and in areas where ri does not significantly exceed the error.

4.2 Computing K

The shape complexity of a mesh A may be estimated by considering each edge **e** of A. In a manifold mesh, **e** has exactly two incident triangles. The four vertices that bound these triangles define a possibly infinite sphere. When finite, the radius of this sphere may be computed from the following determinant equation, where A, B, C, D are the four points, and b, c and d are vectors defined as (B-A), (C-A), and (D-A) respectively, with bx, by, etc. the components of the vectors [25]:

Equation9 $$r = \frac{\left\| (d)\bullet(d)\left[(b)\times(c)\right] + (c)\bullet(c)\left[(d)\times(b)\right] + (b)\bullet(b)\left[(c)\times(d)\right] \right\|}{2\begin{vmatrix} bx & by & bzz \\ cx & cy & cz \\ dx & cy & dz \end{vmatrix}}$$

We associate with **e** the radius r(**e**) of that sphere and an area p(**e**) that is a third of the sum of the areas of its two incident triangles. K may therefore by computed by summing p(**e**)/r(**e**) for all edges **e** of O.

Quantization on the Sphere

5.1 Tessellation Error on a Sphere

The Hausdorff error, E_T, for a uniform triangulation of a sphere of radius r using an optimal mesh of V_S vertices interpolating the surface of the sphere may be written as:

Equation 10 $$E_V = r\left(1 - \frac{1}{\sqrt{3}}\sqrt{1 + 2\cos\left(\frac{2C}{\sqrt{V}}\right)}\right)$$

where $C = \sqrt{\frac{\pi}{\sqrt{3}}}$ and $\frac{2C}{\sqrt{V_S}} = \theta$ is the angle between a pair of lines from the center of the sphere to two of the triangle's vertices.

Figure 2

This formula is derived by computing the distance from the surface of the sphere to the center of an average-sized equilateral triangle in the interpolating mesh. As shown in Figure 2, we may compute this distance as:

$$d = |L - r| = r - r\cos(\arcsin\left(\frac{2}{\sqrt{3}}\sin(\frac{\theta}{2})\right)) = r - r\sqrt{1 - \frac{4}{3}\sin^2(\frac{\theta}{2})}$$

θ may be found by assuming the model has $2V_S$ triangles, assuming that each triangle has equal area, and then setting equal the area of a spherical triangle as a function of θ and the area of the same triangle as a function of V:

Equation 11 $$A_{patch} = \frac{r^2\theta^2\sqrt{3}}{4} = \frac{4\pi r^2}{2V_S} \Leftrightarrow \theta = 2\sqrt{\frac{\pi}{V_S\sqrt{3}}} = \frac{2C}{\sqrt{V_S}}$$

Note that the above formulas describe the behavior of an 'average' equilateral triangle on the sphere. In practice, uniform triangulations of the sphere into 2 V_S equilateral triangles exist only for a handful of values of V_S, corresponding to Platonic and other special solids. Even distributing vertices uniformly on the sphere is a complex mathematical problem, discussed in depth in [9]. This discrepancy, however, is small relative to other simplifications involved in using spheres as a first-order approximation of a surface.

5.2 Derivation of the marginal error term

To solve for the optimal values of B and V, we assume the error equals its bound of $E_T + E_Q$. We substitute the formulas for E_T and E_Q given in Equation 2 and Equation 10 to obtain an equation for E, and we take partial derivatives with respect to B and V as follows:

$$E = E_B + E_V = S\frac{2^{-B}}{2/\sqrt{3}} + S\left(1 - \frac{1}{\sqrt{3}}\sqrt{1 + 2\cos(\frac{2C}{\sqrt{V_S}})}\right)$$

$$\frac{\partial E}{\partial B} = -\frac{S}{2/\sqrt{3}}\ln(2)2^{-B}$$

$$\frac{\partial E}{\partial V} = -\frac{r}{\sqrt{3}}\left(1 + 2\cos(\frac{2C}{\sqrt{V_S}})\right)^{-1/2}\left(\sin(\frac{2C}{\sqrt{V_S}})\right)\left(CV_S^{-3/2}\right)$$

6.1 Formulas for Optimal Values of B and V

We use the relation $V_S = \frac{4\pi}{K}V$ to modify the above equations for a sphere represented with V_S vertices to apply to a more complex shape O approximated with V vertices. We derive relations between B, V, and K by substituting the derivative equations above into Equation 4, manipulating the terms algebraically, and taking the logarithm to the base 2.

Equation12: the full equation

$$B-\log(B+\frac{T}{3\alpha})=\frac{1}{2}\log(\frac{V}{K})-\log\left(\sin(\frac{\sqrt{K}}{\sqrt{V\sqrt{3}}})\right)+\frac{1}{2}\log\left(1+2\cos(\frac{\sqrt{K}}{\sqrt{V\sqrt{3}}})\right)+\frac{3}{4}\log(3)+\log(\ln(2))$$

The sum of the -log(sin) and ½ log(cos) terms converges rapidly to ½ log(V/K)+1.189, with 1% error for V/K=1. Replacing the constant terms $\frac{3}{4}\log(3)+\log(\ln(2))+1.189$ with a constant B_0=1.838, we get:

Equation 13: simplified equation

$$B-\log(B+\frac{T}{3a})=\log(\frac{V}{K})+B_0$$

6.2 Formulas Relating B, V, F, E, and K

The equations for B-V above may be used to write equations for B and V in terms of constraints on the maximum file size or maximum acceptable error for a model. The following formulas are computed from the expressions for K, F, and E above, in the asymptotic limit as V increases. Note that for the expressions involving error, the error values are based on our spherical model and optimality assumptions. As a result, the error is likely to differ from the true geometric error of a mesh with V vertices, which depends on the granularity of the model's features and on constraints used in mesh generation, such as whether vertices are approximating or interpolating (see [24]). These equations are therefore best used to make decisions about the relative error of different levels of detail. Using them to estimate absolute error may require calibration against more detailed information on the original model.

Equation 14: B as a function of filesize $B=\log(\frac{F}{K\alpha})+B_0$

Equation 15: V as a function of filesize $V=\frac{F}{3\alpha B+T}=\frac{F}{3\alpha\log(\frac{F}{K\alpha})+T+3\alpha B_0}$

Equation 16: Filesize as a function of B $F=K\alpha 2^{B-B_0}$

Equation 17 The E-V-K relation $\frac{E_T}{S}=1-\frac{\sqrt{1+2\cos\left(\frac{\sqrt{K}}{\sqrt{V\sqrt{3}}}\right)}}{\sqrt{3}}\approx 1-\frac{\sqrt{3-\frac{K}{V\sqrt{3}}}}{\sqrt{3}}$

Equation 18: The V-E-K relation $V\approx\frac{K}{6E_T/S+2\sqrt{3}}\Rightarrow\frac{E_T}{S}=\frac{K}{6V}-\frac{1}{\sqrt{3}}$

Equation 19: B from tessellation error $B - \log(B + \frac{T}{3a}) = -\log(Et / S + 1/\sqrt{3}) + B_0 - 1$

7. Algorithm for computing B and V

The following simple algorithm may be used to compute B and V for a given mesh:

1. Identify constraints -- either a fixed value of V, a limit on file size F, or a maximum allowed error E
2. Compute K as described above
3. Plug the constraint and the value of K into the formulas above to compute optimal values of B and V
4. Select from the meshes available the one closest to the optimal value of V, or generate a mesh with as close to V vertices as possible.
5. Represent vertex coordinates with B bits each in the normalized coordinate system
6. If additional application-specific error information is available, find the error of the chosen mesh; if it does not satisfy the constraint, adjust the value used as a constraint to compensate, and repeat.

8. Experimental validation

Figure 3 Error plot for a sphere as a function of triangulation and quantization

We validate our analysis of the sphere by computing equilateral triangles with vertices on a unit sphere, quantizing their coordinates, and measuring the resulting error as the distance from the midpoint of each triangle to the sphere's surface. The computations were performed in an Excel spreadsheet, with the quantization implemented by rounding to appropriate powers of 2. The vertex coordinates were computed as follows, with the angle θ formed by any two vertices and the center of the sphere computed from V by Equation 10, ϕ chosen to make the triangles equilateral, and with θ_0 (an arbitrary starting position) and z_0 (a random displacement of the sphere's center from the origin) chosen differently for each triangle.

Equation20
$$\begin{aligned} v1 &= (\cos\theta_0, \sin\theta_0, z_0), \\ v2 &= (\cos(\theta_0 + \theta), \sin(\theta_0 + \theta), z_0), \\ v3 &= (\cos(\theta_0 + 0.5\theta)\cos\phi, \sin(\theta_0 + 0.5\theta)\cos\varphi, z_0 + \sin\phi)) \end{aligned}$$

Figure 2 shows the resulting error surface as a function of B and V for a single triangle. Figure 3 plots the maximum error for a set of three triangles at different positions.

Figure 4 shows the error surface for three triangles as a function of B and a constraint on F. The constraint makes the optimal (B,V) curve much more prominent by imposing a severe penalty on excess B. Figure 5 shows a plot of the optimal values of B selected from the error surface in Figure 4. It confirms the relationship between B and log(V) predicted above.

Figure 4 Error plot for a sphere as a function of triangulation and file size

Figure 5 Plot of optimal B and optimal V

To validate the application of our results to more complex shapes, we also tested the predictions of our algorithm on simplified versions of actual 3D models. We computed K for the models, rounded their coordinates to integer values from $-2^{B-1} + 1$ to 2^{B-1}, and inspected the results in an Inventor viewer in order to check for the visual artifacts. Note that the point where quantization artifacts appear may not be the same as the point at which B becomes lower than the optimal value; it is theoretically possible for quantization artifacts smaller than the tessellation error to be visible, or for the quantization error to exceed an error bound without producing visual artifacts. Checking for visual artifacts, however, is necessary to confirm that the recommended values of B produce acceptable visual results. Furthermore, our tests have revealed that although the critical value of B where distortions appear may not be exactly the same as the predicted optimal B, the critical values follow the log(V/K) relationship that our formula predicts.

Figure 6 and Figure 7 show this log(V/K) pattern. In Figure 6 a row of bunny ears from models simplified to different V shows that the distortions appear at lower B values for models with lower V. Figure 7 shows that the quantization artifacts appear at the same log(V/K) values for different models with the same surface area but different values of K. The hand shown has V~4000 and K~42, while the horse has V~5000 and K~56. For both, quantization becomes visible in a view of the whole object at B=6, while smaller distortions are visible in a zoomed view at B=7. In the other part of Figure 6, the sphere shown is chosen to have the same (V/K) as a horse with a much larger number of vertices. In both cases, distortions appear at the same value of B. We note as well that for each model, there is a value of B for which no distortions are visible at any magnification, since any quantization effects are small relative to the smallest triangles. These are the values used for the second row of bunny ears in Figure 6.

Figure 6 Closeup views of bunny ears

Figure 7 Comparisons of different models

Figure 8 Locations of views in Figures 6 and 7 on an Isoerror/Isofilesize plot

The simplified models used in these experiments were produced with an implementation of Hoppe's mesh optimization [13], although we have tested the relationship on other simplification methods as well. To compare the predicted and the observed values of B more accurately, we computed K for these models and used it to estimate B according to formulas and algorithm above, with the results presented in Table 1. To compute K, we used the highest level of detail models available, scaled to give each model the same surface area. In general, however, K is reduced slightly as successive simplifications remove small-r features from an object. Understanding how K varies with V for different models is an area for future research, perhaps expressing the relationship in terms of a fractal dimension or a histogram of different sized features instead of a single parameter.

Table 1 Simplified and quantized models used in Figures 6 and 7

MODELS USED IN FIGURES	VERTICES	K	VS	OBS. B/ NO JAGGIES	OBS. B -- SOME JAGGIES	OBS. B -- MANY JAGGIES	LABEL IN FIG. 8
Bunny	15157	54.39	985	10	9	8	1a,b,c
Bunny	7384	54.39	480	8	7	6	2 a,b,c
Bunny	3286	54.39	214	8	7	6	3 a,b,c
Bunny	1499	54.39	98	7	6	5	4 a,b,c
Bunny	701	54.39	46	7	6	5	5 a,b,c
Bunny	332	54.39	22	6	5		6 a,b,c
Bunny	174	54.39	11	6		5	7 a,b
Bunny	83	54.39	5	5		4	8 a,b
Horse	10375	56.07	655	7	6		HS
Sphere	632	3.53	632	7	6		HS

Horse	4997	56.07	315	7	6	5	HH
Hand	4172	42.95	344	7	6	5	HH

One conclusion from these experiments indicates that a mesh with more vertices may produce *worse* images than a mesh with less V if the quantization is too strong (i.e., too few bits). The assumption that adding vertices never increases error does not hold for a quantized mesh. To understand this phenomenon, consider a vertex inserted into a triangle and constrained to lie in the plane of that face, so that its insertion does not affect the Hausdorff error under high-precision coordinates. In limited-precision integer coordinates, it may not be possible to compute a vertex position constrained to lie inside that face. Intuitively, the undesirability of adding too many vertices for a given quantization level may be compared to the Pauli exclusion principle, which restricts the number of electrons in an atomic energy level according to the number of quantized states available for holding them. Likewise, for a curved surface represented with a triangle mesh, there may not be enough quantized positions available near the surface to fit additional vertices without some of them increasing the error.

This suggests an intuitive explanation for why the quantization level must be related to the number of vertices in the mesh. Simplification of a mesh to a particular number of vertices can be viewed as a low-pass filter that eliminates high-frequency components of the mesh, corresponding to features the size of a single triangle or smaller. If a quantized mesh has more V relative to B than the equation predicts, the quantization will distort the smaller/higher-frequency triangles, leading to aliasing. If, however, we choose a mesh that has simplified to the appropriate number of vertices and overall uniform error, the simplification ensures the removal of any high-frequency components that might produce jaggies. Both figures show the jaggies that appear when we choose too high a V for a given value of B.

9. Conclusions

How many bits do we need to approximate, using a triangle mesh, a given 3D surface, O, with an error that does not exceed a given tolerance E? We define this number of bits to be the optimal storage cost for O, given E. We devise a simple and efficient algorithm that estimates the complexity K for objects bounded by triangle meshes. Then, we formulate the optimal storage size as a function F1(K,E) of K and E.

The storage cost of a compressed version of a triangle mesh A is also function F (V,B) of the number, V, of its vertices and of the number, B, of bits used in the uncompressed representation of its vertex coordinates. We formulate the error E resulting from using A as a substitute for O in terms of the Hausdorff distance between them. That error is bounded by the sum, E_t+E_q, of the tessellation error E_t, resulting from the use of only V optimally placed vertices, and of the quantization error E_q, resulting from rounding the vertex coordinates to B bits each.

Combining both formulations of the file size leads to expressions of optimal B and V for a given error.

We use these formulation to provide answers to the following two questions. Consider an optimal approximation A of a surface O.

- Given a limit F(V,B) imposed by constraints on file size or bandwidth, which choices of B and V minimize the total error E?
- Given a relative error bound E imposed by a geometric accuracy requirement, which choices of B and V minimize F(V,B)?

We hope that these answers will provide a framework for improving, evaluating and comparing simplification and compression results.

10. Future Work

Our analysis of vertex quantization may be extended to address progressive refinement and compression schemes using variable quantization levels. A major issue in adapting it to progressive refinement is to determine how best to update the quantization levels of the already-received vertices, as well as to optimize the transmission of the next batch of vertices.

The shape factor K may also be useful for investigating the intrinsic complexity of shapes. Since it may be used to estimate the total number of bits needed to represent a shape within a given accuracy, it may be interpreted as a measure of the information content of a 3D model. Measures of shape complexity may be useful in many other applications such as vision, CAD/CAM, and biology, as well as in designing heuristics for 3D compression and simplification. We plan to study how K relates to other measures of shape and curvature.

11. Acknowledgments

Thanks to Peter Lindstrom and Greg Turk for providing us with simplified models and error results from their paper [16] for our use. Thanks to the Stanford Computer Graphics Library for the bunny model. This work was supported by NSF grant 9721358.

12. Bibliography

[1] N. Amenta, M. Bern, and M. Kamvysselis, “A New Voronoi-Based Surface Reconstruction,” Proc. ACM SIGGRAPH’98, 415-421, 1998.

[2] M. Chow, “Optimized Geometry Compression for Real-time Rendering,” Proc. IEEE Visualization'97, pp. 347-354, Phoenix, AZ, October 19-24, 1997.

[3] P. Cignoni, C. Rocchini, and R. Scopigno. “Metro: Measuring Error on Simplified Surfaces,” Technical Report B4-01-01-96, Istituto I. E. I.-C. N. R., Pisa, Italy, January 1996.

[4] M. Deering, “Geometry Compression,” Computer Graphics, Proc. ACM Siggraph'95, 13-20, August 1995.

[5] M. Garland and P. Heckbert, “Surface simplification using quadric error metrics,” Proc. ACM SIGGRAPH'97. pp. 209-216. 1997.

[6] M. Garland, *Quadric-Based Polygonal Surface Simplification*, PhD Thesis, Carnegie Mellon University, 1998.

[7] A. Gueizec, “Surface Simplification with Variable Tolerance,” in Second Annual Intl. Symposium on Medical Robotics and Computer Assisted Surgery (MRCAS ‘95) pp. 132-139, November 1995.

[8] S. Gumhold and W. Strasser, “Real Time Compression of Triangle Mesh Connectivity.” Proc. ACM Siggraph 98, pp. 133-140, July 1998.

[9] H. Hardin, N. J. A. Sloane and W. D. Smith, Spherical Codes, book in preparation, http://www.research.att.com/~njas/electrons/dim3/

[10] P. Heckbert and M. Garland, “Optimal Triangulation and Quadric-Based Surface Simplification,” submitted for publication.

[11] P. Heckbert and M. Garland, “Survey of Polygonal Surface Simplification Algorithms,” in *Multiresolution Surface Modeling Course*, ACM Siggraph Course notes, 1997.

[12] H. Hoppe, “Progressive Meshes,” Proceedings ACM SIGGRAPH'96, pp. 99-108, August 1996.

[13] H. Hoppe, T, DeRose, T, Duchamp, J, McDonald, and W. Stuetzle, “Mesh optimization” Proceedings SIGGRAPH'93, pp:19-26, August 1993.

[14] K. Keeler and J. Westbrook, “Short Encodings of Planar Graphs and Maps,” Discrete Applied Mathematics, No. 58, pp. 239-252, 1995.

[15] J. Li, J. Li and C.C Kuo, "Progressive Coding of 3D Graphic Models," IEEE Proc. Multimedia Computing and Systems, pp. 1052-1063. June 1998.

[16] P. Lindstrom and G. Turk, "Fast and Memory Efficient Polygonal Simplification." IEEE Visualization '98, Raleigh, NC, 1998.

[17] E. Nadler, "Piecewise Linear Best L_2 approximation on triangulations." In C. K. Chui et al., editors, Approximation Theory V, pages 499-502, Boston, 1986, Academic Press.

[18] W. Press, S Teukolsky, W. Vetterling, B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press.

[19] R. Ronfard. and J. Rossignac, "Full-range approximation of triangulated polyhedra," Proc. Eurographics'96 , Computer Graphics Forum, pp. C-67, Vol. 15, No. 3, August 1996.

[20] J. Rossignac & A. Requicha, "Offsetting Operations in Solid Modeling," Comput. Aided Geom. Design 3, No.2, 129-148, August 1986.

[21] J. Rossignac and P. Borrel, "Multi-resolution 3D approximations for rendering complex scenes," pp. 455-465, in Geometric Modeling in Computer Graphics, Springer Verlag, Eds. B. Falcidieno and T.L. Kunii, Genova, Italy, June 28-July 2, 1993.

[22] J. Rossignac, "Edgebreaker: Compressing the incidence graph of triangle meshes," Jarek Rossignac, GVU Technical Report GIT-GVU-98-17, Georgia Institute of Technology, http://www.cc.gatech.edu/gvu/reports/1998.

[23] J. Rossignac, "Geometric Simplification and Compression", in *Multiresolution Surface Modeling Course*, ACM Siggraph Course notes 25, Los Angeles, 1997.

[24] W. Seibold and G. Wyvill, "Too Many Triangles", SIGGRAPH 98 Conference Abstracts and Applications, p. 242.

[25] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic Press, 1982.

[26] J. Shewchuk, "Re: circumsphere," April 1, 1998, compgeom-discuss@research.bell-labs.com

[27] J. Snoeyink and M. van Kerveld, "Good orders for incremental (re)construction," Proc. ACM Symposium on Computational Geometry, pp. 400-402, Nice, France, June 1997.

[28] R. Klein and W. Strasser, "Generation of multiresolution models from CAD data for realtime rendering." In W. Strasser, R. Klein, R. Rau, editors. Theory and practice of Geometric Modeling. Springer-Verlag, 1996.

[29] G. Taubin and J. Rossignac, *3D Geometry Compression,* Course Notes 21, Siggraph 98, Orlando, Florida, July 18-24, 1998.

[30] G. Taubin and J. Rossignac, Geometric Compression through Topological Surgery, ACM Transactions on Graphics, Volume 17, Number 2, pp. 84-115, April 1998.

[31] G. Taubin, "Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation," Proc. Fifth International Conference on Computer Vision '95, pp. 902-907, June 1995.

[32] G. Taubin, A. Gueziec, W. Horn, F. Lazarus, Progressive Forest Split Compression, Proc. ACM Siggraph 98, pp. 123-132, July 1998.

[33] G. Taubin, W. Horn, F. Lazarus, and J. Rossignac, "Geometry Coding and VRML," Proceedings of the IEEE, pp. 1228-1243, vol. 96, no. 6, June 1998.

[34] C. Touma and C. Gotsman, "Triangle Mesh Compression," Proceedings Graphics Interface 98, pp. 26-34, 1998.

[35] G. Turk, "Re-tiling polygonal surfaces," Computer Graphics, 26(2):55-64, July 1992.

[36] E. Weisstein, The CRC Concise Encylopedia of Mathematics, Boca Raton, FL, 1998, CRC Press. http://www.astro.virginia.edu/~eww6n/math/math.html