Progressive Compression and Transmission of Arbitrary Triangular Meshes [*]

Chandrajit L Bajaj Valerio Pascucci Guozhong Zhuang
Department of Computer Sciences
University of Texas at Austin, Austin, TX 78712

Abstract

The recent growth in the size and availability of large triangular surface models has generated research in compact multi-resolution progressive representation and data transmission. An ongoing challenge is to design an efficient data structure that encompasses both compactness of a geometric representation and visual quality of a progressive representation.

In this paper we introduce a topological layering based data-structure and a encoding scheme to build a compact progressive representation of an arbitrary triangular mesh (a 2D simplicial complex in 3D) with attached attribute data. This compact representation is composed of multiple levels of detail that can be progressively transmitted and displayed. The global topology, which is the number of holes and connected components, can be flexibly changed among successive levels while still achieving guaranteed size of the coarsest level mesh for very complex models. The flexibility in our encoding scheme also allows topology preserving progressivity.

1 Introduction

The design of compact progressive representations for triangular surface models has been a fundamental topic in recent research because of the continual growth of internetworked 3D graphics. High resolution scanning devices for object reconstruction or simulations on supercomputers have resulted in ever increasing generation and use of large surface models in scientific visualization. Geometry compression is currently being worked into the MPEG 4 3D coding standard by the MPEG-SNHC working committee [24].

The primary difficulty of designing an effective progressive and compact encoding scheme is to meet contradictory requirements like progressivity, compression capability and visual efficacy. Such schemes are therefore evaluated by their trade-offs and flexibility in prioritizing the different properties or adaptation to the needs of the particular visualization applications.

Representation Power. The representation power is the class of domain models that a scheme can correctly handle. Different schemes can represent different surfaces that could be open or closed, simple or high-genus, orientable or non-orientable, manifold or non-manifold. For instance, isosurfaces of physical functions generated by a scientific simulation can be high genus and non-manifold. Analytic mathematical Surfaces can be non-orientable, like the Möbius strip or Klein bottle. Increasingly 3D CAD modeling systems include the creation and processing of non-manifold geometries.

Compression. The representation of large surfaces for storage or network transmission needs be space efficient. Compressed representation can make effective use of disk space and network bandwidth as well as substantially reduces network transfer time.

[*]This research is supported in part by grants from NSF-CCR-9732306, NSF-KDI-DMS-9873326, DOE-ASCI-BD-485, and NASA-NCC 2-5276.

Progressivity. A progressive representation has several advantages. First, it enhances high performance interactivity with large remotely archived models because the compressed data can be incrementally transmitted and displayed. Second, the embedded bit stream can be truncated at any point by the decoder to created the ̈bestërror-bounded approximation of the model with exact bit rate control.

Topology Constraints. A topology constrained scheme does not change the genus or number of connected components of input surfaces. Strict constraints definitely limit the size of the coarsest mesh it can produces. For high genus objects or surfaces with many components this limitation can make the scheme impractical because substantial simplification can only be obtained by removing holes and/or merging components. On the other hand, topology preserving lossy compression can be essential for certain applications where small geometry error is tolerable, but topology features should be preserved. Ideally, a multiresolution representation should have the flexibility of allowing between topology preserving and non-preserving strategies.

Multilevel Mapping. In a multi-resolution representation scheme the correspondence between geometry and associated attributes (like color, textures and normals) is best maintained by constructing continuous maps between different levels of resolution.

In this paper we introduce a new compact multiresolution representation scheme which has the above criteria and provides the flexibility to trade visual quality and adaptability for storage complexity and topology. Priority can be chosen according to the requirements of a particular application. This scheme is able to express arbitrary triangular meshes. That is, it can be used to compactly encode and progressively retrieve any set of triangles without any constraint on the topological type, orientability or manifold characterization. The non-manifold mesh in Figure 1 has sixteen components, with the adjacent two sharing a non-manifold vertex. Input objects can also have attached properties such as colors, normals or texture coordinates. Our scheme is comparable in several aspects to prior published methods [14, 27, 30, 19] and improves on a few important criteria [1, 33].

The rest of this paper is as follows. Section 2 discusses several of the prior published work on compact multiresolution representations, including tradeoffs to our afore mentioned criteria. Section 3 details our multiresolution topological layering scheme. Section 4 details the flexibility on topology preserving/non-preserving simplification. Section 5 explains our geometric encoding of both mesh vertex coordinates as well as attached attribute data. Section 6 provides compression performance analysis of our scheme as well as several examples.

2 Prior Work on Multiresolution Representations

There have been several scheme for single resolution compression of triangle meshes with attribute data [1, 5, 31, 20, 32, 13, 4]. Here we concentrate on prior work on multiresolution representations.

Zorin, Schroder and Sweldens use subdivision to connect and unify patches and polygonal meshes in order to produce a tool to do mesh manipulation [34]. Their algorithms are designed for interactive multiresolution editing over meshes of arbitrary topology. However compression and progressive transmission are only suggested as possible future research objectives via the multiresolution transform in their paper.

Recursive subdivision schemes [26, 3, 6, 21] represent a polygonal mesh as a low resolution base mesh and an ordered sequence of details which are actual subdivision steps. Connectivity refinement and geometry smoothing are two operations in each of such steps. Connectivity refinement adds more vertices and faces to the mesh. Geometry smoothing adjusts vertex positions to obtain better approximation. Recursive subdivision and refinement schemes make it possible for the transition between two consecutive intermediate meshes to go smoothly. On the

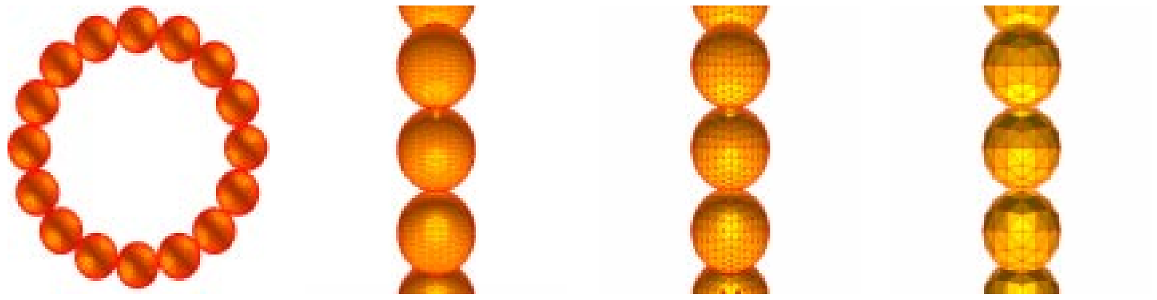


Figure 1: Multiresolution representation of a non manifold mesh model of a pearl necklace are showing the topological layering. The last three pictures show enlarged local details of three intermediate meshes created with our layering based simplification scheme. V = vertices, F = triangle faces.

other hand, these schemes only provide progressive representation for a narrow class of meshes which should have recursive subdivision connectivity.

Eck et al.[7] use a mesh with recursive subdivision connectivity to first approximate a mesh of arbitrary topology. Wavelets are then introduced to do multiresolution analysis of meshes with such connectivity property. Hierarchical quaternary subdivision trees are used to organize the vertices which will be coded in a top-down mode. Prediction coding is used for vertex positions by means of surface fitting. This wavelet based scheme does not support lossless compression because most meshes do not have subdivision connectivity. Also extension to meshes of high genus or arbitrary simplicial complexes could be problematic.

The adaptive-refinement Progressive Mesh (PM) scheme [14] introduced by Hoppe stores a manifold mesh as a low resolution coarse mesh together with an ordered sequence of details that can be used to refine the coarse mesh. There are two basic operations in this PM scheme: edge collapse and vertex split. Each vertex split operation adds a new vertex and two new triangles and locally refines the coarse mesh. $O(n\log(n))$ bits are needed to double the size of a mesh with n vertices. This degrades compression performance, especially for very large models.

Popovic and Hoppe[27] propose a lossy connectivity compression technique which outputs progressively transmittable bit stream. They express the original simplicial mesh at different resolutions through successive edge collapse and merging operations [15]. While not an efficient coding method, its fundamental contribution is its support for the progressive transmission. The compression method is improved by Li et al [20] through integrating the bit stream and the attribute stream under certain optimized criteria.

Progressive Forest Split (PFS) of Taubin et al [30] is a compact representation for any manifold mesh [30]. PFS decomposes a given mesh by a low resolution level-of-detail and a sequence of forest split operations. The forest split operation is specified by a forest in the graph of vertices and edges of the mesh, a sequence of simple polygons, and a sequence of vertex displacements. In order to express each forest split operation, forest edges, sequence of simple polygons, and vertex displacements are coded into compressed data streams.

Lee et al. [19] construct smooth parameterizations of irregular connectivity triangulation of genus 2-manifolds. A hierarchical simplification technique is used so that a parameterization of the original mesh over a coarse base mesh is obtained. To get an approximation with a specified error bound, the base mesh is hierarchically subdivided by the Loop method [21]. The remeshing procedure does not introduce much connectivity complexity. This multiresolution adaptive parameterization and remeshing technique can be used for progressive transmission of polygonal meshes in applications that do not require perfect connectivity recovery.

Kobbelt et al provide an interactive multiresolution modeling approach by building a hierarchy of nested spaces for unstructured data [17]. De Floriani et al. propose a general framework for multiresolution hierarchical representation [8] where the refinement step is the replacement of a portion of the mesh. Due to the high storage cost they experiment a variation of the data-structure that improves storage efficiency and allows progressive transmission [9].

In this paper, progressive connectivity transmission is based on the construction of multiresolution surfaces for arbitrary triangular meshes, including non-manifold and high genus. For an arbitrarily given mesh, its geometry and connectivity information is first organized by a layering structure. In this structure, vertices are further grouped into contours while triangles are merged into strips or fans. Through operations on these basic geometric primitives, multiresolution surfaces can be constructed with flexibility on topology preservation. To support flexible resolution, a topology non-preserving technique is also provided. Triangle contractions without any constraints are the fundamental operations. The coarsest mesh can be as small as the single vertex mesh. Our progressive scheme compares favorably with PM [14, 15, 27] for its visual efficacy and matches the compactness of PFS [30]. It holds a much larger input mesh domain than PFS as well as has the flexibility of topological preserving/non-preserving progressive meshes.

Our multiresolution surfaces will be constructed in two phases: the first phase preserves mesh topology and details in a compact way; the second phase provides flexible resolution degree and may change the mesh topology while the base mesh and details are still expressed in an economical way.

Topological Layering Our topological layering structure [1, 33] is inspired by the layering scheme that is used to construct vertex spanning trees for manifold meshes in [31].

The topological layering structure based on vertex neighborhood is used to encode the connectivity information of arbitrary triangular meshes as well as to index and establish local neighborhood and the second order of predict or corrector geometry encoding scheme. The input meshes are partitioned into two basic kinds of layers: **vertex layers** and **triangle layers**. The 0^{th} vertex layer is a randomly chosen vertex (could be a chain of vertices) of the mesh. The k^{th} vertex layer (with $k > 0$) includes a vertex V if V is not included in any previous vertex layer and there exists an edge $E = (V, V^*)$ where V^* is included in the $(k-1)^{th}$ vertex layer. The k^{th} triangle layer (with $k \geq 0$) includes a triangle T if T has one vertex in k^{th} vertex layer and T is not included in any previous triangle layer. This decomposition of the input triangular mesh provides a "Morse encoding" [22] capturing the branching and extreme points of edge cycles boundary holes and components. We use this effectively for generating both topological preserving and non-preserving simplifications. See Figure 2 color encoded vertex and triangle layers.

The topological layering structure has the property that any mesh edge and thus any triangle can only span two vertex layers. Therefore all edges are classified into two categories: transversals and chords. A **chord** is an edge that connects two vertices in different vertex layers while a **transversal** is an edge that connects two vertices in the same vertex layer.

Geometric Primitives There are four geometric primitives contours, branching points, triangle strips, and triangle fans (see Figure 3).

A **contour** is an ordered chain of vertices $\{v_0, v_1, \cdots, v_n\}$ in a vertex layer where each vertex pair (v_i, v_{i+1}) is connected by a transversal edge and every intermediate vertex v_i $(0 < i < n)$ is incident to exactly two transversal edges. In a closed contour v_0 is coincident with v_n and if n is 0 the contour degenerates to an isolated

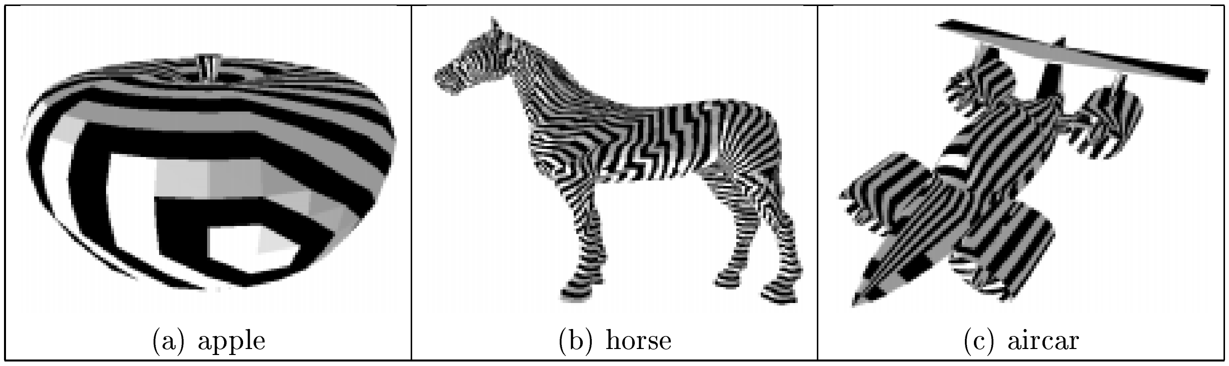


Figure 2: Topological layering of three different models with the last model having non-manifold features. Triangles in two adjacent layers are differently colored.

point. Contours are built in a greedy fashion by connecting vertices with transversal edges. A **branching point** is a vertex of a vertex layer which is incident to more than two transversal edges.

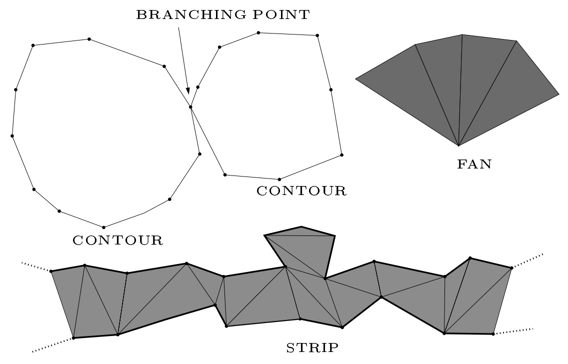


Figure 3: Contour, branching point, triangular strip and fan.

A **triangle strip** is an ordered sequence of triangles in a triangle layer where each pair of consecutive triangles share a common edge which is a chord. All the vertices of a strip belong to exactly two contours in two adjacent vertex-layers. Triangle strips are constructed using a greedy approach by merging triangles that share chords.

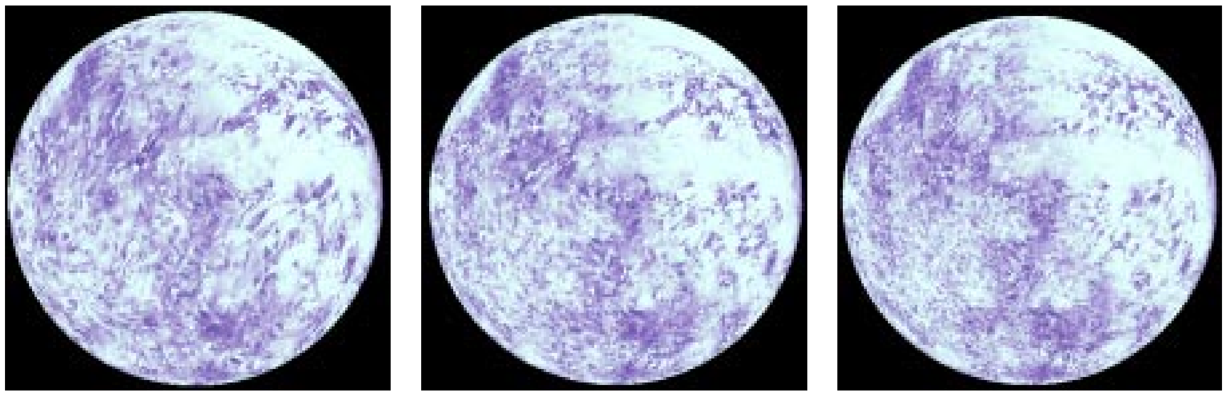
A **triangle fan** is an ordered sequence of triangles in a single triangle layer where all triangles have a common vertex, each pair of consecutive triangles share a common edge, and no edge is shared by more than two triangles. Once all the triangle strips are collected in a triangle layer, the remaining triangles are grouped into triangle fans again in a greedy approach to create longer fans.

Error Resilient Incremental Transmission The layering structure used in the single resolution compression and coding scheme naturally divides the compressed stream into small blocks because of its following locality properties: every triangle layer is dependent only on two adjacent vertex layers; every vertex layer is necessary to decode only two adjacent triangle layers. This implies that the geometric primitives for each vertex layer and triangle layer can be independently encoded. With the locality property, error resilience is also supported [1, 33] over unreliable communication channels.

We claim here a simple yet fundamental result that advocates the use of the layering structure as a general purpose representation for compact storage and transmission of triangle meshes.

Lemma 2.1 *(Complete Representation Power) Any triangle mesh can be represented by layering structure.*

Proof. The breadth-first mesh traversal guarantees to visit all the mesh edges and hence classifies vertices and triangles into layers. By representing each vertex with a separate contour and each triangle as a separate fan, one can represent a mesh consisting of any set of triangles.



28,684 faces, 120,922 bytes
45,954 faces, 189,330 bytes
82,743 faces, 326,799 bytes

Figure 4: Multiresolution of representation of an object with color attribute.

◇

The mesh to be expressed could be non-manifold or a mesh that does not form a simplicial complex or that needs to be repaired [18, 29]. Thus this layering scheme is general enough to represent any set of triangles. Its other advantage is the automatic avoidance of the so-called crack problem which occurs when a non-manifold mesh is converted into several manifold components by duplicating non-manifold features such as vertices, edges and faces [12].

3 Multiresolution Topological Contouring

The topological layering structure [1, 33] is first extended to a multiresolution representation.

3.1 Mesh Simplification and Progressive Reconstruction

The multiresolution layering representation is composed of a coarse mesh M_k along with a sequence of details $D_k, D_{k-1}, \ldots, D_1$. The mesh simplification procedure starts with the finest mesh M_0 and decomposes it into detail D_1 and a coarser mesh M_1. Iteratively, the procedure generates the details $D_1, D_2, \ldots, D_k$ the base level coarsest mesh M_k. We allow for flexible topological preserving and non-preserving multiresolution representation.

(i) Topological layering based simplification that simplifies the mesh while preserving the layering structure of the input mesh M with a very compact representation. In this stage the topology of the mesh is not modified.

(ii) Topological non-preserving simplification which is based on a generalized triangle contraction primitive [11]. The details generated in this stage are relatively expensive to express in the encoded bit stream. However it produces better quality simplification and guarantees the progress of the simplification. For this reason this second stage is applied only to generate the coarser levels especially when storage needs become progressively negligible

The block diagram in Figure 7 outlines our progressive coder which creates the multiresolution representation where C and G stand for connectivity and geometry data of mesh M_i and D for the detail. The entropy coder removes redundance in the digital representation of the details. We explain below how these details are digitally encoded.

3.2 Topological Layering Based Simplification

The layering based simplification uses two basic decimation operations: the intra-layer simplifi-

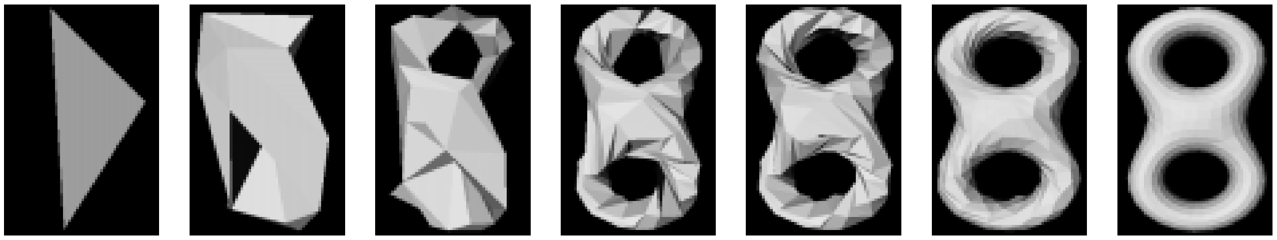


Figure 5: Progressive transmission of a triangular mesh. The topology of this model changes between the third and fourth pictures.

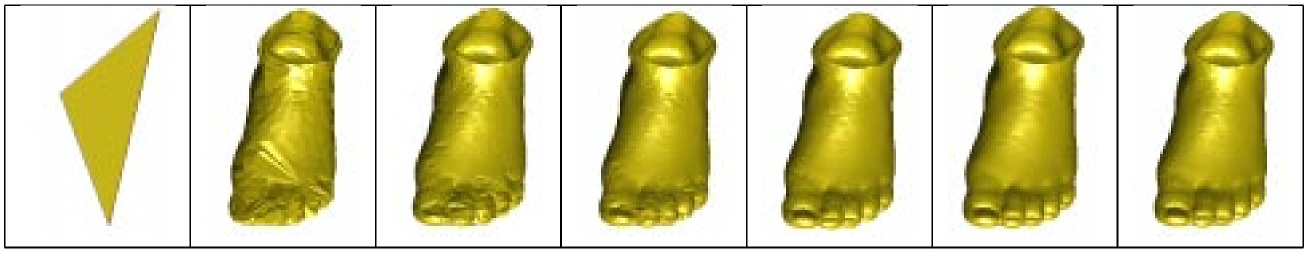


Figure 6: Progressive transmission of a triangular mesh. This sequence of non-consecutive intermediate models consist of 1, 1,285, 3,328, 8,366, 12,380, 16,392 and 40,432 triangles.

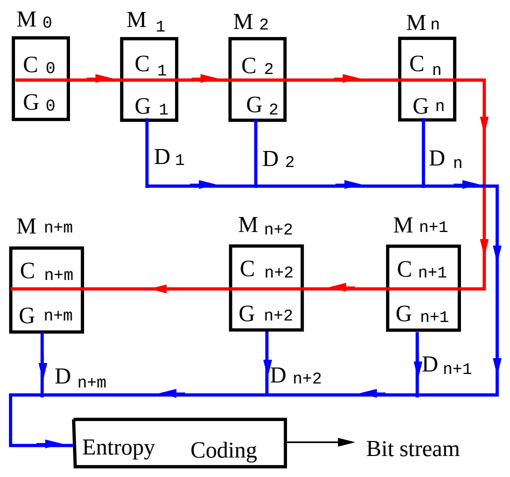


Figure 7: Block diagram of the progressive transmission coder.

cation that removes vertices within a contour without changing the topology; and the inter-layer simplification that removes an entire contour. Figure 8 shows how these two operations can alternatively used to reduce the visual artifacts induced by the layering structure.

3.2.1 Intra-layer Simplification

In the intra-layer simplification a local vertex removal is performed with retriangulation and it maintains the layering structure.

Figure 9 shows the transition of a simple strip when some vertices in its child and parent contours are removed. One obvious characteristic of this simplification is that the simplified strip still has the same parent and child vertex contours.

The selection of the vertices to be removed in one contour is based on three sufficient criteria to guarantee topology preservation and the compact encoding of the details. First, there are at least three vertices remaining in the contour of the candidate vertex. Second, the two extreme vertices of the contour cannot be decimated. Finally, two consecutive vertices in a contour cannot be decimated in the same mesh transition.

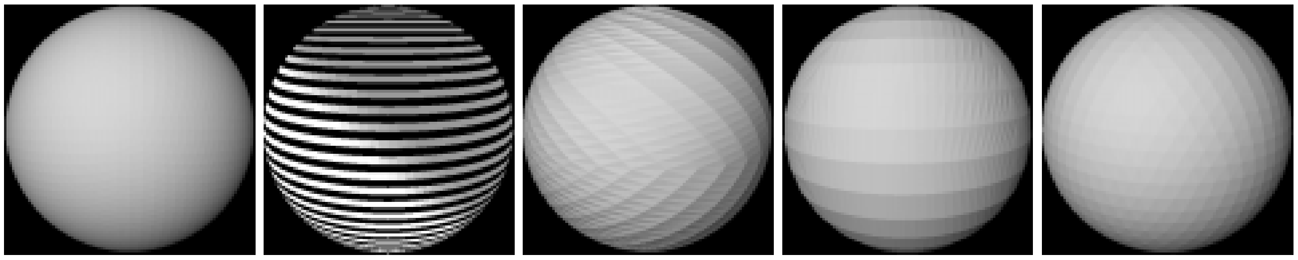


Figure 8: A sphere model, its layering structure and three combinations of intra-layer and inter-layer simplification.

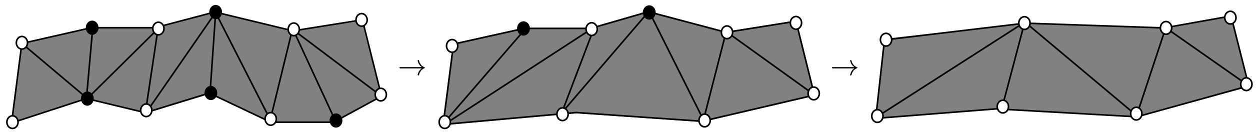


Figure 9: Intra-layer simplification: retriangulation of a simple triangle strip after the solid vertices are removed from its bottom and top contours.

Connectivity Details. Figure 10 shows the decimation procedure of a contour with the connectivity details. The solid vertices are decimated while the gray ones remain. One bit is associated with each remaining vertex: the value of this bit is 1 if its successor is decimated and is 0 otherwise.

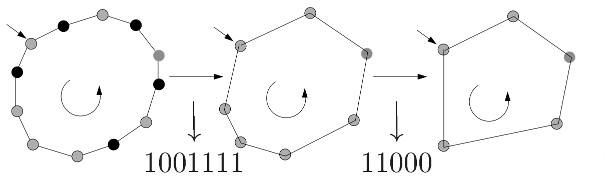


Figure 10: Connectivity details for two transition steps in the intra-layer simplification where the first vertex of the contour is marked with an arrow.

Connectivity reconstruction of a contour is straightforward. Starting from a bit string encoding the contour detail, one bit is associated with each edge in the contour. If the bit value of an edge is 1, it is split into two edges. Otherwise, the edge is not split.

The remaining information necessary for the encoding is related to the modification of triangle strips incident to the decimated contour. Figure 11 (left) shows a star polygon around the vertex v to be removed. Vertices vL, vR lie on the same contour as v. $v1, v2$ and $v3$ are chord-connected in the previous vertex layer. The remaining vertices are chord-connected to the next layer. To preserve the layering structure after the decimation a new transversal edge must be added to connect vL to vR. The first half, approximated for excess, of the vertices in the previous and next layer are connected to vL. The second half, approximated for defect, of the vertices in the previous and next layer are connected to vR. The middle vertices of the previous and next layer are connected to both vL and vR. With this fixed retriangulation rule, the details, sufficient to recover the original triangulation, are the numbers of vertices in the previous and next layer that are connected to v.

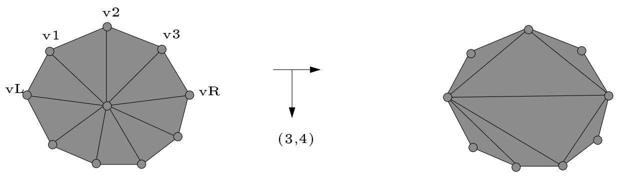


Figure 11: Constrained retriangulation and detail extraction for one transition of intra-layer simplification.

Simplification Order The order of the simplification procedure goes as follows. For a single component, it goes from the first vertex layer to the last one; for each vertex-layer, it goes from the first contour to the last one; for each contour, it does from the starting vertex to the ending vertex. If multiple simplification transitions are performed, the above procedure executes multiple times. The reconstruction procedure follows the exact reverse order.

3.2.2 Inter-layer Simplification

In the inter-layer simplification stage the decimation operation is the removal of a contour. Consequently, the gap left between the two adjacent contours of the removed one must be retriangulated. There are three conditions on the decimation of an entire contour.

- Its two adjacent contours in the previous and next vertex layers do not contain branching points;
- The error introduced by the contour decimation does not exceed a given tolerance.
- The gap left must be triangulatable.

Connectivity Detail The detail necessary to reconstruct the topology needs to include the configuration of the removed removed and the modification of the triangulation induced by the decimation.

The contour itself is simply characterized by the number of its vertices plus one bit set to 1 if it is an open contour or 0 if it is closed. For example, in Figure 12 an open contour of eight vertices is being decimated with the detail (0,8).

For retriangulation, the two strips sharing the removed contour are replaced by a single strip using [2]. This new strip is confined to have a chord, called constraining chord, that is corresponding to every pair of triangles having a common edge on the removed contour. The middle picture of Figure 12 shows this correspondence in dotted lines. Figure 12 also displays the final retriangulation. To reconstruct a fine triangulation from the coarse strip it is sufficient to know which edges are constraining chords. We express the details as an ordered sequence of small integers. Each integer stands for the number of triangles in the coarse strip that lie between two consecutive constraining chords. The starting edge of a coarse strip is always a constraining chord.

In the reconstruction procedure, the bit stream of the details, are used to locate all the constraining chords. Every chord is then turned into a pair of triangles. By connecting the sequence of edges shared by the pairs of triangles, a new contour (the removed one in the simplification procedure) is formed. Any vertex in the two contours of the coarse strip that is not chord-connected to the new contour is connected to the same vertex as its predecessor.

Simplification Order The order of the simplification procedure goes as follows. For a single component, it goes from the first vertex-layer to the last one; for each vertex-layer, it goes from the first contour to the last one. This procedure needs to be executed multiple times if there are multiple runs. The reconstruction process follows the exact reverse order of the encoding.

4 Topology Non-preserving Simplification

Since the intra-layer and inter-layer simplifications are constrained to be topology preserving, there is a limit on the level of simplification that can be achieved. To guarantee the progress of mesh simplification process it is sometimes necessary to change the topology of the input mesh [28, 10]. We introduce a topology non-preserving generalization of our scheme based on the triangle contraction primitive introduced in [11].

In our scheme (Figure 7), the input mesh to the topology non-preserving simplification is the coarsest mesh generated by the layering based simplification of section 3 after several topology iterations of intra-Layer and inter-Layer operations.

Generalized Triangle Contraction When a triangle t is contracted, all its vertices are merged into a single point p. The index used for p after

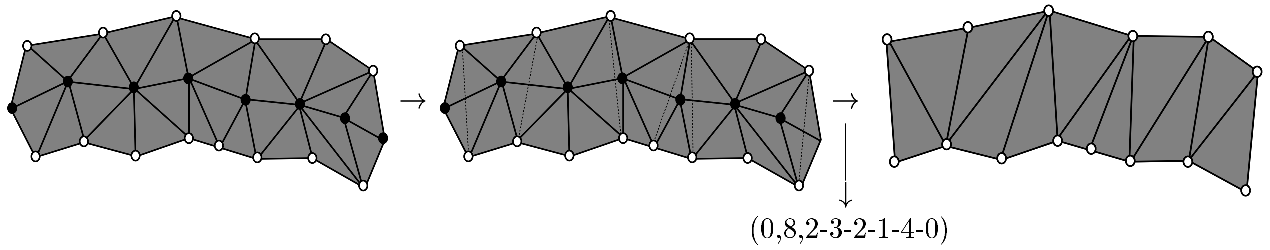


Figure 12: Inter-layer simplification. (left) Fine level being decimated. (center) Constraining chords drawn on the corresponding triangle pairs. (right) Coarse level and connectivity detail.

the contraction is the the smallest among the indices of the vertices of t.

As shown in [11] p is chosen to be the best local shape fit to achieve its high visual quality. The downside of this approach is high storage cost that arises from storing both p and the three vertices of t. For lower storage overhead we choose p to be a fixed linear combination of the vertices of t so that we only store as detail p and two vertices of t. The third vertex of t is recovered by inverting the linear constraint between p and t. In our implementation we choose p to be the barycenter of t.

Every triangle incident to t is classified into three categories:

- type I: one vertex in common with t
- type II: two vertices in common with t
- type III: three vertices in common with t

In most cases, there are no triangle of type III. But if there are such triangles, the details can still be efficiently coded: First, a small number is used to indicate the number of all type III triangles; Then for each such triangle, a bit is used to specify its orientation with respect to the contracted triangle.

Figure 13 shows the procedure of one triangle contraction and the corresponding triangle classification. The contracted triangle t is black-colored. After the contraction, the non-manifold edge of the original mesh is removed and the topology is changed.

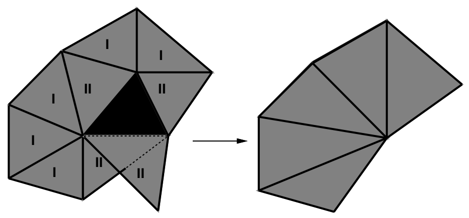


Figure 13: Topology non-preserving simplification by triangle contraction.

Connectivity Detail Before starting the triangle contraction procedure we encode the topological layering structure of the mesh which implicitly induces the vertex numbering to be used later. The index order of the vertices is key information that the decoder requires to correctly reconstruct subsequent layers.

For each decimation operation, one bit is spent to indicate if t is actually a triangle of the mesh or a triangular hole. Type I triangles remain after each contraction operation, but the indices of their coincident vertices may change. The detail of the triangle contraction needs to record this change. Let the vertex indices of t be v_0, v_1 and v_2, with the indices $v_0 < v_1, v_0 < v_2$. One bit is used to report if the index of the common vertex with type I triangle is v_0 or not. If this is the case no more information is needed. Otherwise, another bit is used to report if the common vertex is either v_1 or v_2. Type I triangles are sorted by the two index values of the non-common vertices and then, in this order, their details are stored.

Type II triangles degenerate into edges. The index of the vertex that is not common with t must be reported in detail. Moreover three bits are used to distinguish which edge they share with t. Isolated vertices may appear after a triangle contraction because of the degeneration of type II triangles. We record the indices and geometry (possibly, with other attributes) of such isolated vertices when they occur.

Decimation Priority Two parameters are used to prioritize the order of the triangle contractions. The first parameter is the area $A(t)$ of the contracted triangle t. Small size triangles are considered first since they affect the overall mesh shape the least. The second parameter is a measure used to avoid producing sliver triangles. We define the sliver factor $S(t)$ of a triangle t to be the ratio between the perimeter $P(t)$ of t and the maximum perimeter of its adjacent triangles:

$$S(t) = \max_{s \in Adj(t)} P(s)/P(t)$$

where $Adj(t)$ is the set of triangles adjacent to t. The topology non-preserving simplification procedure creates a priority queue with all the triangles of the mesh. The priority key associated with each triangle t is the product of its area by the sliver factor $A(t)S(t)$. The triangle with smallest key is selected for contraction. After the contraction, the removed triangle and its degenerate neighbors are removed from the queue. Other affected triangles have their key values updated and the priority queue is adjusted correspondingly. Unless a stopping criterion is set the procedure makes progress until the mesh become empty.

Figures 5 and 6 exhibit the reconstruction of simple “eight” and “foot” models. All the three simplification operations are used. As can be seen, the displayed coarsest mesh only consists of one triangle. It is obvious that triangle contraction changes topology.

5 Geometry Encoding

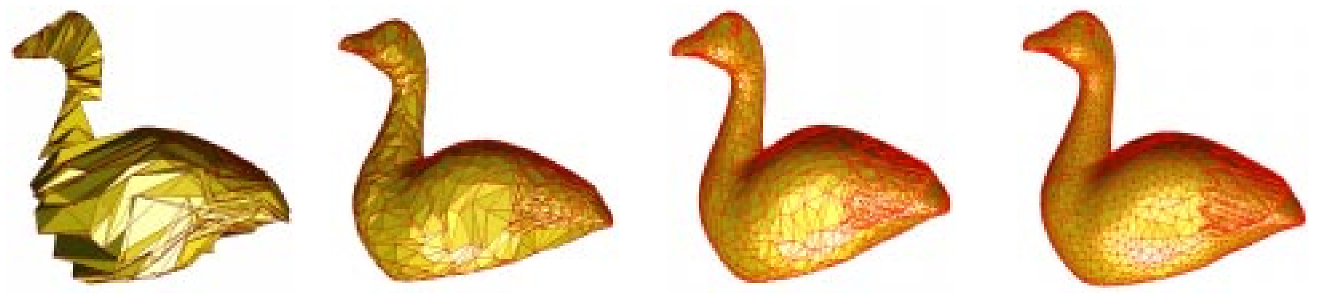
Geometry data may affect progressive transmission dramatically because it usually takes more space to encode. The main problem is its need to efficiently represent in the bit stream the positions of removed vertices. Linear prediction is often used to predict the position of a vertex from encoded positions of its adjacent vertices. In the following, we will describe how second order predictive and entropy coding is used in our multiresolution representation.

Intra-layer Decomposition The geometry detail of a decimated vertex is the difference between its predicted position and its actual position of a vertex. A simple predictive coding would take the center of the segment that the vertex splits. An alternative and potentially more accurate way is to build a quadratic Bézier curve that interpolates the coarse contour and to take the midpoints of the curvilinear sides instead of the straight lines of the original contour. Prediction in this way produces correction vectors with less variation and are thus more suitable for entropy coding.

Intra-layer Decomposition For a contour removed in the inter-layer decomposition procedure, vertex positions are encoded in the same way as the single-resolution geometry coding. Linear [5], high-order [31], and parallelogram predictors [20] can also be used. We use a second-order predictor to encode the geometry (similarly for attached properties). The basic idea is as follows. First, linear prediction is used to get a set of correction vectors which are parameterized and quantized using spherical coordinates (r, ϕ, θ). With the observation that the code difference of two successive correction vectors has small variation, the second-order prediction is then performed in the code space. The geometric error is bounded by the maximum quantization distortion. The coding scheme is also designed so that error propagation is prevented[1, 33]. In our implementation, classical entropy codings such as Huffman coding [16] and Arithmetic coding [23] are used to further improve space efficiency.

Figure 5 gives an example of layering based simplifications where inter-layer and intra-layer operations are alternatively performed.

Topology Non-preserving Decomposition For each decimation operation, the three vertices of the contracted triangle are merged into a sin-



150 faces, 931 bytes   1,228 faces, 7,368 bytes   5,856 faces, 16,689 bytes   10,009 faces, 24,021 bytes

Figure 14: Topological contouring simplification. Alternative inter-layer and intra-layer simplifications are used.

gle point. The remaining vertex of the triangle takes the center of the triangle as its geometry position. The positions of all removed vertices during this operation are encoded by predictions on this central position. In the reconstructing process, the remaining vertex, which is recovered at earlier stage, is pushed back to its true position when the other two vertices are recovered.

6 Performance Analysis and Results

Storage Analysis of Connectivity Details The vertex ordering is crucial for efficient storage of the details. Hoppe's progressive mesh (PM) needs $log(n)$ bits to specify the vertex from which a new vertex is split. Thus the overall connectivity cost of PM is $(nlog(n)+5)$. Our scheme squeezes off the $O(log(n))$ factor by taking advantage of the locality property of the layering structure. The space efficient representation is gained through the local indexing technique.

Intra-layer Decomposition With the constraints on the selection of vertices, at most one half of vertices in a contour can be removed in a intra-layer decomposition run. Practice shows that nearly one-half of vertices in a contour can be removed in each decomposition run. On average, $4.5\alpha m$ bits are used to expressed details if αm $(0 \leq \alpha \leq 0.5)$ vertices are removed with m being the number of vertices before decomposition.

Inter-layer Decomposition Suppose that three involved contours have n_0, n_1 and n_2 vertices, respectively. Then the two strips totally contain about $n_0 + 2n_1 + n_2$ triangles. Assume that number is $n_0 + 2n_1 + n_2$. The cost of coding the bit march string for the coarse level strip is about $n_0 + n_2$ bits since there exists that number of triangles in the coarse level strip. From the way the details are expressed, their coding bits are equal to the summation of the triangle number of the coarse level strip and the cost of coding the n_1 separators. That is, the detail coding needs $n_0 + n_1 + n_2$ bits.

Topology Non-preserving Decomposition According to the analysis above, the coding cost in bits is linearly proportional to the number of removed vertices with a small linear factor. No vertex indices are needed to be encoded and thus no $log(n)$ factor. For topology non-preserving decomposition, the logarithmic factor comes back but with a much smaller n because topology non-preserving decomposition are usually performed on a much coarser mesh obtained after several steps of the topological layering simplification.

Geomorph. For intra-layer simplification, the transition from one level to the next can be performed by continuous deformation using two vertex splits followed by one edge contraction. Figure 15 shows the reconstruction process where the vertices vL and vR are temporarily split into (vL, vL^*) and (vR^*, vR). Then the edge (vL^*, vR^*) contracts into the vertex v. For inter-

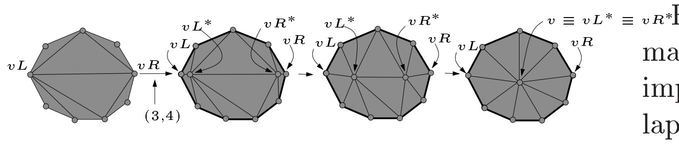


Figure 15: Geomorph for the reconstruction from an Intra-Layer detail.

layer simplification, the transformation from one level to the next can be performed by continuous deformation. For example, the geomorph from a coarse level to a finer one is performed by inserting a vertex at the midpoint of each chord edge in the coarse. The midpoints are connected by a sequence of dummy edges which split the strip triangles into coplanar triangles so that the surface geometry is not altered yet. Then, while the midpoint of the constraining chords are split to reconstruct the contour c the dummy edges are contracted. This continuously deforms the coarse strip into the two of the finer level.

Mapping For each intra-layer decomposition we construct an isomorphism between adjacent levels in the multiresolution hierarchy. In particular both the fine and coarse triangulation are mapped to the same rectangular region R as in Figure 16 using piecewise affine mapping from each triangle to its image on R. Two points at the two levels of resolution are in correspondence if they map to the same point in R.

Note that if $v1$ is coincident with $v2$ then the top part of R becomes a triangle. The same holds for the bottom part.

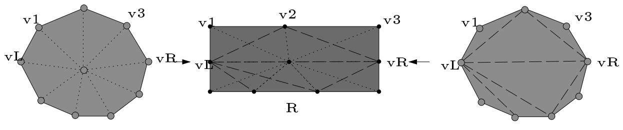


Figure 16: Mapping between adjacent levels of resolution for Intra-Layer Simplification.

Similarly for the case of intra layer simplification, we can map both the fine and the coarse triangulation to a common rectangle R. The two triangulations of R corresponding to fine and coarse levels induce a bijective mapping between points at different levels of resolution.

For topological non-preserving operation, the mapping between adjacent levels of resolution is implemented in a way similar to [19] by overlapping the local retriangulations generated by the triangle decimation primitive. This can be accomplished directly when the topology is not changed. Hence we stop building the multilevel map as soon as the topology non-preserving mode is selected.

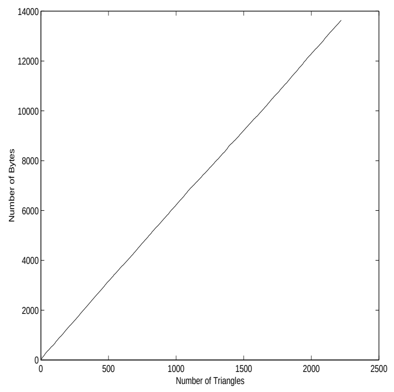
Results For geometry encoding, 12 bits are used to code each coordinate of correction vectors of vertices decimated in the topology non-preserving phase, while 10 bits for correction vectors of vertices decimated in the topology preserving phase. The two coding-bit numbers can flexibly be adjusted, dependent on the fidelity requirements of the application. Generally, more bits should be used for coding vertices that appear in coarser levels (thus recovered earlier) because better prediction can be achieved in this way.

Table 1 gives the average number of bits to recover each triangle at several levels of details. Figure 17 shows the statistics of progressive transmission performance. A curve is plotted for each decomposition phase which shows the number of bytes received as a function of the number of triangles reconstructed. Clearly, topological layering based decomposition provides a more compact representation than topology non-preserving decomposition.

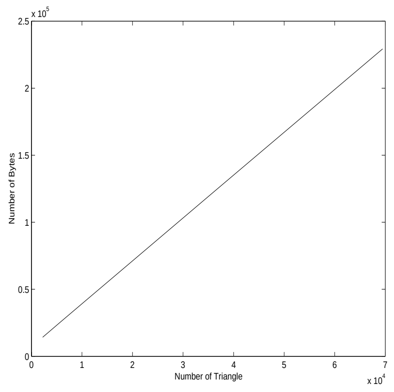
Figure 4 and 6 exhibit four levels of progressive compression for several triangular surface models. In Figure 6 the rightmost picture in each row shows the decoded object with perfect connectivity, and the leftmost one is at quite coarse level with less than 10% of the original number of triangles, while still being recognizable.

Duck							
T	150	414	1,228	2,341	4,012	5,856	10,009
$\Delta C/\Delta T$	18	14	4.6	4.4	4.2	4.2	4.1
$\Delta(G+C)/\Delta T$	44.1	37.8	33.6	25.4	21.3	22.7	19.3
C/T	19.4	18.1	15.7	11.3	8.5	6.4	5.2
(G+C)/T	42.2	37.4	36.2	30.5	24.6	23.1	21.7
Bunny							
T	1,000	9,315	17,869	18,955	34,885	36,557	69,473
$\Delta C/\Delta T$	18	3.7	4.5	4.2	4.6	4.5	4.5
$\Delta(G+C)/\Delta T$	38	24.7	18.0	25.2	18.1	24.5	18.0
C/T	21.3	7.9	6.3	6.2	5.4	5.3	5.0
(G+C)/T	48.7	23.0	20.6	20.9	19.6	19.8	19.0
Crocodile							
T	300	8,589	12,105	12,701	16,784	23,272	34,404
$\Delta C/\Delta T$	12	9.5	5.3	7.9	5.6	6.3	4.9
$\Delta(G+C)/\Delta T$	37.3	20.1	18.7	28.2	19.0	26.4	18.4
C/T	20.0	18.6	14.7	14.4	12.3	10.6	8.7
(G+C)/T	48.4	35.5	30.7	30.5	27.7	27.4	24.4
Phone							
T	1020	3,919	5,469	12,272	48,591	84,495	165,963
$\Delta C/\Delta T$	18	4.3	4.3	4.4	4.3	4.1	4.0
$\Delta(G+C)/\Delta T$	41.1	22.2	21.8	20.9	21.3	19.7	19.1
C/T	17.3	10.2	7.8	7.3	6.0	5.1	4.5
(G+C)/T	42.2	27.6	23.2	22.6	22.4	21.8	20.2
Horse							
T	870	3,122	5,496	5,922	9,754	13,188	22,258
$\Delta C/\Delta T$	18	4.2	4.7	4.9	4.8	3.8	4.7
$\Delta(G+C)/\Delta T$	42	25.2	18.3	25.9	18.3	24.8	18.2
C/T	18.3	11.6	8.6	8.3	6.9	6.1	5.5
(G+C)/T	42.7	28.5	24.1	24.2	21.9	22.6	20.8

Table 1: Coding statistics of multiresolution representation. T: number of triangles; C/T: connectivity cost in bits that is needed to recover each triangle; (G+C)/T: total cost in bits to recover each triangle. G: geometry encoding cost; C: connecting encoding cost.



(a)



(b)

Figure 17: Reconstruction of progressively transmitted model. (a) Bunny (topology non-preserving reconstruction): average 6.50 bytes per triangle; (b) Bunny (layering based reconstruction): average 3.20 bytes per triangle

7 Conclusion

We have presented a scheme for progressive encoding and transmission of triangular surfaces. A successive quantization scheme [1, 33] is used to support progressive bit transmission. We also present a technique to construct multiresolution surfaces consisting of a coarse mesh with additional levels of detail. The combination of intra-layer and inter-layer decomposition provides a compact encoding of both the coarse mesh and these details. Topology non-preserving decomposition increases the resolution degree without much sacrifice of space efficiency. The scheme also has the flexibility for topology preserving simplification as well as is capable of geomorphs and the successive mapping property. The ability of handling non-manifold meshes coupled to the object symmetries that can be captured in the layered decomposition are also currently been evaluated for a mirroring and stitching core experiment in MPEG 4 [25].

References

[1] Chandrajit Bajaj, Valerio Pascucci, and Guozhong Zhuang. Single resolution compression of arbitrary triangular meshes with properties. In *Data Compression Conference*, pages 247–256. IEEE, 1999.

[2] Chandrajit L. Bajaj, E. J. Coyle, and K. Lin. Arbitrary topology shape reconstruction from planar cross sections. *Graphical Models and Image Procession*, 58(6):524–543, 1996.

[3] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10:350–355, 1978.

[4] Jonathan Cohen, Amitabh Varshney, Dinesh Manocha, Greg Turk, Hans Weber, Pankaj Agarwal, Frederick P. Brooks, Jr., and William Wright. Simplification envelopes. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 119–128. ACM SIGGRAPH, Addison Wesley, August 1996.

[5] M. Deering. Geometric compression. *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 13–20, 1995.

[6] D. Doo and M. Sabin. Behavior of recursive subdivision surfaces near extraordinary points. *Computer Aided Design*, 10:356–360, 1978.

[7] Mattias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, and Michael Lounsbery. Multiresolution analysis of arbitrary

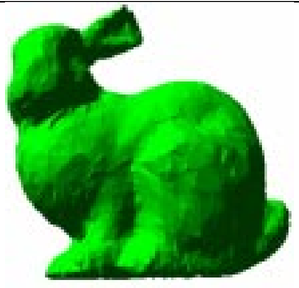
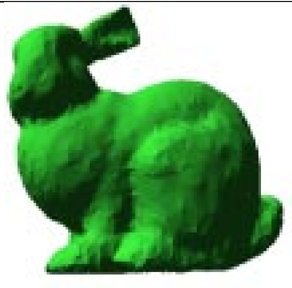
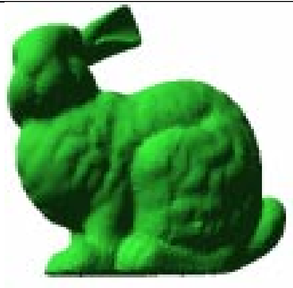
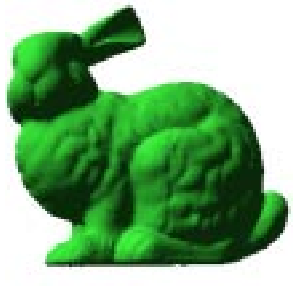
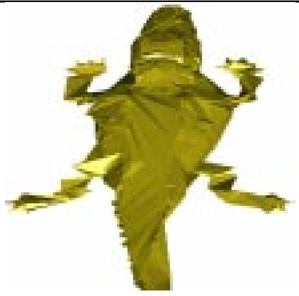
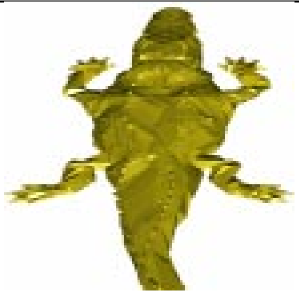
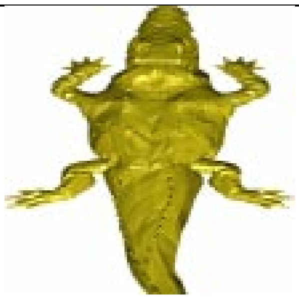
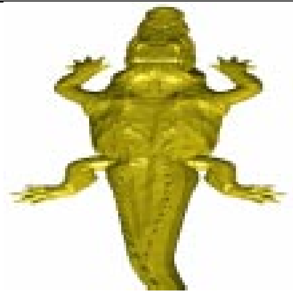
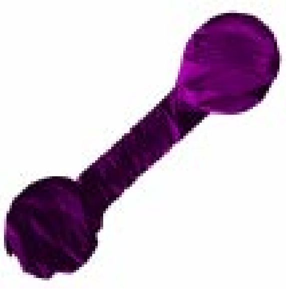
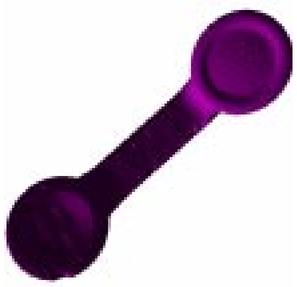
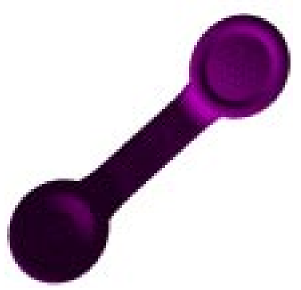
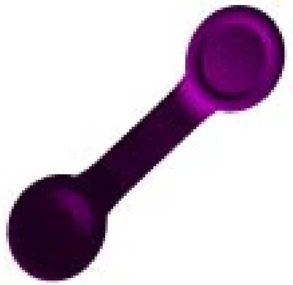
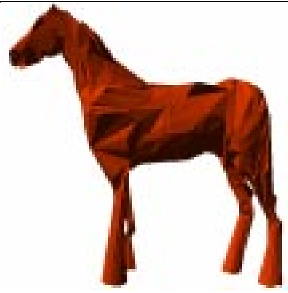
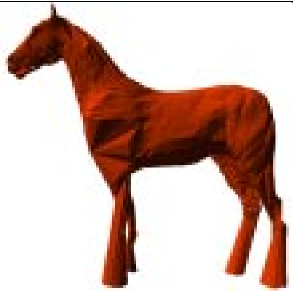
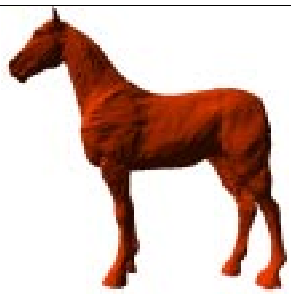
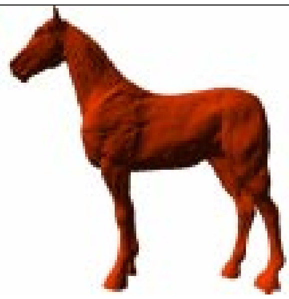
 4,293 △, 6.1% 16,384 bytes	 18,955 △, 27.2% 49,539 bytes	 36,553 △, 52.6% 90,737 bytes	 69,473 △, 100% 165,060 bytes
 1,260 △, 3.6% 7,417 bytes	 12,701 △, 36.9% 48,563 bytes	 23,272 △, 67.6% 79,759 bytes	 34,404 △, 100% 106,294 bytes
 3,919 △, 2.3% 17,589 bytes	 48,591s △, 29.2% 126,133 bytes	 84,495 △, 50.9% 232,476 bytes	 165,963 △, 100% 415,998 bytes
 1,516 △, 6.8% 6,485 bytes	 5,150 △, 23.1% 16,566 bytes	 13,188 △, 59.2% 37,400 bytes	 22,258 △, 100% 58,080 bytes

Figure 18: Compact multiresolution representation. The first numerical number at the bottom of each picture gives the number of triangles, the second gives the encoded byte size and the third is the percentage of object size with respect to the original one.

meshes. *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 173–182, 1995.

[8] L. D. Floriani and E. Puppo. Hierarchical triangulation for multiresolution surface description. *ACM Transactions on Graphics*, 14(4):363–411, 1995.

[9] Leila De Floriani, Paola Magillo, and Enrico Puppo. Efficient implementation of multi-triangulations. In *Proceedings IEEE Visualization'98*, pages 43–50. IEEE, 1998.

[10] M. Garland and P. Heckbert. Surface simplification using quadratic error metrics. *Computer Graphics (SIGGRAPH '97 Proceedings)*, pages 209–216, 1997.

[11] Tran S. Gieng, Bernd Hamann, Kenneth I. Joy, Gregory L. Schussman, and Issac J. Trotts. Constructing hierarchies for triangle meshes. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):145–161, April 1998.

[12] Andre Gueziec, Gabriel Taubin, Francis Lazarus, and William Horn. Cutting and stitching: Efficient conversion of a non-manifold polygonal surface to a manifold. Technical Report RC-20935, IBM T.J. Watson Research Center, P.O.Box 704, Yorktown Heights, NY 10598, 1997.

[13] S Gumhold and W. Strasser. Real time compression of triangle mesh connectivity. *Computer Graphics (SIGGRAPH '98 Proceedings)*, pages 133–140, 1998.

[14] Hugues Hoppe. Progressive meshes. *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 99–108, 1996.

[15] Hugues Hoppe. View-dependent refinement of progressive meshes. *Computer Graphics (SIGGRAPH '97 Proceedings)*, pages 189–198, 1997.

[16] David Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(10):1098–1101, 1952.

[17] L. Kobbelt, S Campagna, J. Vorsatz, and H. Seidel. Interactive multi-resolution modeling on arbitrary meshes. *Computer Graphics (SIGGRAPH '98 Proceedings)*, pages 105–114, 1998.

[18] F.-L. Krause, C. Stiel, and J. Luddemann. Processing of CAD-Data: Conversion, verification and repair. In Christoph Hoffmann and Wim Bronsvort, editors, *Proceedings of the 4th Symposium on Solid Modeling and Applications*, pages 248–254, New York, May14–16 1997. ACM Press.

[19] A. Lee, W. Sweldens, P. Schroder, L. Cowsar, and D. Dobkin. Maps: Multiresolution adaptive parameterization of surfaces. *Computer Graphics (SIGGRAPH '98 Proceedings)*, pages 95–104, 1998.

[20] Jiankun Li, Jin Li, and C.-C. Jay Kuo. Progressive compression of 3D graphic models. In *IEEE Proceedings of Multimedia Computing and Systems*, Ottawa, Canada, 1997.

[21] C. Loop. Smooth subdivision surfaces based on triangles. Technical report, Master's thesis, Department of Mathematics, University of Utah, August, 1987, 1987.

[22] John Milnor. *Morse Theory*, volume 51 of *Annals of Mathematics Studies.* Princeton University Press, Princetom, 1963.

[23] A. Moffat, R. Neal, and I. Witten. Arithmetic coding revisited. In *IEEE Data Compression Conference, Snowbird, Utah*, 1995.

[24] MPEG4/SNHC. Description of core experiments on 3d model coding. Technical report, ISO/IEC JTC1/SC29/WG11 MPEG98/M3373, October, 1998.

[25] MPEG4/SNHC. Support for non-manifold meshes and symmetries. Technical report, ISO/IEC JTC1/SC29/WG11 MPEG98/M4061, October, 1998.

[26] Jorg. Peters and U. Reif. The simplest subdivision scheme for smoothing polyhedra. *ACM Transactions on Graphics*, 16(4):420–431, 1997.

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

[28] William J. Schroeder. A topology modifying progressive decimation algorithm. In Roni Yagel and Hans Hagen, editors, *IEEE Visualization 97́*, pages 205–212. IEEE, November 1997.

[29] A. Sheffer, T. Blacker, and M. Bercovier. Cad data repair based on virtual topology. In *Proc. 24th Design Automation Conference*, ASME Design Engineering Technical Conferences and Computers in Engineering Conference, pages 13–16, September 1998.

[30] Gabriel Taubin, Andre Geuziec, William Horn, and Fancis Lazarus. Progressive forest split compression. *Computer Graphics (SIGGRAPH '98 Proceedings)*, pages 123–132, 1998.

[31] Gabriel Taubin and Jarek Rossignac. Geometric compression through topological surgery. *ACM Transactions on Graphics*, 17(2):84–115, 1996.

[32] Costa Touma and Craig Gotsman. Triangle mesh compression. In Wayne Davis, Kellogg Booth, and Alain Fourier, editors, *Proceedings of the 24th Conference on Graphics Interface (GI-98)*, pages 26–34, San Francisco, June18–20 1998. Morgan Kaufmann Publishers.

[33] Guozhong Zhuang. *Compression and Progressive Transmission of Three-Dimensional Models.* PhD thesis, Department of Computer Sciences, Purdue University, 1998.

[34] D. Zorin, P. Schroder, and W. Sweldens. Interactive multiresolution mesh editing. *Computer Graphics (SIGGRAPH '97 Proceedings)*, pages 259–268, 1997.