Preprint
UCRL-JC-146852-REV-2

Multiresolution Distance Volumes for Progressive Surface Compression

D.E. Laney, M. Bertram, M.A. Duchaineau, N.L. Max

This article was submitted to
1st International Symposium on 3D Processing, Visualization and Transmission, Padova, Italy, June 19-21, 2002

April 18, 2002

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

Approved for public release; further dissemination unlimited

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

This report has been reproduced directly from the best available copy.

Available electronically at http://www.doc.gov/bridge

Available for a processing fee to U.S. Department of Energy
And its contractors in paper from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-mail: reports@adonis.osti.gov

Available for the sale to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/ordering.htm

OR

Lawrence Livermore National Laboratory
Technical Information Department's Digital Library
http://www.llnl.gov/tid/Library.html

Preprint

Multiresolution Distance Volumes for Progressive Surface Compression

Daniel E. Laney
Martin Bertram
Mark A. Duchaineau
Nelson L. Max

Submitted to the symposium on
3D Processing, Visualization, and Transmission

April 2002

University of California
Lawrence Livermore National Laboratory

Multiresolution Distance Volumes for Progressive Surface Compression

Daniel Laney
U.C. Davis
Dept. of Applied Science
dlaney@llnl.gov

Martin Bertram
University of Kaiserslautern
FB Informatik
P.O Box 3049
D-6753 Kaiserslautern

Mark Duchaineau
Lawrence Livermore
National Laboratory
7000 East Ave.
Livermore, CA 94550
duchaine@llnl.gov

Nelson Max
U.C. Davis
Dept. of Applied Science
max2@llnl.gov

Abstract

We present a surface compression method that stores surfaces as wavelet-compressed signed-distance volumes. Our approach enables the representation of surfaces with complex topology and arbitrary numbers of components within a single multiresolution data structure. This data structure elegantly handles topological modification at high compression rates. Our method does not require the costly and sometimes infeasible base mesh construction step required by subdivision surface approaches. We present several improvements over previous attempts at compressing signed-distance functions, including an $O(n)$ distance transform, a zero set initialization method for triangle meshes, and a specialized thresholding algorithm. We demonstrate the potential of sampled distance volumes for surface compression and progressive reconstruction for complex high genus surfaces.

1. Introduction

The rapid increase in computing power and advancements in surface acquisition techniques have enabled the creation of meshes of 400 million triangles and larger [18, 15]. This has led to a dilemma in surface visualization: meshes of this size and complexity require both efficient compression techniques and a capacity for level-of-detail interrogation. Progressive compression algorithms enable both efficient compression and level of detail reconstruction. A progressive compression algorithm re-orders the bit stream in such a way that the most relevant information is near the front of the stream. Thus, with a small number of bits a usable approximation of a surface can be obtained for interaction and browsing. This paper presents a system for progressively compressing surfaces via a signed-distance representation. Our approach efficiently represents complex surfaces with arbitrary numbers of components, removes the need to explicitly store the topology of the surface, and can be extended to time-varying surfaces.

Recently, subdivision surfaces have been shown to be effective for surface compression as the connectivity information only needs to be stored for the base mesh. The work of Khodakovsky [14] and Bertram [2] show that wavelet-based techniques on subdivision surfaces result in competitive compression rates and allow for progressive decompression. However, subdivision techniques require a coarse base mesh. Base mesh construction for large and complex surfaces with many components is difficult and often infeasible. Even if a base mesh is produced, a surface with hundreds or thousands of components requires topological modification in order to achieve usable progressive reconstructions.

In this paper we advocate an alternative approach to surface compression which is based on a signed-distance volume representation [26, 6]. A signed-distance volume is a trivariate function encoding the minimum distance to a surface for each volume sample. The sign changes as the surface is crossed. Figure 1 depicts the data flow in our system. The resulting compressed surface is reconstructed by extracting the isosurface with zero distance.

The signed-distance representation does not directly specify the topology of the surface. This freedom from storing the topology increases the potential for using simple algorithms that will extend elegantly to high genus surfaces

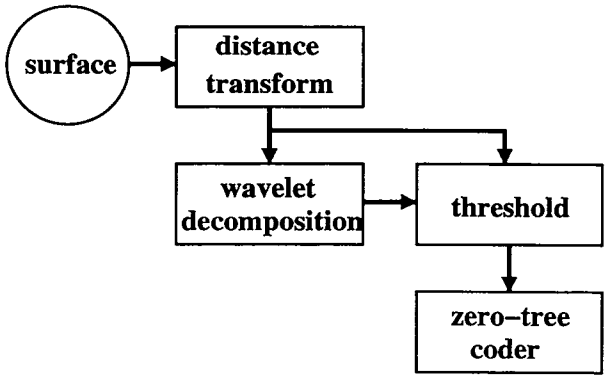


Figure 1. The compression system comprises four modules. The input surface is transformed into a distance representation then decomposed into linear B-spline wavelet coefficients. A thresholding algorithm is applied that sets a large number of wavelet coefficients to zero. The distance information is used to insure that the surface geometry is retained. Finally, a zero-tree coder produces the progressive bit stream.

and time-varying surfaces. We achieve a multiresolution representation by applying a linear B-spline wavelet decomposition to the implicit function. Compressing multiresolution signed-distance functions has been studied in [8, 7]. Our contribution is a complete system that overcomes the problems of existing distance-based compression methods. The main features of our method are as follows:

1. **Progressiveness:** we generate a progressive encoding of the distance function which can be partially reconstructed from the most relevant bits of the most relevant wavelet coefficients. The implicit nature of our representation facilitates topology modification to reduce the complexity of the approximate surfaces beyond what is attainable by subdivision surface methods.

2. **Scalability:** the method is not limited by the need to re-map a complex surface to a base mesh with subdivision connectivity. Our thresholding method removes wavelet coefficients that do not contribute to the zero set resulting in a size related to the surface complexity.

3. **Simplicity:** the distance volume representation dispenses with a lot of the algorithmic complexity associated with base mesh construction and explicit topology tracking. All operations in our method are performed on regularly sampled volumes.

4. **Autonomy:** the algorithm requires only a desired bit count in order to produce a compressed file. This is in contrast to subdivision methods which may require explicit base mesh vertex positioning for sharp features [14], editing operations [11], or multiple fitting parameters for obtaining the base mesh [1].

2. Related Work

Wavelet transforms have been used to obtain multiresolution representations of scalar volume data for rendering and compression [19, 20, 21, 9, 10]. Tao [23] describes a system for progressively transmitting volume data encoded as wavelet coefficients. Volume compression techniques [18, 13] based on wavelet transforms have been used to facilitate the visualization of large data sets. The present work uses standard wavelet transforms on volumetric data but is not concerned with representing the entire volume. We retain only the minimal number of wavelet coefficients necessary to represent a surface.

Multiresolution techniques have also been investigated in the implicit surface literature. Velho et. al. [25] proposed a multi-scale implicit representation based on a biorthogonal B-spline wavelet transform. Their technique produces a representation based only on B-spline scaling functions. They eliminate the wavelet coefficients by projecting the wavelets onto the scaling basis functions at the next finer scale. This eliminates the wavelet coefficients at the cost of an increased number of scaling coefficients. The trade-off is that all modeling and rendering operations are performed on a hierarchical B-spline representation. The work does not explicitly treat the problem of compressing the resulting data.

Closely related to the present work is the technique of Grisoni [8, 7]. They represent the field function of an implicit surface as a sampled volume and apply a wavelet transform to obtain a multiresolution representation. Following Velho's method, they project the wavelets onto the scaling basis at the next finer level producing a data structure with only scaling coefficients. Their thresholding scheme operates on the projected coefficients. The location of the wavelet coefficients is not considered in the thresholding process. In some cases coefficients affecting the reconstructed surface may be thresholded. They propose a sparse representation based on a hash table storing the location and value of each coefficient in a packed three byte block. Coefficients at coarser scales require fewer bits for encoding position and thus increase the number bits available for quantizing the coefficient value. The present work provides both a location-based thresholding scheme and a progressive bit ordering that reduces geometric error and improves compression.

3. Signed-Distance Volumes

A signed-distance volume encodes the minimum distance to a surface for each sample point. The distance changes sign at the surface so that negative values lie on one side and positive values on the other. Given a closed shape, the sign determines whether a point is inside or outside of the shape. For isosurfaces, the notion of inside and outside is not always applicable as the surface may exit the distance volume. In these cases the sign of the distance is determined by the scalar function without relying on notions of inside/outside. We formally define the signed distance from a surface Y as:

$$\text{dist}(x) := \text{sign}(x) * \min_{\forall y \in Y}(\| x - y \|) \qquad (1)$$

where $\text{sign}(x)$ is negative on one side of the surface and positive on the other. Most scanned objects are single closed components. An inside/outside relation can be defined for closed meshes if triangle normal vectors are oriented consistently. Isosurfaces from trilinearly interpolated scientific data also have this property although an isosurface may have a boundary on the boundary of the sampled volume. In such cases the boundary of the distance volume must coincide with the boundary of the scientific data. In the remainder of the paper we will use $\text{d}(x)$ to denote the approximate distance as computed by a distance transform algorithm.

3.1. Error Metrics

Surface errors are required to study the rate distortion properties of our algorithm. We adopt the L^2 error metric used in [14] and measured by the METRO tool [4]. The error is defined by taking the maximum of $\text{D}(X,Y)$ and $\text{D}(Y,X)$, where $\text{D}(X,Y)$ is the distance between two surfaces X and Y defined as:

$$\text{D}(X,Y) = \left(\frac{1}{\text{area}(X)} \int_{x \in X} \text{D}(x,Y)^2 dx \right)^{1/2} \qquad (2)$$

and $\text{D}(x,Y)$ is the Euclidean distance from a point $x \in X$ to the closest point on Y. All errors reported in this paper are relative to the bounding box diagonal length.

4. The Distance Transform

We apply a distance-transform algorithm to surfaces defined by triangle meshes and to isosurfaces from regularly sampled volumetric data. The transform produces an approximation of the actual distance function based on the closest-point propagation algorithm of Breen [3]. We have modified Breen's algorithm so that it runs in $\text{O}(n)$ time. The distance volume is initialized with closest-point information for all cells intersecting the surface to be encoded (the *zero set*). These are the only explicit computations with respect to the input surface. Once the zero set is initialized the propagation algorithm assigns the closest points to the rest of the volume samples. The zero set signs are also initialized and this information is propagated along with the closest points.

$\text{d}(s)$	Current approximate unsigned distance
$\text{sn}(s)$	Sign of distance function
$\text{amb}(s)$	True if sign is ambiguous ($\text{sn}(s)$ undefined)
$\text{cp}(s)$	The closest point of s ($\text{d}(s) = \|\|s - \text{cp}(s)\|\|$)

Table 1. Properties of distance samples

The propagation technique is essentially a point sampling approach, as the approximation is produced with respect to the initial set of closest points in the zero set. We begin by describing the propagation algorithm, then describe the zero set initialization methods for scanned and scientific surfaces.

4.1. Closest Point Propagation

The distance transform operates on a volume of regularly spaced samples. Let s denote a sample of a distance volume. Table 1 lists the properties of a distance sample s. The output of the algorithm is a regular volume of signed distance values.

The closest point propagation algorithm relies on the following heuristic: the closest point of a sample s will in most cases be geometrically close to the closest points of the neighbors of s. The propagation algorithm is as follows:

```
For all distance samples s:  d(s) = max_float
Initialize the zero set of the distance
  field as described in the following
  sections.
Place all zero set samples in a FIFO
  queue Q
while Q is not empty do
    Let s ← front(Q)
    For each 26-neighbor t of s do
        If ||cp(s) − t|| < d(t) then
            cp(t) ← cp(s)
            d(t) ← ||t − cp(t)||
            Place t onto back of Q
        If amb(s) = false then sn(t) ← sn(s)
```

Breen et. al. presents a method based on a priority-queue that always examines the sample with the smallest distance, insuring that each sample is visited only once by the algorithm. This leads to an expected running time of $\text{O}(n \log n)$ where n is number of samples in the distance volume. The algorithm presented here computes the same approximate distance volume, but may set the distance value of a sample

multiple times. In our experiments, the number of updates per distance sample tends towards one as the resolution of the distance volume increases. For the data shown in this paper each distance sample is updated only 1.1 times on average. Thus, the algorithm runs in time $O(n)$ on these surfaces because a simple queue provides constant time access.

4.2. Zero Set Initialization of Isosurfaces

Our implementation can produce signed-distance volumes of isosurfaces defined on regularly sampled scalar fields. Instead of formally defining the isosurface with respect to trilinear interpolation, we compute closest points based on local gradient estimates. The distance approximation is constructed at the same resolution as the initial scalar field. The algorithm examines each volume cell in the scalar field. If the cell contains the isosurface, then the distance samples at the cell corners are initialized with closest point information. Once a distance sample has been initialized it is not reinitialized later for another incident cell.

We denote the scalar field by $f(s)$. Let f_0 denote the isovalue of the desired isosurface. We define a linear approximation about a sample s as $\hat{f}(s') = f(s) + \nabla f(s)(s' - s)$ and compute the closest point:

$$cp(s) = s + \frac{f_0 - f(s)}{\|\nabla f(s)\|} \nabla f(s) \qquad (3)$$

The scalar field gradient at a given sample point s is estimated by central differencing. The sign of the distance is positive if $f(s) > f_0$ and negative otherwise. This approximation is inaccurate for high curvature regions but can be computed very efficiently. Greater accuracy can be obtained by performing Newton iterations, or by extracting a mesh and applying the technique in the next section.

4.3. Zero Set Initialization of Meshes

The zero set initialization for triangle meshes operates on individual triangles. The algorithm does not use edge or vertex adjacency information. In the pseudocode below, cell_width is defined as the distance between distance samples in the x, y, and z directions. The zero-set initialization algorithm proceeds as follows:

```
For every triangle T in the input mesh:
    Compute the triangle bounding box
      defined by (x_min, y_min, z_min) and
      (x_max, y_max, z_max)
    Reduce (x_min, y_min, z_min) by cell_width
    Increase (x_max, y_max, z_max) by cell_width
    For each sample s in the adjusted
      bounding box:
        Compute cp(s) on T
        Record whether cp(s) lies on a
          face, edge, or vertex.
        Let δ = ||cp(s) − s||
        If δ ≤ √3 * cell_width and δ < d(s)
          then:
            place the sample on the queue
              of zero set samples.
            Set a flag indicating that
              the sample is queued to
              prohibit duplication.
            Set sn(s) based on whether cp(s)
              lies on a face, edge, or
              vertex as explained below.
```

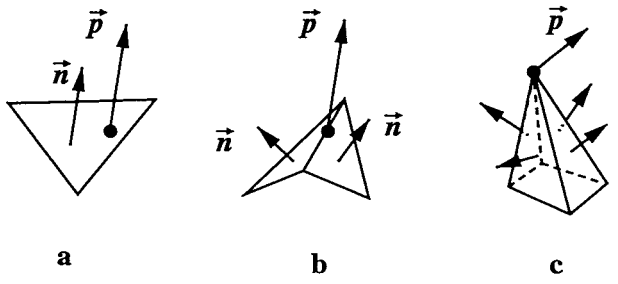


Figure 2. Zero set initialization for triangle meshes: the sign of a distance sample is determined using the vector $\vec{p}$ from the closest point to the sample and the triangle normal vectors $\vec{n}$.

The sign of a given sample is computed based on the location of its closest point (vertex, edge, or face). Figure 2a shows the simplest case where the closest point lies on a face. In this case the sign is given by the sign of $\vec{n} \cdot \vec{p}$ where $\vec{n}$ is the triangle normal and $\vec{p}$ is the vector from the closest point to the distance sample. If the closest point lies on an edge as in Figure 2b, then there are are two dot products (one for each triangle sharing the edge). The absolute values of the dot products are compared and the sign of the larger dot product is taken. Finally, a closest point which coincides with a vertex of the mesh as in Figure 2c may be ambiguous if some dot products are negative and others are positive. In these cases the distance sample is marked as *ambiguous* and no sign information is propagated for it. After the distance transform has completed the ambiguous samples are revisited and the following heuristic is applied: the signs of the 26-neighbors are examined and the sign of the majority of the neighbors is assigned to the sample. Our method is very similar to the technique in Huang et. al. [12]. However, their algorithm does not detect the ambiguous vertex clos-

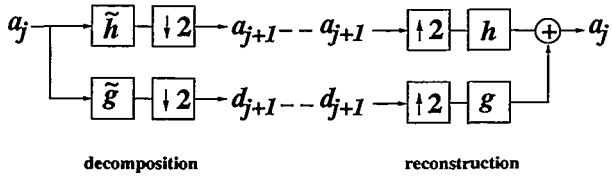


Figure 3. (Left) Decomposition filter bank with low pass filter $\tilde{h}$ and high pass filter $\tilde{g}$. (Right) Reconstruction filter bank with low pass filter h and high pass filter g.

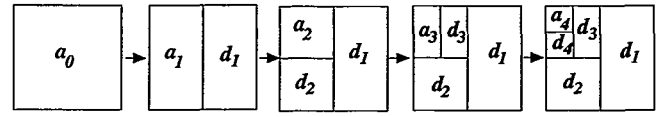


Figure 4. Two dimensional extension of the filter bank by alternating the directions of the filtering steps.

est points and may initialize the incorrect sign.

5. Wavelet Transforms

A wavelet transform [17] decomposes a signal into a sequence of wavelet coefficients representing the details of the signal at several levels of resolution. These coefficients are often of small value and can be compressed efficiently.

5.1. Fast Wavelet Transform

We use the fast wavelet transform of Mallat [16] to generate multiresolution representations. The left half of Fig. 3 shows one step of decomposition algorithm. At each step a low pass filter $\tilde{h}$ produces a set of scaling coefficients (a_{j+1}) which coarsely approximate the input data. Additionally, a high pass filter $\tilde{g}$ produces a set of wavelet coefficients (d_{j+1}) representing the details lost in the coarse approximation. These two filtering steps are repeated recursively on the coarse approximations to obtain a multiresolution representation. At each stage the size of the data is down sampled by a factor of two. We will use the term *subband* to refer to a set of wavelet coefficients generated by one step of the transform.

The original data can be reconstructed by reversing the process with another set set of filters h and g. The right half of Fig. 3 shows one reconstruction step. Both filters are preceded by an up sampling by two which inserts zeros between each pair of input values. The low pass filter h, when combined with up-sampling, is similar to a subdivision operator, smoothing coarse approximations. The high pass filter g re-introduces the details encoded by $\tilde{g}$ and enables exact reconstruction of the data. In the present work we use the linear B-spline wavelets with $\tilde{h}[n] = 1/8\,(-1,\ 2,\ 6,\ 2,\ -1)$, $h[n] = 1/2\,(1,\ 2,\ 1)$, $\tilde{g}[n] = 1/2\,(1,\ -2,\ 1)$, and $g[n] = 1/8\,(1,\ 2,\ -6,\ 2,\ 1)$.

The 2D extension of the algorithm in Fig. 3 is shown in Fig. 4. The one dimensional transform is alternately applied to each dimension, creating subbands 1 and 3 in the x direction (d_1, d_3) and subbands 2 and 4 in the y direction (d_2, d_4). The 3D case follows the same pattern as Fig. 4 except that the transform directions cycle through the x, y, and z directions. Some readers may note that in image processing applications the high pass coefficients resulting from the x direction filtering are processed by the filter bank a second time in the y direction yielding three sub-bands per level in the 2D case and seven subbands in 3D (after a z pass). In contrast, our approach generates one subband per level for data of any dimension. Our method requires fewer computations, and in our tests produces better compression.

5.2. Thresholding

The goal of the thresholding step is to reduce the number of values that need to be coded by setting insignificant wavelet coefficients to zero. An aggressive thresholding method is required for efficient distance volume compression. Our distance-based thresholding method removes all wavelet coefficients that do not contribute to the reconstructed surface. Thresholding too many coefficients could result in spurious surface components appearing in the distance field. Currently, we do not have a formalism that allows us to prove that new components or handles are not added under the method we present. For complicated surfaces a verification step can be performed that checks the original distance volume against the distance volume reconstructed after the thresholding step and warns of any irregularities. The thresholding method presented here did not modify the topology of surfaces we have tested.

Figure 5 illustrates the thresholding operation in 2D. On the left is the wavelet transformed signed-distance field showing five wavelet subbands. On the right we have the original signed-distance field as computed for the two curves shown inside. The support of a wavelet coefficient in d_5 is shown as the shaded rectangle. The wavelet coefficient in d_5 can be thresholded because its support does not effect the curves being represented.

Two methods are used to determine whether a given wavelet coefficient should be set to zero. First, a bounding sphere is computed that contains the wavelet support. The distance value at the center of the sphere is sampled, and if this distance is greater than the radius, the coefficient

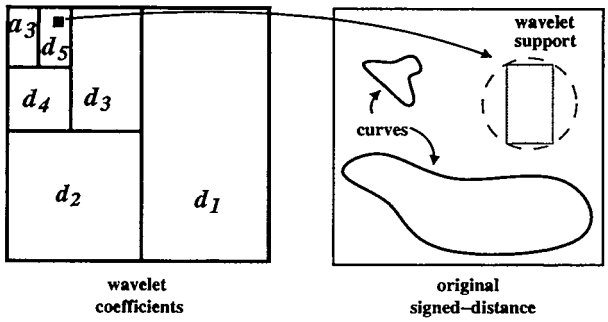


Figure 5. Distance based thresholding: the coefficient in d_5 is set to zero because its support shown on the right does not overlap the curves.

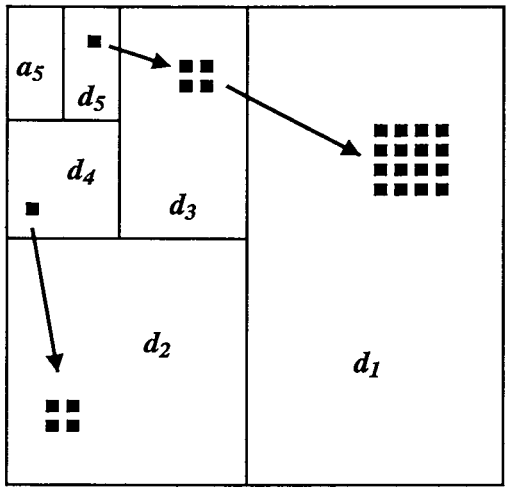


Figure 6. Hierarchies of wavelet coefficients for the zero tree relation. If all of the coefficients are less than a given threshold, the entire tree can be skipped until later in the coding process.

is set to zero. If the radius of the bounding sphere is greater than the distance value, then the surface must intersect the sphere. For these coefficients, the distance values in the support of the wavelet are examined, and the coefficient is set to zero if all of the distance values are the same sign.

6. Zero Tree Coding

A progressive wavelet coder should send the most significant bits of the most significant wavelet coefficients first. This amounts to encoding the locations of the significant coefficients as efficiently as possible. A zero tree coder [22] generates a progressive bit stream by utilizing the observation that wavelets decay in magnitude at finer resolutions. That is, if one defines a hierarchy of wavelet coefficients from one subband to the next it is likely that the child coefficients will be smaller than the parent.

A zero tree is defined as a hierarchy of coefficients for which $c \leq T$ for every coefficient c in the hierarchy, where T is a threshold used to determine the significance of any given coefficient. The zero tree hierarchy is based on observations of the decay of wavelet coefficients [22] in image data and is independent of the support of the wavelets. The zero tree relation is defined for a quadtree-like hierarchy in 2D and an octree-like hierarchy in 3D. Figure 6 depicts the 2D case for our subband ordering. Two hierarchies are shown, one for the x direction (subbands 1, 3, and 5) and one for the y direction (subbands 2 and 4). A zero tree coder is particularly well suited to the distance based thresholding method as the thresholded coefficients are spatially contiguous.

The coding algorithm repeatedly traverses the wavelet coefficients in a predefined order. At any point in the coding process the wavelet coefficients are divided into two groups: those that are not yet significant, and those that have been found to be significant during the current traversal or a preceding one. The threshold T starts at half the value of the largest wavelet coefficient, and is divided in half after each traversal. Thus, each traversal progressively adds more wavelet coefficients to the significant group and removes them from the insignificant group. A zero tree is coded by a single symbol the informs the decoder that every coefficient in the hierarchy is insignificant with respect to the current threshold. Zero trees efficiently encode the positions of the insignificant coefficients. Once a coefficient is deemed significant, its sign bit and its most significant bit are transmitted. On each subsequent traversal another bit is added to its representation. Our implementation follows [22] which contains pseudocode and a small worked example of the algorithm.

7. Results

We demonstrate our system on two surfaces. First, a horse model [5] provides a comparison of the performance of the signed-distance volume approach with the subdivision surface approach in [14]. Second, we compress a large isosurface to demonstrate the ability of our system to represent complex surfaces with many components. All file sizes are the result of applying the gzip utility to the progressive bit-streams resulting from our zero tree coder. The images referenced in this section were generated by extracting a triangulation from the distance field and rendering the triangles with smooth (Gouraud) shading.

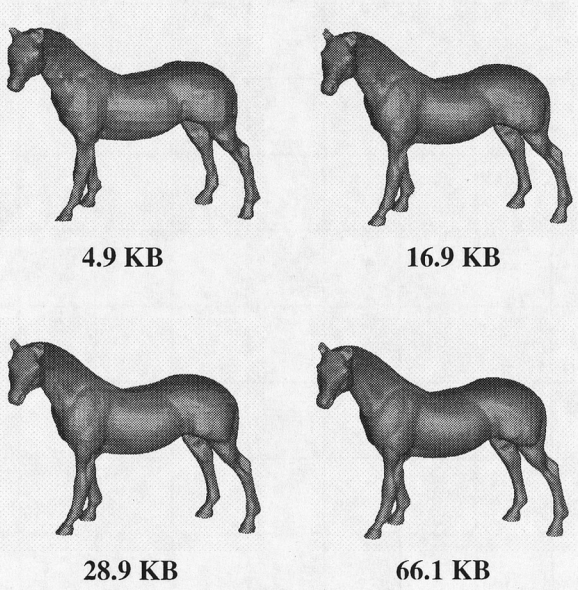


Figure 7. Progressive reconstructions of the horse model from a compressed distance volume of 96x208x173 samples.

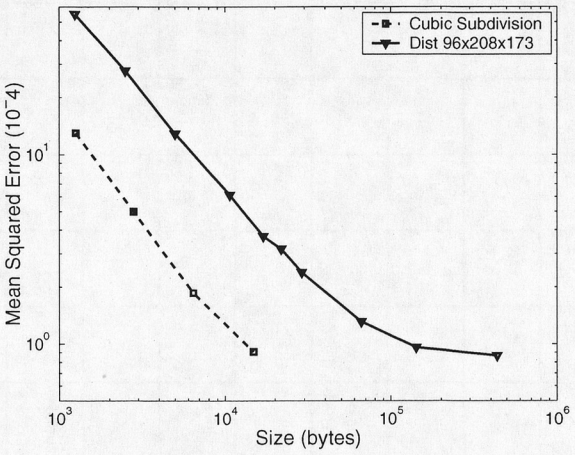


Figure 8. Comparison of cubic subdivision surface compression and wavelet signed-distance volume compression for the horse model. The signed-distance volume contains 96x208x173 samples.

Figure 7 shows four progressive reconstructions of the horse model for increasing file sizes. The original horse model consists of 48485 vertices and 145450 triangles. A good lossless non-progressive mesh compression algorithm [24] generates between $8-15$ bits per vertex. Thus, one could expect the lossless non-progressive compression of the horse mesh to produce a file between $48-80$ kilobytes in size. Our progressive coding begins to represent the muscle structure of the horse at about 30 kilobytes. Our algorithm is of course a lossy compression technique.

Figure 8 compares the performance of the subdivision surface method of Khodakovsky [14] with our approach. The vertical axis in figure 8 shows the mean-squared geometric error defined in section 3.1 normalized to the bounding box diagonal. The cubic subdivision surface approach produces smaller compressed sizes than the distance-volume method for this smooth model. The base mesh used for the subdivision method consists of 122 vertices. This small size enables quite efficient compression of the base mesh. The subdivision-surface method utilizes cubic B-spline wavelets, while the distance volume is decomposed over linear B-spline wavelets. The distance-volume curve approaches a minimum possible error at large file sizes. This limiting error is determined by the sampling rate of the distance volume.

Figure 9 shows four reconstructions of a complex isosurface compressed with our system. The data set depicts the turbulent mixing of two fluids [18] at a resolution of 2048x2048x1920. The isosurface shown here derives from a subset of 256x256x384 samples. The distance transform was computed at the same resolution as the subset of the scalar field. Our system retains the major features of the surface for small sizes and achieves good compression through topological modification. This surface is less well suited to subdivision surface techniques as the size of the base mesh is much larger due to the complex topology and large number of components.

Data Set	Zero Set (s)	Propagation (s)
Horse (96x208x173)	95	54
Turbulence (256x256x128)	17	157

Table 2. Elapsed times for the zero set initialization and closest point propagation steps of the distance transform.

Finally, we present performance results for our system. Table 2 shows the time required for each stage of the distance transform. All times are reported for an SGI O2 work station with 320 megabytes of ram and a 200 megahertz MIPS R5000 processor. A smaller version of the turbulence data was used to avoid virtual memory effects. The zero set initialization cost of triangle meshes is due to the large number of intersection computations. In contrast, the simple linear approximation used for scalar field data is very efficient. The propagation time is manageable even for large data sets.

Some improvement is possible if the distance samples are reordered to improve cache coherence when visiting a sample's 26-neighborhood.

The wavelet transform required $1 - 10$ seconds depending on the resolution of the data. This range of times reflects data sets capable of fitting in the main memory of our workstation. The distance based thresholding required 14 seconds for a 32x68x57 distance volume of the horse model and 52 seconds for a 96x208x173 volume. The threshold time was roughly 2 minutes for the 256x256x384 turbulence data. The turbulence data was processed on an Onyx machine capable of holding all run time data structures in main memory. The threshold times are the result of examining samples of the distance volume multiple times for each wavelet coefficient near the surface. The zero tree coder took from 0.5 to 3 seconds for the horse model at various resolutions and geometric errors. The zero tree coder took between 1 and 20 seconds for the isosurface depending on the number of bits produced.

8. Discussion

Our algorithm successfully produces progressive encodings of signed-distance volumes. However, subdivision surface approaches still produce more compact surface representations for smooth surfaces. One drawback of our current implementation is the use of the gzip utility as a backend. This is less efficient than an entropy coding technique specifically tailored for wavelet transforms. An arithmetic coder applied to the coefficient magnitudes should reduce the file sizes even further.

The geometric error for small bit counts can be improved by modifying the ordering of the bits. A standard zero tree coder assumes an L_∞ error because the significance test depends only on the value of the coefficient and not on the size of the corresponding wavelet support. It is possible for wavelet coefficients at the finest level to have bits emitted along with coefficients at a coarser level. However, the mean-squared geometric error metric integrates over the surface area, implying that the significance test should include the support of the wavelet. Incorporating the support of the wavelets would insure that all of the early bits increase the accuracy of the coarser scale wavelets instead of potentially adding fine scale wavelets, thus improving the overall error.

The encoding process should be modified so that the topology is simplified at early stages and is refined as more bits are added. Our implementation does not track topology changes and allows both simplifications and refinements to occur. This can produce holes in thin shapes at small numbers of bits that disappear later in the decoding process. A method for ordering the topology changes and increasing the significance of the wavelet coefficients affecting those areas could mitigate this problem. Topological modification requires that new error metrics be constructed that do not overly penalize surfaces for which many small components have been eliminated in favor of larger ones. The error metric used in this paper over-emphasizes the errors of small components that have been removed at early stages of reconstruction.

Finally, an important area of future work is to characterize the trade offs between low-order compact basis functions and higher order basis functions with large support. For smooth functions, higher order wavelets are more efficient and exhibit faster convergence than lower order wavelets. However, higher order basis functions have larger support, resulting in fewer thresholded coefficients.

9. Conclusion

We have presented an algorithm that produces progressively compressed signed-distance volumes. Our method does not require a re-meshing step and can handle surfaces with an arbitrary number of components given an appropriate sampling rate. Our representation does not explicitly represent the surface topology, enabling topological modification without complicating the data structures used in the implementation. We believe our approach is best suited to surfaces with complicated topology and many components. Time-varying surfaces pose many problems that can be overcome with an implicit representation. We believe our compression techniques can be extended to the time domain to produce an efficient yet simple surface representation.

10. Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

References

[1] M. Bertram, M. A. Duchaineau, B. Hamann, and K. I. Joy. Bicubic subdivision-surface wavelets for large-scale isosurface representation and visualization. In *Proc. of the 11th Ann. IEEE Visualization Conference (Vis) 2000*, 2000.

[2] M. Bertram, M. A. Duchaineau, B. Hamann, and K. I. Joy. Generalized b-spline subdivision-surface wavelets for geometry compression. *Submitted for publication*, 2001.

[3] D. E. Breen, S. Mauch, and R. T. Whitaker. 3d scan conversion of csg models into distance volumes. In *Proc. 1998 Symposium on Volume Visualization*, pages 7–14, Oct 1998.

[4] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998.

[5] Cyberware, Inc. www.cyberware.com. 2110 Del Monte Avenue, Monterey, California 93940.

[6] S. F. Gibson. Using distance maps for accurate surface representation in sampled volumes. In *Proceedings of the 1999 Conference on Volume Visualization*, pages 23–30, Oct 1998.

[7] L. Grisoni, B. Crespin, and C. Schlick. Multiresolution implicit representation of 3d objects. Technical Report 1217-99, 1999.

[8] L. Grisoni and C. Schlick. Multiresolution representation of implicit objects. In *Implicit Surfaces*, pages 1–10, 1998.

[9] M. H. Gross, L. Lippert, R. Dittrich, and S. Häring. Two methods for wavelet-based volume rendering. *Computers & Graphics*, 21(2):237–252, Mar. 1997. ISSN 0097-8493.

[10] R. Grosso, T. Ertl, and J. Aschoff. Efficient data structures for volume rendering of wavelet-compressed data. In *Winter School of Computer Graphics 1996*, Feb. 1996. held at University of West Bohemia, Plzen, Czech Republic, 12-16 February 1996.

[11] I. Guskov, A. Khodakovsky, P. Schröder, and W. Sweldons. Hybrid meshes. Submitted for publication, 2001.

[12] J. Huang, Y. Li, R. Crawfis, S.-C. Lu, and S.-Y. Liou. A complete distance field representation. In T. Ertl, K. Joy, and A. Varshney, editors, *Proceedings of the 2001 IEEE Conference on Visualization (VIS-01)*, pages 247–254, N.Y., Oct. 21–26 2001. ACM Press.

[13] I. Ihm and S. Park. Wavelet-based 3D compression scheme for very large volume data. In *Graphics Interface*, pages 107–116, June 1998.

[14] A. Khodakovsky, P. Schröder, and W. Sweldens. Progressive geometry compression. In K. Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, Annual Conference Series, pages 271–278. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[15] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3D scanning of large statues. In K. Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, Annual Conference Series, pages 131–144. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[16] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Patt. Recog. and Mach. Intell.*, 11(7):674–693, July 1989.

[17] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, 1998.

[18] A. A. Mirin, R. H. Cohen, B. C. Curtis, W. P. Dannevik, A. M. Dimits, M. A. Duchaineau, D. E. Eliason, D. R. Schikore, S. E. Anderson, D. H. Porter, P. R. Woodward, L. J. Shieh, and S. W. White. Very high resolution simulation of compressible turbulence on the IBM-SP system. In ACM, editor, *Super Computing 1999, Oregon*. ACM Press and IEEE Computer Society Press, 1999.

[19] S. Muraki. Approximation and rendering of volume data using wavelet transforms. In *Proc. IEEE Visualization*, pages 21–28, 1992.

[20] S. Muraki. Volume data and wavelet transforms. 13(4):50–56, July 1993.

[21] S. Muraki. Multiscale volume representation by a dog wavelet. In *IEEE Trans. on Visualization and Computer Graphics*, volume 1, pages 109–116, 1995.

[22] A. Said and W. Perlman. A new, fast, and efficient image codec based on set partioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3):243–250, 1996.

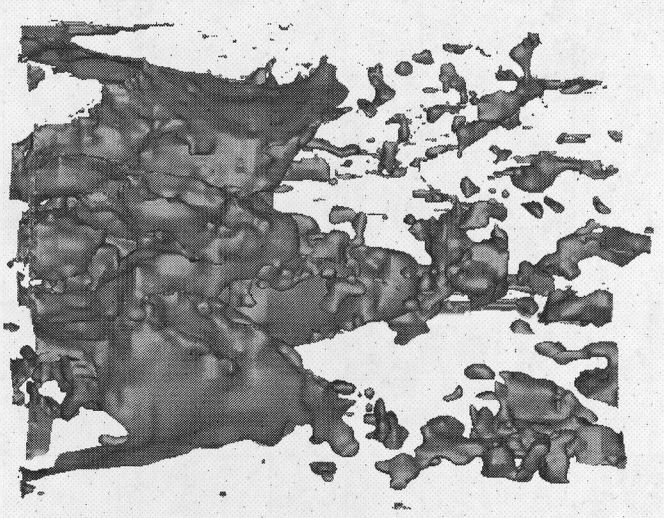
[23] H. Tao and R. J. Moorhead. Progressive transmission of scientific data using biorthogonal wavelet transform. In R. D. Bergeron and A. E. Kaufman, editors, *Proceedings of the Conference on Visualization*, pages 93–99, Los Alamitos, CA, USA, Oct. 1994. IEEE Computer Society Press.

[24] C. Touma and C. Gotsman. Triangle mesh compression. In *Graphics Interface*, pages 26–34, 1998.

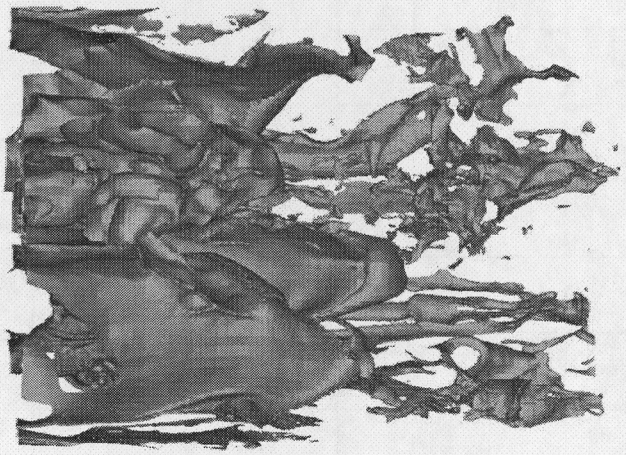
[25] L. Velho, D. Terzopoulos, and J. de Miranda Gomes. Multiscale implicit models. In *Proc. of SIBGRAPI*, pages 93–100, 1994.

[26] R. T. Whitaker and D. E. Breen. Level-set models for the deformation of solid objects. In *Proc. of the 3rd International Workshop on Implicit Surfaces*, pages 19–35, June 1998.

6.3 KB



27.6 KB



661 KB

1.4 MB

Figure 9. Progressive reconstructions of an isosurface of a turbulent mixing simulation.