Octree approximation and compression methods*

Hanan Samet
Computer Science Department
Center for Automation Research
Institute for Advanced Computer Studies
University of Maryland
College Park, Maryland 20742
hjs@cs.umd.edu
www.cs.umd.edu/~hjs

Andrzej Kochut
Computer Science Department
University of Maryland
College Park, Maryland 20742
kochut@cs.umd.edu
www.cs.umd.edu/~kochut

Abstract

Techniques are presented to progressively approximate and compress in a lossless manner two-colored (i.e. binary) 3D objects (as well as objects of arbitrary dimensionality). The objects are represented by a region octree implemented using a pointerless representation based on locational codes. Approximation is achieved through the use of a forest. This method labels the internal nodes of the octree as GB or GW, depending on the number of children being of type GB or GW. In addition, all BLACK nodes are labeled GB, while all WHITE nodes are labeled GW. A number of different image approximation methods are discussed that make use of a forest. The advantage of these methods is that they are progressive which means that as more of the object is transmitted, the better is the approximation. This makes these methods attractive for use on the worldwide web. Progressive transmission has the drawback that there is an overhead in requiring extra storage. A progressive forest-based approximation and transmission method is presented where the total amount of data that is transmitted is not larger than MIN(B,W), where B and W are the number of BLACK and WHITE blocks, respectively, in the region octree of the set of objects.

1 Introduction

The transmission of graphical objects over the Internet is gaining importance. Clearly, it is desirable to transmit the minimum possible amount of data that is needed to rebuild the original object. However, it is also important to present the data as soon as possible. Thus the ideal transmission method should not only achieve compression, but should also be progressive. This means that we want to have a rough idea of the shape and form of the object before it has been transmitted in its entirety. In this paper we propose such a method for a collection of three--dimensional objects that is represented by a region octree (e.g., [5, 9]). Our method is a direct extension of a technique that we developed for a collection of two-dimensional objects represented by a region quadtree [13, 14] implemented using a pointerless representation based on locational codes (e.g., [3]). This technique is both progressive and can achieve compression. Although the extension that we describe is straightforward, the proof that it can achieve compression is different in that it can be extended to arbitrary dimensions whereas the previous approach could not.

Some other work in the area includes that of Sloan and Tanimoto [16] who treat the problem of transmitting a two-dimensional image by successively approximating it by use of pyramid-based approaches [18]. They are able to deal with gray scale images; however, their method does not feature any compression. Knowlton [7] addresses a similar problem by making use of a bintree [15, 17]. Here much of the compression is obtained by using special coding techniques to encode primitive 2×3 blocks. The method that we describe does not make of any such techniques.

In the case of three-dimensional objects, most of the research on compression and transmission has concentrated on data represented by polygonal meshes. For example, Deering [2] presents a method for triangle compression which makes use of a generalized triangle mesh, delta compression, and modified Huffman compression. However, the issue of progressiveness is not addressed. Moreover,

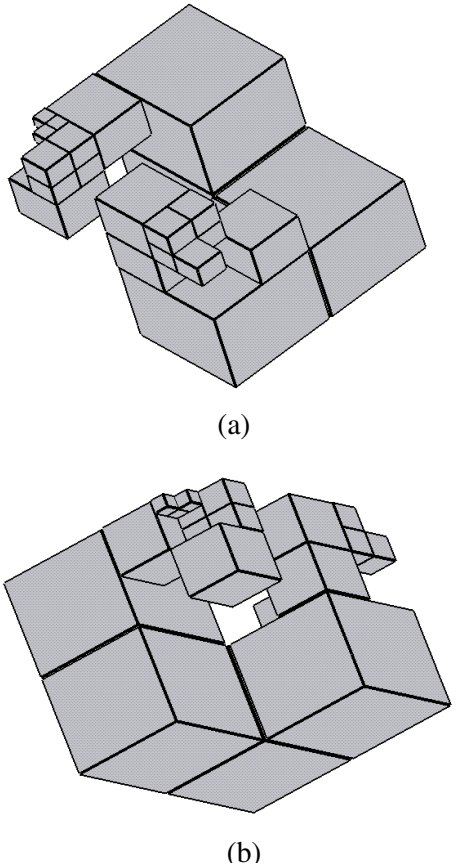
*This work was supported in part by the National Science Foundation under Grants EIA-99-00268, IIS-00-86162, and EIA-00-91474.

Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02)
0-7695-1521-5/02 $17.00 © 2002 IEEE

the compression may be lossy. Li et al. [8] address the problem of progressiveness by using a layer approach to build a hierarchical structure on the basis of the polygonal mesh. In this case, the first approximation is built by collapsing vertices in the original mesh. Subsequent approximations are achieved by adding previously removed vertices. Taubin [19] gives an interesting overview of mesh compression methods.

A different approach is used in the broadly-researched wavelet-based image compression domain. This has led to the JPEG 2000 compression standard [11]. The JPEG compression may lead to lossy images, which is not the case in our approach. The JPEG standard is intended to cover a broad variety of applications in image and signal compression. Naveen and Woods [10] and Glenn et al. [4] are examples of wavelet-based methods for compression of 3D objects.

The rest of this paper is organized as follows. Although our method can be extended to arbitrary dimensions, we focus on the three-dimensional case as the proof that it exhibits compression is easily extended to higher dimensions. Section 2 reviews the region octree as well as the pointerless representation that we use. It also reviews a number of existing octree approximation methods, and presents the family of forest-based approximation methods that we use. Section 3 analyzes the compression that is achievable when employing our forest-based approximation. Section 4 gives the results of experiments, while Section 5 contains concluding remarks and directions for future research.



(a)

(b)

Figure 1: Two orthogonal projections of the region octree decomposition of a sample object.

2 Octree representation

Given a $2^n \times 2^n \times 2^n$ array of voxels, a region octree is constructed by repeatedly subdividing the array into octants, suboctants, ... until obtaining blocks which consist of a single value. This process is usually represented by a tree of out degree 8 in which the root node corresponds to the entire array, while the 8 children of the root node correspond to the octants UNW, UNE, USW, USE, DNW, DNE, DSW and DSE. The leaf nodes correspond to those blocks of the array for which no further subdivision is necessary. The nodes at level k (if any) represent blocks of size $2^k \times 2^k \times 2^k$, and are often referred to as nodes of size 2^k. Thus a node at level 0 corresponds to a single voxel in the voxel array, while the single node at level n corresponds to the root of the region octree. For example, Figure 1 shows two orthogonal projections of the region octree block decomposition of a sample object, while Figure 2 is its tree representation.

We assume that the set of objects to be transmitted is two-colored with BLACK (WHITE) denoting voxels that are (not) occupied by the objects. The blocks in the region octree are labeled BLACK (WHITE) corresponding to the color of their constituent . Blocks corresponding to nonleaf nodes in the tree are labeled GRAY. In addition, each nonleaf node is labeled as GB if at least four of its children are BLACK or GB, and as GW otherwise.

In order to efficiently transmit the region octree we need a representation that is more compact than a tree consisting of WHITE, BLACK an GRAY nodes. Instead, we represent the region octree using a pointerless representation which consists of its leaf nodes. Each leaf node q is uniquely identified by the path leading from the root of the octree to q, termed its *locational code*. In particular, we associate a directional code with each direction in the octree. The directions UNW, UNE, USW, USE, DNW, DNE, DSW, DSE are represented by directional codes 1, 2, 3, 4, 5, 6, 7, 8, respectively, Thus the path from root node to any leaf may be represented as a sequence of directional codes. The directional codes are accessed using the function CHILDTYPE. Letting the sequence $< x_i >$ represent directional codes on the path from the root node x_n to node x_m, where $x_i = \mathrm{FATHER}(x_{i-1})$, the locational code for node x_m is given by z_n where z_i is defined as:

Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02)
0-7695-1521-5/02 $17.00 © 2002 IEEE

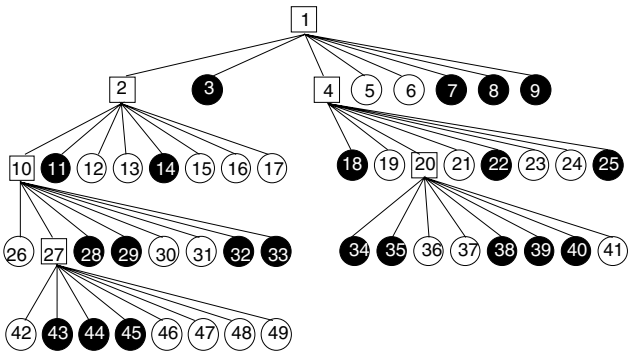


Figure 2: The tree representation of the region octree block decomposition corresponding to the sample object in Figure 1. The internal nodes are depicted as white squares, while the WHITE and BLACK leaf nodes are depicted as white and black circles, respectively. The nodes are referenced in the discussion by their numeric label.

$$z_i = \begin{cases} 0 & i = m \\ 9 \cdot z_{i-1} + \mathrm{CHILDTYPE}(x_i) & m < i \leq n \end{cases}$$

For example, $< 3, 3, 2 >$ is the sequence of directional codes corresponding to node 35 in Figure 2. $\mathrm{CODE}(35) = 3 \cdot 9^0 + 3 \cdot 9^1 + 2 \cdot 9^2 = 192$ is its locational code.

Given the locational codes of the leaf blocks, there is no need for the locational codes of the nonleaf blocks. As pointed out above, the objects that we are approximating are restricted to being two-colored. Thus it is sufficient to transmit the locational codes of the leaf blocks of just one color since the locational codes of the remaining leaf blocks are known to be of the other color and thus it is easy to reconstruct the region octree from the locational codes of the leaf blocks of one color. Such an algorithm is presented in [13] to reconstruct the region quadtree from the locational codes of the BLACK leaf blocks. It can be extended easily to reconstruct the region octree.

2.1 Hierarchical approximation techniques

Since any internal node in the region octree may be used to represent its subtree, the region octree data structure may serve as the basis for an approximation method. Ranade, Rosenfeld, and Samet [12] define the inner and outer approximation of a 2D image. We extend this concept to a set of 3D objects represented by a region octree. The idea is to truncate the region octree at the specified level. More precisely, lets define the inner approximation IB(k) of order k as the set of black nodes at levels $\geq k$. We denote the subset operation $\subseteq$ on two region octrees in such a way so that $A \subseteq B$ means that the volume spanned by region octree A is included in the volume spanned by region octree B. It is easy to see that for any collection of objects in the voxel array V and the inner approximations of its corresponding region octree, $\mathrm{IB}(n) \subseteq \mathrm{IB}(n-1) \subseteq \ldots \subseteq \mathrm{IB}(0) = V$, where n is the number of levels in region octree corresponding to V. Similarly, we can define the outer approximation OB(k) of order k as the set of black and gray nodes at levels $\geq k$. In this case, for any collection of objects in the voxel array V and the outer approximations of its corresponding region octree, $V = \mathrm{OB}(0) \subseteq \mathrm{OB}(1) \subseteq \ldots \subseteq \mathrm{OB}(n)$. Clearly, both of the above approximation techniques are very crude. An approximation obtained in this way does not give a general overview of the connectivity of the object quickly enough. Truncating the region octree at an arbitrarily chosen level does not take into consideration internal structure of particular subtrees that are being truncated. Moreover, if we use one of the above techniques to transmit data would require that we also incur an overhead cost equal to the number of internal nodes in the corresponding region octree.

2.2 Forest-based approximation techniques

As discussed in Section 2.1, simple approximation methods such as the inner and outer approximation not yield satisfying results. To overcome the shortcomings of these techniques, we extend the approach developed by Samet [13] for approximating two-dimensional images to the three-dimensional case. This approach makes use of the concept of a forest which was first introduced by Jones and Iyengar [6]. To make the discussion more precise, let us review the concepts used in that method. Each internal node of the region octree is labeled GB if at least four of its children is black or of type GB. Otherwise we label it GW. Moreover, we will call a node a *maximal black cube* if it is a black leaf node or an internal node of type GB. A *white maximal cube* is a white leaf node or an internal node of type GW. We define a *black forest* as a minimal set of maximal cubes that are not contained in other maximal black cubes, and that span the black volume corresponding to the object. Similarly, the *white forest* is a minimal set of *white cubes* that are not contained in other maximal white cubes and span the white volume of the object. For example in the region octree whose tree representation is given in Figure 2, internal nodes 1, 10, 4 and 20 are of type GB. These nodes, as well as all black leaf nodes, are *maximal cubes*. Nonleaf nodes 2 and 27 are of type GW. The *black forest* associated with the region octree consists only of node 1. Nodes 2, 5, 6, 19, 36, 37, 41, 21, 23, and 24 comprise the *white forest* of the example region octree.

Now, let us define the notion of the approximation of a region octree using *black forests*. First, observe that not

Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02)
0-7695-1521-5/02 $17.00 © 2002 IEEE

only the root of the region octree, but any internal node may be approximated by its *black forest*. Formally, let us define FB(R) to be the region octree obtained by coloring black all root nodes of suboctrees comprising the *black forest* of the region octree rooted at R. Define FB(R, l) — a forest approximation of the region octree rooted at node R of order l — to be those nodes from FB(R) which are at level $\geq l$. For example, in Figure 2, $\mathrm{FB}(2,3) = \{\}$, while $\mathrm{FB}(2,2) = \mathrm{FB}(2,1) = \mathrm{FB}(2,0) = \mathrm{FB}(2) = \{10, 11, 14\}$. It is easy to verify that $\mathrm{FB}(R,n) \subseteq \mathrm{FB}(R,n-1) \subseteq \ldots \subseteq \mathrm{FB}(R,0) = \mathrm{FB}(R)$. The above concepts form the basis of the black forest approximation method, denoted by FBB, and defined as follows. First, approximate the region octree by coloring black all root nodes of suboctrees from FB(root,0). Subsequent levels of approximation are achieved by replacing nonleaf nodes in the approximation by their black forests. Formally, we have:

$$\mathrm{FBB}(i) = \begin{cases} \text{FB(root)} & i = n \\ \{\, j \mid j \in \text{FBB}(i+1) \textbf{ and not } \text{GRAY}(j) \,\} & \\ \quad \cup \{\mathrm{FB}(k) \mid j \in \text{FBB}(i+1) \textbf{ and } \text{GRAY}(j) & \\ \qquad \textbf{and } \mathrm{FATHER}(k) = j\} & \\ & 0 \leq i < n \end{cases}$$

For example, in Figure 2, $\mathrm{FBB}(4) = \{1\}, \mathrm{FBB}(3) = \{10, 11, 14, 3, 4, 7, 8, 9\}$ is achieved by replacing nonterminal node 1 by black forests of its children. FBB(2) is obtained by replacing node 10 by $\{43, 44, 45, 28, 29, 32, 33\}$, and node 4 by $\{18, 20, 22, 25\}$. The resulting approximation is $\mathrm{FBB}(2) = \{\, 43, 44, 45, 28, 29, 32, 33, 11, 14, 3, 18, 20, 22, 25, 7, 8, 9\}$. FBB(1) is obtained by replacing node 20 with its children yielding $\mathrm{FBB}(1) = \{43, 44, 45, 28, 29, 32, 33, 11, 14, 3, 18, 34, 35, 38, 39, 40, 22, 25, 7, 8, 9\}$. $\mathrm{FBB}(0) = \mathrm{FBB}(1) = V$, because all nodes in FBB(1) are leaf nodes. Clearly, $V = \mathrm{FBB}(0) \subseteq \mathrm{FBB}(1) \subseteq \ldots \subseteq \mathrm{FBB}(n)$.

Similar ideas may be used to define a white forest approximation method. First, let us define FW(R), to be the region octree obtained by coloring white all root nodes of suboctrees comprising the *white forest* of the region octree rooted at R. To follow the concept of approximation more closely, we introduce FW(R, l) in a similar fashion to FB(R, l). In particular, FW(R, l) is the set of all nodes from FW(R) which are at level $\geq l$. For example, in Figure 2, $\mathrm{FW}(10,2) = \{\}$, $\mathrm{FW}(10,1) = \{26, 27, 30, 31\}$, $FW(10,0) = FW(10,1) = FW(10)$. Clearly, $\mathrm{FW}(R,n) \subseteq \mathrm{FW}(R,n-1) \subseteq \ldots \subseteq \mathrm{FW}(R,0) = \mathrm{FW}(R)$. The above concepts form the basis of the white forest approximation method, denoted by FWW, and defined in an analogous manner to that of FBB as follows.

$$\mathrm{FWW}(i) = \begin{cases} \text{FW(root)} & i = n \\ \{\, j \mid j \in \text{FWW}(i+1) \textbf{ and not } \text{GRAY}(j) \,\} & \\ \quad \cup \{\mathrm{FW}(k) \mid j \in \text{FWW}(i+1) \textbf{ and } \text{GRAY}(j) & \\ \qquad \textbf{and } \mathrm{FATHER}(k) = j\} & \\ & 0 \leq i < n \end{cases}$$

For example, in Figure 2, $\mathrm{FWW}(4) = \{2, 19, 36, 37, 41, 21, 23, 24, 5, 6\}$. FWW(3) is obtained by replacing nonleaf node 2 by its black forest $\{26, 27, 30, 31, 12, 13, 15, 16, 17\}$ yielding $\mathrm{FWW}(3) = \{26, 27, 30, 31, 12, 13, 15, 16, 17, 19, 36, 37, 41, 21, 23, 24, 5, 6\}$. The white forest approximation of order 2 is obtained by replacing the only remaining nonleaf node 27 by nodes $\{42, 46, 47, 48, 49\}$. After this step, no nonleaf nodes remain in the approximation set and thus $\mathrm{FWW}(2) = \mathrm{FWW}(1) = \mathrm{FWW}(0) = \{26, 30, 31, 12, 13, 15, 16, 17, 19, 36, 37, 41, 21, 23, 24, 42, 46, 47, 48, 49, 5, 6\}$. It can be easily seen that $\overline{V} = \mathrm{FWW}(0) \subseteq \mathrm{FWW}(1) \subseteq \ldots \subseteq \mathrm{FWW}(n)$, where $\overline{V}$ is the complement of V.

Note that the FWW approximation is the dual of the FBB approximation. In particular, instead of approximating the black part of the volume, it approximates the white part of the volume. Observe also that FWW underestimates the black volume of the object. This is caused by the fact that FWW overestimates the white volume. Thus FBB and FWW are complementary methods. This observation leads to our final approximation method which alternates between FBB and FWW. We use FBW to denoted this method. In particular, without loss of generality, the first approximation is given by FBB(n). The second level of approximation is obtained by replacing nonleaf nodes in FBB(n) by their white forests. The third level of approximation is obtained by replacing nonleaf nodes by their black forests. More precisely, we have:

$$\mathrm{FBW}(i) = \begin{cases} \text{empty} & i = n+1 \\ \text{FBB}(n) & i = n \\ \mathrm{FBW}(i+1) & \\ \quad \cup \{\mathrm{FW}(j) \mid j \in \mathrm{FBW}(i+1) & \\ \qquad \textbf{and } j \notin \mathrm{FBW}(i+2)\} & \\ \qquad (n-i) \bmod 2 = 1 \textbf{ and } i < n & \\ \mathrm{FBW}(i+1) & \\ \quad \cup \{\mathrm{FB}(j) \mid j \in \mathrm{FBW}(i+1) & \\ \qquad \textbf{and } j \notin \mathrm{FBW}(i+2)\} & \\ \qquad (n-i) \bmod 2 = 0 \textbf{ and } i < n & \end{cases}$$

As an example, let us examine the formation of FBW on the example region octree in Figure 2. FBW(4) is given by the black forest approximation of the root. Thus $\mathrm{FBW}(4) = \{1\}$. The next level of approximation is obtained by adding the white forest of node 1 yielding $\mathrm{FBW}(3) = \{1, 2, 19, 36, 37, 41, 21, 23, 24, 5, 6\}$. Next, we use the black forest approach to obtain $\mathrm{FBW}(2) = \{1, 2, 19, 36, 37, 41, 21, 23, 24, 5, 6, 10, 11, 14\}$. FBW(1) is obtained by adding the white forests of nonleaf nodes

Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02)
0-7695-1521-5/02 $17.00 © 2002 IEEE

from FBW(2) to yield $\mathrm{FBW}(1) = \{1, 2, 19, 36, 37, 41, 21, 23, 24, 5, 6, 10, 11, 14, 18, 20, 22, 26, 27, 30, 31, 36, 37, 41\}$. Finally, adding the black forest of node 27 yields $\mathrm{FBW}(0) = \{1, 2, 19, 36, 37, 41, 21, 23, 24, 5, 6, 10, 11, 14, 18, 20, 22, 26, 27, 30, 31, 36, 37, 41, 43, 44, 45\}$. The approximation process is illustrated on Figure 3 for the object whose orthogonal projection is given in Figure 1a.

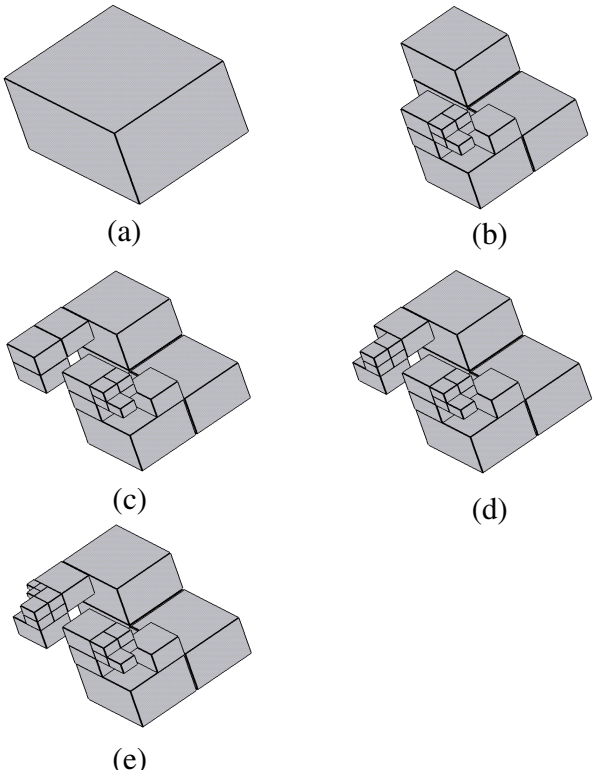


Figure 3: Example illustrating the application of the FBW approximation to the example object from Figure 1. The objects in (a)–(e) show the progressive aspect of the approximations with (a) corresponding to FBW(4) and (e) corresponding to FBW(0). Observe how the FBW approximation alternates between over-estimation and under-estimation of the black volume.

The FBW approximation alternates between the FBB and FWW approximation methods. The first approximation is achieved using FBB. However, we could also consider starting the approximation sequence with FWW. In fact, in the context of discussing the compression properties of the forest-based approximation methods in Section 3, it is shown that the FWB approximation turns out to be preferable for region octrees whose root node is of type GB. The FWB approximation technique is defined as follows:

$$\mathrm{FWB}(i) = \begin{cases} \text{empty} & i = n+1 \\ \mathrm{FWW}(n) & i = n \\ \mathrm{FWB}(i+1) & \\ \quad \cup\{\mathrm{FB}(j) \mid j \in \mathrm{FWB}(i+1) & \\ \qquad \textbf{and } j \notin \mathrm{FWB}(i+2)\} & \\ \qquad (n-i) \bmod 2 = 1 \textbf{ and } i < n & \\ \mathrm{FWB}(i+1) & \\ \quad \cup\{\mathrm{FW}(j) \mid j \in \mathrm{FWB}(i+1) & \\ \qquad \textbf{and } j \notin \mathrm{FWB}(i+2)\} & \\ \qquad (n-i) \bmod 2 = 0 \textbf{ and } i < n & \end{cases}$$

As an example, let us examine the formation of FWB on the example region octree in Figure 2. The first approximation is the white forest of the root node and thus $\mathrm{FWB}(4) = \{2, 19, 36, 37, 41, 21, 23, 24, 5, 6\}$. The next level of approximation is obtained by adding the black forest of node 2 yielding $\mathrm{FWB}(3) = \{2, 19, 36, 37, 41, 21, 23, 24, 5, 6, 10, 11, 14\}$. Next, we use the white forest approach to obtain $\mathrm{FWB}(2) = \{2, 19, 36, 37, 41, 21, 23, 24, 5, 6, 10, 11, 14, 26, 27, 30, 31\}$ by adding the white forest of nonleaf node 10. Finally, FWB(1) is obtained by adding the black forest of node 27 to yield $\mathrm{FWB}(1) = \{2, 19, 36, 37, 41, 21, 23, 24, 5, 6, 10, 11, 14, 26, 27, 30, 31, 43, 44, 45\}$. Because FWB(1) does not contain any nonleaf nodes, $\mathrm{FWB}(0) = \mathrm{FWB}(1)$.

It is easy to reconstruct the original image from the FBW approximation. The algorithm need not know the colors or types (i.e., GB or GW) of the blocks that are being transmitted. All that is needed is to make sure that blocks P and Q are added to the tree in such an order so that if P is an ancestor of Q, then P is added before Q. If there is no ancestral/descendant relation between two nodes, then they can be added in an arbitrary order. Of course, we know that the first nodes are BLACK (WHITE) in the case of an FBW and FWB (FBW' and FWB') approximation.

3 Compression

As pointed out above, a two-colored region octree is defined by enumerating the locational codes of all of its WHITE nodes, or equivalently by enumerating the locational codes of all of its BLACK nodes. Depending on the set of objects, in order to reduce the amount of space that is needed, we use the color with the smaller cardinality. Given a region octree having B and W black and white nodes, respectively, and encoding it using approximation method M with $\mathrm{F}(M)$ nodes, then we say that M achieves compression whenever $\mathrm{F}(M) < \mathrm{MIN}(\mathrm{B}, \mathrm{W})$. We define the compression ratio of method M as the number of nodes in the approximation sequence divided by cardinality of the minimal color, i.e., $\mathrm{C}(M) = \mathrm{F}(M)/\mathrm{MIN}(B,W)$. In this section we show that variants of FBW can be constructed so that $\mathrm{F}(M) \leq \mathrm{MIN}(B,W)$ (where M is one of the variants

Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02)
0-7695-1521-5/02 $17.00 © 2002 IEEE

of FBW chosen depending on the type of the root node of the region octree).

We first prove four theorems setting upper bounds on the number of nodes in FBW and its variants. Our proofs are different, and more general, than the ones used to prove the existence of compression for the use of these methods in the two-dimensional case in [13]. In particular, we cannot assume that all the brothers of a GB (GW') node are of type GW (GB'). Thus we resort to a method that groups brother nodes of the same type (i.e., GB, GW, GB', or GW').

Theorem 3.1 *The maximum number of nodes in an FBW approximation is less than or equal to one plus the number of WHITE nodes in the region octree (i.e., F(FBW)≤W+1).*

Proof: The proof is by induction on the tree structure as well as on the approximation sequence. We consider two cases. The first is when the root of the region octree is of type GB, and the second is when the root is of type GW. In the following discussion we extend the definitions of GB and GW to include leaf BLACK nodes and leaf WHITE nodes, respectively.

Case a: The root is of type GB. Thus FBW(n)={root}. Let us define FBWN(i)= $\{j | j \in$FBW(i) **and** $j \notin$FBW($i+1$) $\}$ for $0 \leq i \leq n$ and FBWN($n+1$) is empty. Observe, that FBWN(i) denotes the set of new nodes added to the approximation by step i. Let w(z) denote the number of WHITE terminal nodes in the region octree rooted at node z. Let F(z) denote the number of nodes in the FBW approximation of the region octree rooted at z assuming that the root is of type GB and not including the root. So, to prove our theorem we need to show that F(z)≤W. Suppose that FBWN($n-1$) has m elements. For each element $x_i \in$FBWN($n-1$) that is not a terminal node, FBWN($n-2$) contains all descendants of x_i of type GB that have no ancestor of type GB in the subtree rooted at x_i. Now, let us partition the descendants of x_i into y_i groups, where each group contains those nodes which are brothers in the region octree. Observe that for each $z \in$FBWN($n-2$), FATHER(z) = GW. The reason is that z has no ancestor of type GB in the subtree rooted at x_i, and x_i is itself of type GW. Therefore, z has at least 5 brothers of type GW and at most 2 brothers of type GB. Therefore, the number of nodes of type GW subsumed by FBW(n) and FBW($n-1$) is at least $5 \cdot y_i$. It is more if the brothers of z of type GW correspond to nonleaf nodes. Now, assuming that the claim holds for trees of size $n-2$ — that is, F(z)≤w(z), we have:

$$\text{F(root)} \leq 1 + m + \sum_{i=1}^{m} 3 \cdot y_i + \sum_{z \in \text{FBWN}(n-2)} F(z)$$

$$\leq 1 + \sum_{i=1}^{m} 1 + \sum_{i=1}^{m} (3 \cdot y_i) + \sum_{z \in \text{FBWN}(n-2)} \text{w}(z)$$

$$\leq 1 + \sum_{i=1}^{m} (3 \cdot y_i + 1) + \sum_{z \in \text{FBWN}(n-2)} \text{w}(z).$$

However, use of $3 \cdot y_i + 1$ GB nodes in FBW($n-2$) results in subsuming at least $5 \cdot y_i$ GW nodes. The fact that $(3 \cdot y_i + 1) < 5 \cdot y_i$ means that F(root) requires fewer nodes than the number of WHITE leaf nodes as $\sum_{z \in \text{FBWN}(n-2)}$ w(z) accounts for all the WHITE leaf nodes in the subtrees rooted in the $3 \cdot y_i + 1$ GB nodes in FBW($n-2$). Therefore, F(root)≤w(root)+1.

Case b: The root is of type GW. Let y denote the number of groups of GB nodes contributed to FBW(n) by the root node. Here we assume the same definition of a group of nodes as in the case a. Observe that there are at most $3 \cdot y$ nodes of type GB in FBWN(n). The reason is that each group has a father of type GW. Similarly, the number of nodes of type GW in all groups is at least $5 \cdot y$, and this is the lower bound on the number of nodes of type GW that are subsumed by FBW(n). Now, assuming that the claim holds for trees of size $n-1$ — that is, F(z)≤w(z), we have:

$$\text{F(root)} \leq 3 \cdot y + \sum_{z \in \text{FBWN}(n-1)} F(z)$$

However, use of $3 \cdot y$ GB nodes in FBW(n) results in subsuming at least $5 \cdot y$ GW nodes. The fact that $3 \cdot y \leq 5 \cdot y$ means that F(root) requires fewer nodes than the number of WHITE leaf nodes as $\sum_{z \in \text{FBWN}(n-1)}$ w(z) accounts for all the WHITE leaf nodes in the subtrees rooted in the $3 \cdot y$ GB nodes in FBW(n). Therefore, F(root)≤w(root). We can get an even tighter bound by noting that $y \geq 1$ and for $y = 1$, we have $\text{F}(root) = 3 = \text{W} - 2$. Therefore, $\text{F}(root) \leq \text{w}(root) - 2$. ∎

Theorem 3.2 *The maximum number of nodes in an FWB approximation is less than or equal to the number of WHITE nodes in the region octree (i.e., F(FWB)≤W).*

Proof: The proof uses the proof of the Theorem 3.1. Again, we consider two cases. The first is when the root of the region octree is of type GB, and the second is when the root is of type GW.

Case a: The root is of type GB. FWB(n) ignores the root and collects all descendants of the root that are of type GW and that do not have an ancestor of type GW. However, as we saw in the proof of case a of Theorem 3.1, these are precisely the nodes that comprise FBW($n-1$). In other words, {root }+FWB(n)=FBW($n-1$). Similarly, {root }+FWB(i)=FBW($i-1$) for $1 \leq i < n$. Since the size of approximation FBW is bounded by W+1 and FWB does not include the root node, the size of FWB is bounded from above by W.

Case b: The root is of the type GW. FWB(n) includes the root. FWB($n-1$) consists of the root and all of its descendants of type GB that do not have an ancestor of type GB. However, as we saw in the proof of case b of Theorem 3.1, these are precisely the nodes that comprise FBW(n). Thus, FWB($n-1$)={root}+FBW(n). Similarly,

Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02)
0-7695-1521-5/02 $17.00 © 2002 IEEE

FWB($i-1$)={root}+FBW(i) for $1 \leq i \leq n$. Since approximation FBW is bounded by W-2, and FWB also includes the root node, the size of FWB is bounded from above by W-1. ∎

Theorems 3.1 and 3.2 let us conclude that the size of the approximation is always bounded from above by the number of WHITE nodes in the region octree. To obtain an analogous result in terms of the BLACK nodes in the region octree, we need to redefine our approximation sequence. To do so, we relabel octree with GB' and GW' as follows. A nonleaf node is said to be of the type GW' if at least four of its children are of type GW' or WHITE; otherwise, the node is said to be of type GB'. We now redefine FB, FW, FBB and FWW in terms of GB' and GW' to yield FB', FW', FBB' and FWW', respectively. At this point, we can define FWB':

$$\mathrm{FWB}'(i) = \begin{cases} \text{empty} & i = n+1 \\ \text{FWW'}(n) & i = n \\ \mathrm{FWB}'(i+1) & \\ \quad \cup\{\mathrm{FB}'(j) \mid j \in \mathrm{FWB}'(i+1) & \\ \qquad \textbf{and } j \notin \mathrm{FWB}'(i+2)\} & \\ \qquad (n-i) \bmod 2 = 1 \textbf{ and } i < n & \\ \mathrm{FWB}'(i+1) & \\ \quad \cup\{\mathrm{FW}'(j) \mid j \in \mathrm{FWB}'(i+1) & \\ \qquad \textbf{and } j \notin \mathrm{FWB}'(i+2)\} & \\ \qquad (n-i) \bmod 2 = 0 \textbf{ and } i < n & \end{cases}$$

We define FBW' in a similar manner as follows:

$$\mathrm{FBW}'(i) = \begin{cases} \text{empty} & i = n+1 \\ \text{FBB'}(n) & i = n \\ \mathrm{FBW}'(i+1) & \\ \quad \cup\{\mathrm{FW}'(j) \mid j \in \mathrm{FBW}'(i+1) & \\ \qquad \textbf{and } j \notin \mathrm{FBW}'(i+2)\} & \\ \qquad (n-i) \bmod 2 = 1 \textbf{ and } i < n & \\ \mathrm{FBW}'(i+1) & \\ \quad \cup\{\mathrm{FB}'(j) \mid j \in \mathrm{FBW}'(i+1) & \\ \qquad \textbf{and } j \notin \mathrm{FBW}'(i+2)\} & \\ \qquad (n-i) \bmod 2 = 0 \textbf{ and } i < n & \end{cases}$$

As we can see, approximations FWB' and FBW' are formed in a manner analogous to the way in which approximations FWB and FBW were formed. The only difference is the interchanging of the roles of WHITE and BLACK nodes. We obtain the following analogs of Theorems 3.1 and 3.2:

Theorem 3.3 *The maximum number of nodes in the FWB' approximation is equal to B+1 when the root node is of type GW' and B-2 when the root node is of type GB'.*

Proof: Replace the role of WHITE nodes with BLACK nodes in the proof of Theorem 3.2. ∎

Theorem 3.4 *The maximum number of nodes in the FBW' approximation is equal to B.*

Proof: Replace the role of WHITE nodes by BLACK nodes in the proof of Theorem 3.1. ∎

Approximation	GB(GB')	GW(GW')
FBW	W+1	W-2
FWB	W	W
FBW'	B	B
FWB'	B-2	B+1

Table 1: Summary of the upper bounds on the number of nodes in discussed approximations.

The results of Theorems 3.1–3.4 are summarized in Table 1. Observe that we may pick the optimal variant of the FBW approximation technique based on the type of the root of the region octree and the number of BLACK and WHITE nodes. For example, if we are given a region octree which has more BLACK nodes than WHITE nodes, and its root is of type GB, then we should choose the FWB approximation. On the other hand, if the number of WHITE nodes is greater than the number of BLACK nodes, and if the root of the region octree is of type GB', then we should choose the FWB'approximation. Thus, as a result we proved that our method consisting of the discussed variants of FBW requires at most MIN(W,B) nodes for representing the complete region octree. However, in practice the compression achieved by the method is usually much better. The MIN(B,W) bound is attained for a checkerboard image. Other images that exhibit more coherence result in more compression.

For example, let us analyze the compression that can be obtained for the example region octree in Figure 4a. The number of WHITE nodes is 14 while the number of BLACK nodes is 15. The FBW approximation consists of a total of 3 nodes and is formed as follows: FBW(4)= $\{2, 10\}$ and FBW(3) = $\{2, 10, 28\}$. All remaining approximations are equal to FBW(3) because there are no nonleaf nodes in FBW(3). The compression ratio for this region octree is equal to $3/14$. In contrast, Figure 4b shows a region octree for which the number of nodes in the approximation is exactly equal to the value of MIN(W,B). In fact, the number of WHITE nodes is equal to 1, and the FWB approximation (which is the best choice in case of a region octree with a small number of WHITE nodes and a root of type GB) contains node 1. Thus in this case the bound is met.

4 Empirical results

In order to evaluate the different methods, we applied them to the region octree corresponding to the bunny in

Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02)
0-7695-1521-5/02 $17.00 © 2002 IEEE

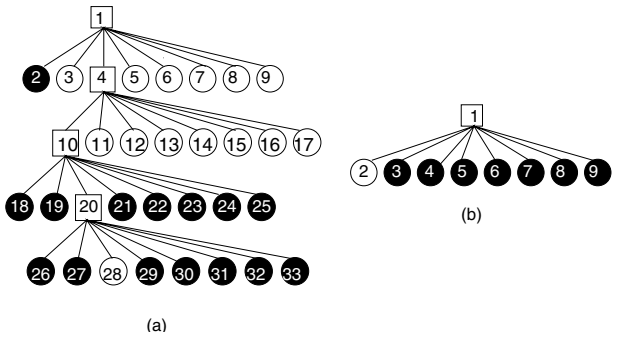


Figure 4: Two octrees showing compression properties. Approximation of the tree (a) gives compression of 3/14, while tree (b) does not get compressed during the approximation.

Figure 5. The maximum depth of the region octree is 7, and thus the resolution is $128 \times 128 \times 128$ voxels. The corresponding region octree consists of 46093 BLACK and 48114 WHITE nodes. In order to compare our approximation technique with the OB approximation, we applied FBW and OB to the octree representing the bunny. The results obtained by the experiments are presented in Table 2. Table 2a shows the number of blocks needed by the OB approximation for each approximation level. For example, in order to transmit OB(4) of the bunny, we need to send 299 blocks. Table 2b shows corresponding results for the FBW approximation. The FBW approximation needs 29007 blocks to rebuild the original bunny. Since MIN(B,W)= 46093, use of FBW results in a compression ratio of 0.62.

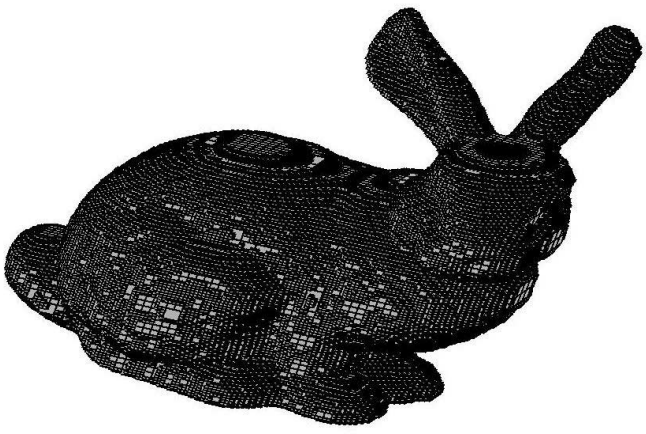


Figure 5: Bunny used in empirical studies.

Figure 6 compares the quality resulting from the use of OB and FBW after transmitting approximately the same number of blocks. Once the first 29007 blocks have been transmitted FBW yields a lossless view of the bunny and

Approx. OB(i)	Blocks sent	Cumulative blocks sent	Leaf blocks at level i		
			BLACK	WHITE	TOTAL
OB(7)	1	1	1	0	1
OB(6)	8	9	8	0	8
OB(5)	47	56	47	17	64
OB(4)	243	299	245	134	379
OB(3)	1222	1521	1250	648	1898
OB(2)	4896	6417	5257	2864	8121
OB(1)	17478	23895	19431	11818	31249
OB(0)	35656	59551	46093	48114	94207

(a)

Approx. OB(i)	Blocks sent	Cumulative blocks sent	Leaf blocks at iteration i		
			BLACK	WHITE	TOTAL
FBW(6)	1498	1498	1498	6259	7757
FBW(5)	13654	15152	27492	19913	47405
FBW(4)	5774	20926	33266	37743	71009
FBW(3)	6373	27299	44057	44116	88173
FBW(2)	1230	28529	45287	47583	92870
FBW(1)	445	28974	46060	48028	94088
FBW(0)	33	29007	46093	48114	94207

(b)

Table 2: Experimental results of applying approximation technique (a) OB and (b) FBW to the bunny object.

is represented by Figure 5. On the other hand, the object produced by use of OB with approximately the same number of blocks (i.e., OB(1) with a cumulative number of 23,895 blocks that have been transmitted) shown in Figure 6 is blocky. In fact, the OB method needs to transmit 35656 more blocks to obtain the original view of the bunny. Figure 7 shows the progress of the FBW technique. The approximation FBW(6) (Figure 7a) requires transmission of only 1498 blocks, but it already gives a good indication of the shape of the bunny's ears. FBW(5) (Figure 7b) which requires transmission of 15152 blocks yields quite a good approximation of the object. Approximations FBW(4) (Figure 7c) and FBW(3) (Figure 7d) require 5774 and 6373 additional blocks, respectively.

The quality of the approximations can also be evaluated by use of the following entropy function [1]:

$$H = -\sum_{i \in \{BLACK, WHITE\}} f(i) * \log_2 f(i)$$

where *f(i)* denotes the observed frequency of blocks with color *i* The entropy function is useful for indicating the extent of the uniformity of the image (i.e., 1 corresponds to uniformity while 0 corresponds to an image that is all black or all white). Table 4 contains the values of the entropy function values for the FBW approximation sequence of the bunny. Note that the entropy of the bunny is 0.828 (i.e., FBW(0)) and that the function oscillates around it in an analogous manner to the way in which FBW over and under approximate the object.

Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02)
0-7695-1521-5/02 $17.00 © 2002 IEEE

Approximation	BLACK voxels	WHITE voxels	Entropy
FBW(6)	908341	1188811	0.987
FBW(5)	490759	1606393	0.785
FBW(4)	575388	1521764	0.848
FBW(3)	545229	1551923	0.827
FBW(2)	548531	1548621	0.829
FBW(1)	547911	1549241	0.826
FBW(0)	547944	1549208	0.828

Table 3: Values of the entropy function for the FBW(i) approximation sequence.

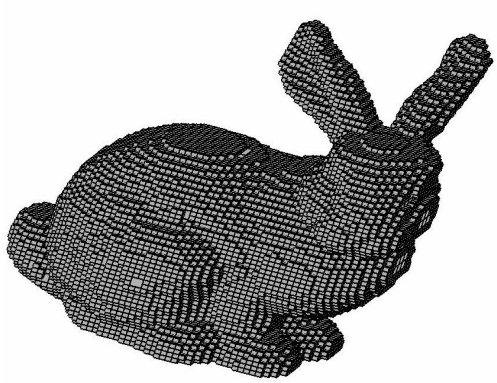


Figure 6: Approximation obtained after transmitting the first 23,895 blocks in the approximation of the bunny in Figure 5 which corresponds to OB(1).

One of the interesting properties of our approximation/compression method is that in two dimensions it has been observed [13] that it is biased in favor of objects with shapes such as "panhandles" rather than "staircases". In this case, the handle is apparent at the initial stage of the approximation which is not the case when either the inner or outer BLACK (IB or OB, respectively) approximation are used. These approximations only reveal the handle as we get deeper and deeper in the tree. The same phenomenon was observed in our three-dimensional example bunny object where we see the ears right away. In general, the less round or spherical is the object, the greater is our method's potential for yielding compression.

5 Concluding Remarks

We have presented techniques for the progressive approximation and compression of sets of three-dimensional binary objects represented by a pointerless region octree. Our approach was an adaptation of a method previously presented by Samet [13]. However, the proof that it exhibits compression is different and is more general so that it is applicable to objects of arbitrary dimensionality.

Directions for future work include the adaptation of these methods to objects of more than just two colors (i.e., different gray levels). This is quite difficult as the method makes strong use of the fact that the part of the object that is not in the foreground is in the background. Nevertheless, this is an important topic for research.

References

[1] P. J. Burt and E. H Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, April 1983.

[2] M. F. Deering. Geometry compression. In *Proceedings of the SIGGRAPH'95 Conference*, pages 13–20, Los Angeles, August 1995.

[3] I. Gargantini. An effective way to represent quadtrees. *Communications of the ACM*, 25(12):905–910, December 1982.

[4] W. E. Glenn, J. Marcinka, and R. Dhein. Simple scalable video compression using 3-D subband coding. In *Proceedings of the SMPTE Advanced Television and Electronic Imaging Conference*, pages 140–143, San Francisco, February 1995.

[5] G. M. Hunter. *Efficient computation and data structures for graphics*. PhD thesis, Department of Electrical Engineering and Computer Science, Princeton University, Princeton, NJ, 1978.

[6] L. Jones and S. S. Iyengar. Space and time efficient virtual quadtrees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):244–247, March 1984.

[7] K. Knowlton. Progressive transmission of grey-scale and binary pictures by simple efficient, and lossless encoding schemes. *Proceedings of the IEEE*, 68(7):885–896, July 1980.

[8] J. Li, J. Li, and C. C. J. Kuo. Progressive compression of 3d graphic models. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems'97*, pages 135–142, Ottawa, Canada, June 1997.

[9] D. Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19(2):129–147, June 1982.

[10] T. Naveen and J. W. Woods. Subband finite state scalar quantization. *IEEE Transactions on Image Processing*, 5(1):150–155, January 1996.

Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02)
0-7695-1521-5/02 $17.00 © 2002 IEEE

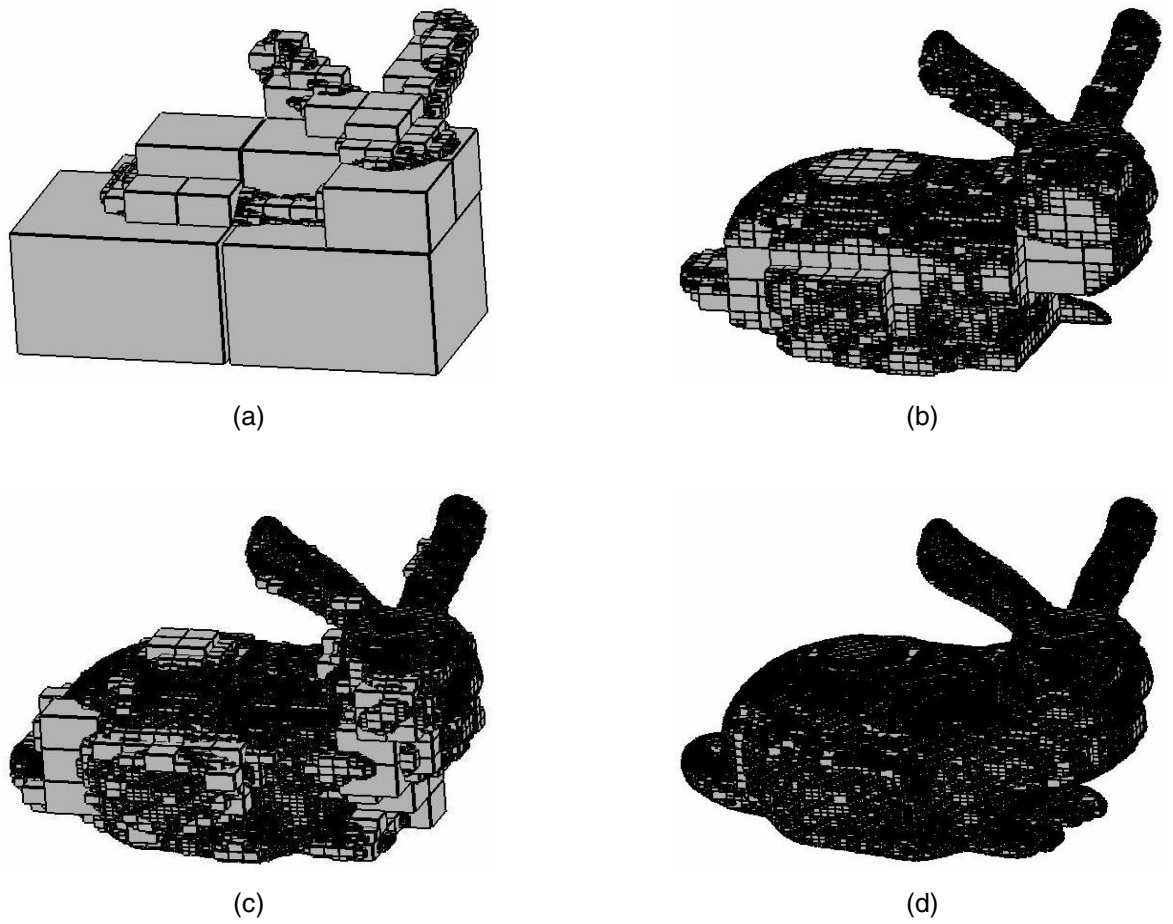


Figure 7: Approximations of the bunny in Figure 5: The result of applying (a) FBW(6), (b) FBW(5), (c) FBW(4), and (d) FBW(3).

[11] M. Rabbani and R. Joshi. An overview of the JPEG 2000 still image compression standard. *Signal Processing: Image Communication*, 17(1): 3–48, January 2002.

[12] S. Ranade, A. Rosenfeld, and H. Samet. Shape approximation using quadtrees. *Pattern Recognition*, 15(1):31–40, 1982.

[13] H. Samet. Data structures for quadtree approximation and compression. *Communications of the ACM*, 28(9):973–993, September 1985.

[14] H. Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley, Reading, MA, 1990.

[15] H. Samet and M. Tamminen. Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):579–586, July 1988.

[16] K. R. Sloan Jr. and S. L. Tanimoto. Progressive refinement of raster images. *IEEE Transactions on Computers*, 28(11):871–874, November 1979.

[17] M. Tamminen. Comment on quad- and octtrees. *Communications of the ACM*, 27(3):248–249, March 1984.

[18] S. L. Tanimoto and T. Pavlidis. A hierarchical data structure for picture processing. *Computer Graphics and Image Processing*, 4(2):104–119, June 1975.

[19] G. Taubin. 3d geometry compression and progressive transmission. In *Proceedings of the EUROGRAPHICS'99 Conference*, P. Brunet and R. Scopigno, eds., pages 81–96, Milan, Italy, September 1999.

Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02)
0-7695-1521-5/02 $17.00 © 2002 IEEE