The Hybrid Octree: Towards the Definition of a Multiresolution Hybrid Framework

Imma Boada[1] and Isabel Navazo[2]

[1] Institut Informàtica i Aplicacions, Universitat de Girona, Spain
imma@ima.udg.es
[2] Dep. LSI , Universitat Politècnica de Catalunya, Spain
isabel@lsi.upc.es

Abstract. The Hybrid Octree (HO) is an octree-based representation scheme for coding in a single model an exact representation of a surface and volume data. The HO is able to efficiently manipulate surface and volume data independently. Moreover, it facilitates the visualization and composition of surface and volume data using graphic hardware. The HO definition and its construction algorithm are provided. Some examples are presented and the goodness of the model is discussed.

1 Introduction

The visualization of scenes that integrate surface and volume data in a single image plays an important role in a large number of scientific visualization applications. In surgical planning, for example, the volume information captured from input devices needs to be visualized, manipulated and analyzed along with objects such as osteotomy surfaces, prosthetic devices or scalpels; in meteorology clouds have to be rendered over terrain data, etc,.

We define a *hybrid scene* as an scene composed of surface and volume data. The different techniques that have been proposed to deal with hybrid scenes are based on one of the two following approaches. The *data conversion* approach that reduces surface and volume data to a common codification scheme, by applying a voxelization process or a polygonalization strategy. Then volume and polygonal data can be rendered using a *classical* rendering pipeline [5, 6, 15, 18]. The *independent representation* approach that preserves original surface and volume data in their independent representation schemes. Data is rendered independently, by the application of two separate rendering processes, and data integration is part of the visualization process which requires the definition of specialized hybrid render algorithms able to composite surface and volume data in depth sorted order. Such an approach usually requires a costly process to properly composite the data [9, 17, 7, 19, 8].

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2330, pp. 121–130, 2002.

In this paper, we introduce a new approach to deal with hybrid scenes: the Hybrid Octree (HO). The HO is an octree-based data structure able to maintain surface and volume data simultaneously and implicitly ordered. The HO avoids conversion artifacts and loss of information of data conversion methods. The capabilities of the HO to represent and efficiently manipulate surface and volume data, either *independently*, taking advantage of classical surface and volume visualization approaches, or in an *integrated* manner, combining surface and volume data in a single image have been presented in detail in [1].

The paper has been structured as follows. In Section 2 we review our previous work related to surface and volume data octree based representations. In Section 3 a detailed description of the HO and its construction process are provided. The capabilities of the HO to manipulate surface and volume data are analyzed in Section 4. Finally, conclusions and future work are given in Section 5.

2 Octree based Codifications. Previous Work.

An octree, originally introduced for solid representations [11,16,14], is a tree that codes the recursive subdivision of a finite cubic universe. The root of the tree represents the universe, a cube with 2^n edge. This cube is divided into eight identical cubes, called octants, with an edge length of 2^{n-1}. Each octant is represented by one of the eight descendants of the root. If the information of an octant can not be represented in an exact way, it is labeled as a *Grey node*. Each Grey node is divided into another eight identical cubes which are represented as descendants of the octant in question. This process is repeated recursively until octants contain data that can be represented exactly (named Terminal Octree Nodes) or octants have a minimum edge length called resolution (Minimal Resolution Nodes). According to the data that we would like to represent by means of an octree, it is needed the definition of criteria to determine which information have to be stored in each node and when a node of the octree is a terminal node.

Our study has been restricted to volume data sampled in 3D regular grids and to fitted surfaces reconstructed from the volume data by the Discretized Marching Cubes (DiscMC) [12,13]. Therefore, the HO has to be designed to support these data in an integrated manner in its terminal nodes.

Previously to the description of the HO we are going to review the octree based structures that we have proposed to represent regular distributed volume data (the Volume Octree, (VO))[2] and a decimated DiscMC fitted surface (the Surface Octree (SO)) [3].

2.1 The Volume Octree

The VO [2] is an octree based codification used to code homogeneous regions of 3D regular sampled data in a compressed way. In the VO, a node is considered *homogeneous* if each sample data inside its associated octant can be approximated by a trilinear interpolation of the eight values on the corners of the octant. The maximum error which is introduced by this trilinear interpolation is the *nodal error*, denoted as ε_0. The nodal error is related to the degree of homogeneity in the area covered by the node.

The construction of the VO starts with the construction of an octree initially complete. Then, during a bottom-up octree traversal nodal errors are computed and all sibling nodes which can be merged in the parent node without introducing an approximation error are purged. Therefore, the VO terminal nodes are those for which their nodal error $\varepsilon_0 \leq \epsilon_u$ with ϵ_u the user required degree of accuracy. The VO allows to obtain multiresolution volume representations by applying error-driven adaptive traversals. These multiresolution representations are the basis of a texture memory assignation policy that reduces the space of texture memory required to obtain 3D texture-based volume data visualizations.

2.2 The Surface Octree

The SO [3] is an octree based codification used to maintain a decimated codification of a fitted surface without introducing error. In addition to the classical White, Black and Grey nodes (octants that are external, internal and crossed by the surface), the SO incorporates five new terminal surface nodes (TS nodes) (see Fig. 1).

To codify the surface in a TS node n_i we maintain the configuration of the node (i.e. F, E, DE, B^n or B^*) and the set of plane equations defining the surface. All the patterns of the Marching Cubes algorithm [10] can be codified as a TS node, therefore the SO is able to maintain any fitted surface obtained with the Marching Cubes. In case of using the DiscMC the plane codification is reduced to a set of integers.

The SO construction process starts with an octree initially complete in which maximal and minimal values covered by each node are stored [20]. The octree is traversed and the DiscMC is applied to the terminal nodes intersected by the surface to provide the surface codification. Afterwards, on a bottom-up octree traversal a merging process is applied to merge coplanar planes and codify them when is possible in upper SO nodes without introducing any simplification error. All details of this merging process can be found in [3]. The capabilities of the SO to support multiresolution surface reconstructions are described in [4].

3 The Hybrid Octree Construction

The HO can be interpreted as the integration of a VO and a SO. Its construction process is composed of two phases: the *Volume Data Integration* and the *Surface Data Integration* (we are going to consider DiscMC fitted surfaces).

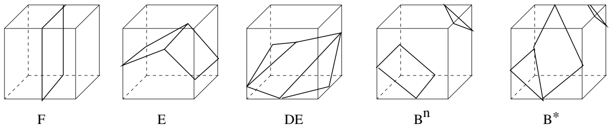

Fig. 1. Terminal Surface nodes: (F) Face node, a node crossed by a single planar face. (E) Edge node, a node crossed by two planar faces that converge into the node generating an edge. (DE) Double Edge node, a node crossed by three planar faces that converge into the node generating two edges. (B^n) Band^n node, a node crossed by n planar faces that never intersect into the node, with $2 \leq n \leq 4$. (B^*) Special Band node, a node obtained from the union of a F and an E node.

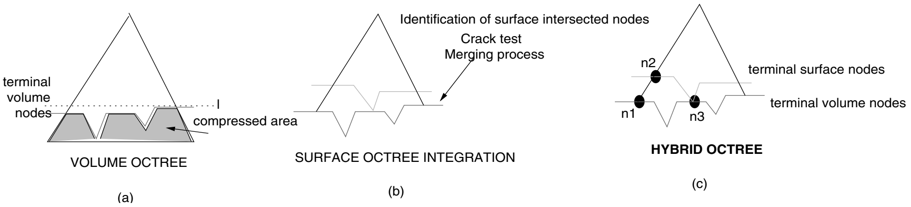

Fig. 2. (a) The Volume Octree is constructed. (b) The construction of the Surface Octree starts at terminal volume nodes. (c) The hybrid octree maintains volume and surface terminal nodes.

3.1 Volume Data Integration

The first phase of the HO construction process is concerned with volume data. This phase starts with the construction of an octree initially complete. Voxel values are not represented explicitly by the octree nodes, but implicitly (the extents of the corresponding region in the voxel space are directly computed from the node index).

For each node n_i of the HO we maintain:

- the maximum (max) and minimum (min) values of the region covered by node (to improve the surface extraction process required in the next phase [20]);
- a list of *nodal errors* $\varepsilon_0, .., \varepsilon_{l_{max}-k}$ where l_{max} is the depth of the octree and k the level of n_i (see Sect. 2.1). These *nodal errors* are computed during a bottom-up octree traversal in which all sibling nodes which can be merged in the parent node without introducing an approximation error are purged (i.e. nodes where $\epsilon_0 \leq \epsilon_u$ with ϵ_u the user allowed accuracy). The list of nodal errors is used to perform multiresolution volume visualizations.

This phase ends when terminal HO nodes with respect to the volume criteria have been detected. (see Fig. 2(a)). Note that these nodes are distributed at different levels of the HO.

3.2 Surface Data Integration

The second phase of the HO construction process codifies the surface in a set of HO nodes. The surface codification strategy is similar to the one used for the SO construction and is composed of the following steps:

(i) *Detect the volume HO terminal nodes intersected by the surface.* During a top-down HO traversal terminal nodes such that $max \leq isovalue \leq min$, where *isovalue* identifies the surface, are detected.

(ii) *Use the TS node configurations (i.e. F, E, DE, B^n or B^*) to codify the surface in the volume HO terminal nodes.* There is a correspondence between the DiscMC patterns and the TS configurations [1]. Therefore applying the DiscMC to the node the TS configuration is directly obtained.
Note that as a consequence of the pruning process that has been applied in the volume integration phase, the terminal HO nodes can be distributed at different levels of the octree. To codify the surface in a node we evaluate its neighbor nodes intersected by the surface: (a) if all evaluated nodes are in the same level no cracks can appear (see Fig. 3(a)); (b) conversely, if the nodes are from different levels cracks can appear (see Fig. 3(b)). In this case the solution we have adopted, that experimentally generates nice results, adapts the intersection points of the processed node to the point position of the deeper neighbor node. This is an straightforward process as the DiscMC allows to determine in advance all possible intersection point positions.

(iii) *Surface Compression.* Once the surface has been codified, a bottom-up octree traversal is performed and the merging process is applied to detect and represent coplanar facets in upper HO levels *without* introducing error. The resultant surface is codified in a set of TS nodes distributed at different upper levels of the HO (see Fig. 2(b)).

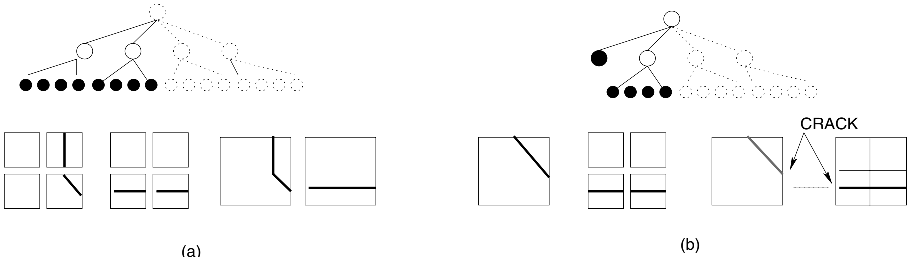

Fig. 3. Black circles identify intersected terminal volume nodes. (a) When all the nodes are at the same level the continuity of the surface is preserved. (b) When terminal volume nodes correspond to different octree levels, the application of the DiscMC leads holes on the final surface.

3.3 Terminal HO nodes

If we compare the VO, SO and HO, it can be seen that unlike VO and SO, the HO construction has applied two different criteria to identify terminal HO nodes. Consequently, we can identify terminal HO nodes with respect to the volume or the surface octree construction criterion (see Fig. 2(c)). In particular:

- HO nodes such that $\varepsilon_0(n_i) \leq \epsilon_u$ are terminal nodes with respect to the volume criterion (see node n1 of figure 2(c)).
- HO nodes intersected by the surface with a TS configuration are terminal nodes with respect to the surface criterion (see node n2 of Fig. 2(c)).
- HO nodes such that $\varepsilon_0(n_i) \leq \epsilon_u$, intersected by the surface and with a TS configuration are terminal nodes with respect to the volume and also with respect to the surface construction criteria (see node n3 of Fig. 2(c)).

4 Hybrid Octree Evaluation

The goal of the testing sessions has been proving the capabilities of the HO to manipulate surface and volume data either independently or in an integrated way. In the case that only surface or only volume data has to be explored the use of the HO has not much sense. In these situations is more appropriate the use of the VO or the SO. However, although we are conscious of this fact, we have considered opportune, in the first tests, to evaluate the suitableness of the HO to perform only volume or only surface data explorations.

4.1 HO volume data exploration

As it has been previously described the first phase of the HO construction process is equivalent to the VO construction. The only difference is the information that has to be stored in the nodes: while in the VO we maintain the nodal error, in the HO we maintain the nodal error, max and min values and the information related to the surface (which is null in the case that only volume data is codified).

To visualize the volume data represented in the HO we apply the 3D texture-based octree volume visualization algorithm with the importance-driven texture memory assignation strategy proposed in [2]. Given some user-defined constraints, the HO octree is traversed at rendering time and a set of nodes is selected. This set of nodes determines the current brick decomposition to be used to render the volume data and how the volume data has to be represented in texture memory. The experimental results achieved with the HO volume visualization are equivalent to the ones achieved with the VO [2] in quality and efficiency.

4.2 HO surface data exploration

In the case of surface data explorations the comparison of the SO and the HO is based on the evaluation of the surface codification and the surface reconstruction processes. The models used for the tests are: a CT-vertebra of 128x128x80 represented as (1) in Table 1; a CT-head of 96^3 represented as (2) and a CT-scanned jaw of $128x128x40$ represented as (3). The tests have been performed on a Pentium III at 450MHz with 512Kb of cache memory and a NVidia graphics card.

The results reported in Table 1 compares the distribution of TS nodes before and after the application of the merging process in case of using the HO or the SO presented in [3]. The small difference in the number of nodes of each configuration is due to that in the HO the surface codification process is done in terminal volume HO nodes that not necessarily are of minimal resolution (homogeneous regions have been compacted in the Volume Integration phase described in Sect. 3.1). Note that the HO and the SO achieve the same degree of decimation after the application of the merging process. The HO requires more time to codify the surface. This increase is consequence of the crack test that has to be applied to codify the surface with no error. Memory requirements are comparable with the SO ones.

Table 1. Columns (2) to (6) represent the number of TS nodes according its configuration before the application of the merging process; columns (7) to (11) TS nodes after the application of the merging process; column (12) simplification rate; column (13) times (in seconds) including the codification of the surface in the terminal intersected nodes and the application of the merging process.

Model	F	E	DE	B^n	B^*	F	E	DE	B^n	B^*	Dec. rate	time
(1)(HO)	60.124	5.215	59	57	0	30.095	5.094	64	70	0	44%	7,9
(1)(SO)	62.250	5.191	59	57	0	30.890	5.079	64	70	0	44%	7,1
(2)(HO)	47.654	13.075	556	2.306	200	34.127	13.013	570	2.554	214	15%	4,7
(2)(SO)	48.927	12.975	556	2.326	200	34.336	13.413	570	2.554	214	15%	3,8
(3)(HO)	95.892	22.367	1.122	2.432	196	37.641	25.832	1.142	2.462	210	30%	7,4
(3)(SO)	99.902	25.522	1.122	2.432	196	38.824	25.888	1.142	2.462	210	30%	6,9

To evaluate the capabilities of the HO to support multiresolution surface reconstructions we have applied the importance driven multiresolution algorithm presented in [4]. This algorithm starts with the definition of the region of interest (ROI) by a simple 3D subregion of the dataset domain. Then the HO is traversed and terminal surface nodes contained in the ROI are detected. The surface of these nodes is reconstructed with maximal accuracy. Then, it is performed a process to reconstruct, in a simplified manner, the outer ROI intersected nodes.

Figure 4 presents visual results related to multiresolution surface reconstructions of a CT-head model codified in the HO. When changing the ROI previous computations can be used, i.e. not all the HO nodes have to be recalculated.

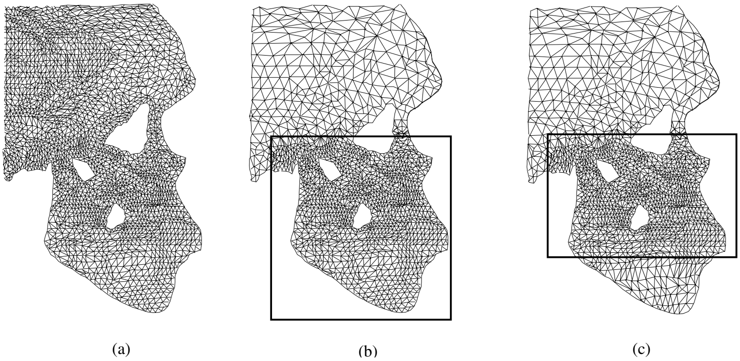

Fig. 4. (a) Lateral view of the CT-skull at full resolution composed by 69.899 planes. (b) Simplified representation composed of 48.108 polygons. The simplification rate if compared with (a) is 30% with 35% of TS nodes into the ROI. (c) Simplified representation composed of 32.213 polygons. The simplification rate if compared with (a) is 53% with 20% of TS nodes into the ROI.

4.3 HO volume and surface data exploration

The most attractive feature of the HO is its capability to maintain surface and volume data simultaneously codified in its nodes. The main advantage of such a capability is that all the sorting phase required by any hybrid renderer is implicitly made when the HO is built, thus during the hybrid visualization process the sorting information can be obtained directly from the HO.

To perform multiresolution hybrid visualizations we have defined a 3D texture-based hybrid visualization function [1]. This function takes advantage of the multiresolution texture memory assignation policy presented in [2], using a compressed texture representation of homogeneous and no importance areas of the volume. The surface is located between the textured polygons according to the position of the TS node inside the HO nodes.

The renderings of figure 5 are obtained from the CT-head dataset. In these images the rendered surface polygons correspond to the dental region. The volume data has been rendered by adopting different voxel-to-texel ratio. At each step the required texture memory is reduced (8Mb in (a), 4 Mb in (b) and 2Mb in (c)). This reduction improves rendering speed.

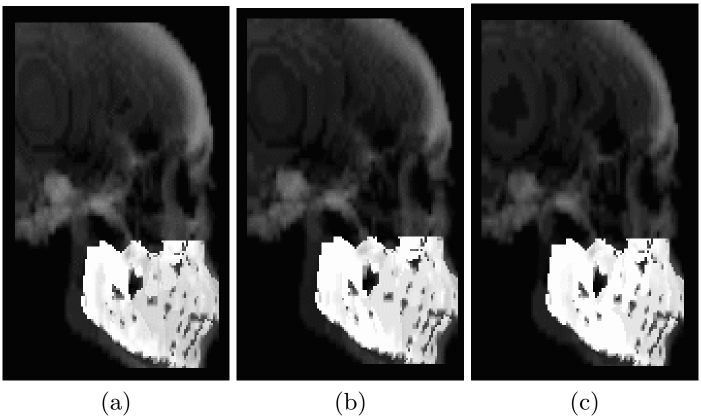

Fig. 5. Different renderings obtained on the CT-head. Surface polygons represent the teeth region.

5 Conclusions and Future Work

We have presented the Hybrid Octree, an extension of the classical octree able to support volume and surface data codified in its nodes and therefore, able to maintain all the data defining an hybrid scene. The main properties of the HO are that: (i) it maintains original surface and volume information. Thus, it avoids conversion artifacts and loss of information of data conversion methods; (ii) it maintains order information implicitly represented in its nodes. Therefore, it reduces the complexity of the depth sorting process required by hybrid renderers [1]; (iii) it preserves all the capabilities of the SO and the VO, therefore it is a well-suited framework for supporting multiresolution surface and volume data explorations.
Our future work is centered on the codification of any fitted surface (not only DiscMC surfaces) by modifying the number of discrete planes used to represent the surface. This modification will produce more compact and smooth surfaces and probably with less error.

Acknowledgements: This work was partially financed by the project 2FD97-1511 of the Spanish Government.

References

1. I. Boada, Towards Multiresolution Integrated Surface and Volume Data Representations. PhD Thesis, Universitat Politècnica de Catalunya, September (2001).
2. I. Boada, I. Navazo, and R. Scopigno, Multiresolution Volume Visualization with a Texture-based Octree. The Visual Computer, Springer International, 17 (3), 185-197, (2001).

3. I. Boada and I. Navazo, An Octree Isosurface Codification based on Discrete Planes. Proceedings IEEE Spring Conference on Computer Graphics (2001) 130-137, Budmerice,Slovakia.
4. I. Boada and I. Navazo, Multiresolution Isosurface Fitting on a Surface Octree. Vision, Modeling and Visualization 2001 proceedings, 318-324, Stuttgart, November(2001).
5. A. Kaufman, Efficient Algorithms for 3D Scan-Conversion of parametric Curves, Surfaces and Volumes, Computer Graphics, 21, 4, 171-79, July (1987).
6. A. Kaufman, Efficient Algorithms for 3D Scan-Converting Polygons, Computer and Graphics, 12, 2, 213-219,(1988).
7. A. Kaufman, R. Yagel and D. Cohen, Intermixing Surface and Volume Rendering. 3D Imaging in Medicine, Edited by K.H. Hohne et al. Springer Verlag Berlin Heidelberg (1990).
8. K. Kreeger and A. Kaufman, Mixing translucent Polygons with Volumes. IEEE Visualization (1999).
9. M. Levoy, A Hybrid Ray Tracer for Rendering Polygon and Volume Data. IEEE Computer Graphics and Applications, 10, 33-40, March (1990).
10. W.Lorensen and H.Cline, Marching cubes a high resolution 3D surface construction algorithm,. ACM Computer Graphics (Proceedings of SIGGRAPH '87), vol.21, n 4, 163-170, (1987).
11. D.Meagher, Geometric modeling using octree encoding. Computer Graphics and Image Processing, 19(2):129-147,(1982).
12. C.Montani, R.Scateni and R.Scopigno, Discretized Marching Cubes, in Visualization '94 Proceedings, R.D. Bergeron and A.E.Kaufman, Eds. (1994), 281-287, IEEE Computer Society Press.
13. C.Montani, R.Scateni and R.Scopigno, Decreasing Isosurface Complexity via Discrete Fitting, Computer Aided Geometric Design, 17 (2000) 207-232.
14. I.Navazo, Extended Octree Representation of General Solids with Plane Faces: Model Structure and Algorithms. Computer and Graphics, vol 13, 1, (1989), 5-16.
15. B.A. Payne and A.W. Toga, Distance Field manipulation of surface models. IEEE Computer Graphics and Applications, 12(1), 65-71. January (1992).
16. H.Samet, Applications of Spatial Data Structures. Addison Wesley, Reading, MA, (1990).
17. L.M.Sobierajski and A. Kaufman, Volumetric Ray-tracing. In Proceedings of 1994 Symposium on Volume Visualization, 11-18. ACM Press, October (1994).
18. M. Sramek, Non-binary Voxelization for Volume Graphics. In Proceedings of Spring Conference on Computer Graphics, (2001) 35-51.
19. D.Tost, A.Puig and I.Navazo, Visualization of mixed scenes based on volume and surface. In Proceedings of the Fourth Eurographics Workshop on Rendering, 281-294, (1993).
20. J. Wilhems, A. Van Gelder. Octrees for Faster Isosurface generation. ACM Transactions on Graphics, 11(3):201-297, July (1992).