# Computing on Rays: A Parallel Approach for Surface Mesh Modeling from Multi-Material Volumetric Data

Charlie C.L. Wang*

Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, China

## Abstract

Ray representation (Ray-rep) of a solid has been studied and used in the solid modeling community for many years because of its compactness and simplicity. This paper presents a parallel approach for mesh surface modeling from multi-material volume data using an extended Ray-rep as an intermediate, where every homogeneous region is enclosed by a set of two-manifold surface meshes on the resultant model. The approach consists of three major algorithms: firstly, an algorithm is developed to convert the given multi-material volumetric data into a Ray-rep for heterogeneous solid; secondly, filtering algorithm is exploited to process the rays of heterogeneous solid in parallel; and lastly, the adaptive mesh surfaces are generated from the ray-rep through a dual-contouring like algorithm. Here the intermediate surfaces between two constituent materials can be directly extracted without building the volumetric mesh, and the manifold topology is preserved on each surface patch. Furthermore, general offset surface can be easily computed in this paradigm by designing a special parallel operator for the rays.

***Keywords***: Surface mesh, implicit representation, heterogeneous models, parallel algorithm.

## 1 Introduction

The methods for computing the interface (in the representation mesh surface) between the homogeneous material regions from a multi-material volume data, which is obtained by CT [1, 2] or MRI [3], have been studied for many years. The main purpose for computing these mesh surfaces bounding homogeneous material regions is to ease the downstream applications of simulation using *finite element method* (FEM) or *boundary element method* (BEM). Although the meshfree analysis and simulation methods [4, 5, 6] can be employed, it is still important to have the surface representation in many computational engineering applications. After getting well-defined surface meshes bounding homogenous regions, the procedure to generate volumetric meshes for each region is standard (see [7, 8] and the references therein). The generated mesh surfaces bounding homogeneous material regions have many applications in both the mechanical engineering and the biomedical engineering.

**Problem Statement**: Suppose a heterogeneous object $H$ in $\Re^3$ is defined by a function $F(\mathbf{p})$ with $\mathbf{p} \in \Psi \subset \Re^3$ (the domain $\Psi$ is given by an input multi-material volumetric data), where the value of $F(\mathbf{p})$ is an integral index of material class between 0 and $n$ – the total number of material types in $H$. $F(\mathbf{p}) = 0$ means $H$ does not occupy $\mathbf{p}$ (i.e., no material). If a region filled with the $i$-th type of material is denoted by $\Omega_i$, we then have $H = \cup_{i=0}^{n} \Omega_i$, $\Omega_i \cap \Omega_j = \Gamma_{ij}$ $(i \neq j)$ and $\Gamma_i = \partial\Omega_i \cap H$ where each homogeneous material region has a meaningful boundary surface $\Gamma_i$ and the interface between $\Omega_i$ and $\Omega_j$ is denoted by $\Gamma_{i,j} = \Gamma_{j,i} = \Gamma_i \cap \Gamma_j$ $(\forall i \neq j)$. Therefore, all the intermediate surface in $H$ can be defined as

$$\Gamma(H) = \cup_{i,j=0}^{n} (1 - \delta_{i,j}) \Gamma_{i,j} \tag{1}$$

with $\delta_{i,j}$ being the Kronecker delta defined to be *one* for $i = j$ and *zero* otherwise. To ease downstream applications like BEM (or FEM), we need to construct adaptive and quality mesh approximation of $\Gamma(H)$, where each intermediate surface $\Gamma_{i,j}$ is represented by a two-manifold mesh patch and the two linked patches should have consistent boundaries (i.e., their corresponding nodes are coincident).

The problem to be solved here is more or less similar to our previous work in [9]; however, we develop a parallel modeling approach here which can generate the results much faster – the computation time is reduced from tens of minutes to several seconds. Moreover, unlike [9] that stored the volumetric distance-field to each homogeneous region, we employ a set of Ray-reps to encode the information for all material regions where the memory requested by Ray-rep models is in the order of surface area but not the volume – it consumes less memory. Therefore, data sets with higher resolution can be processed. Figure 1 gives an illustration of our approach. A multi-material volumetric data is originally given in Fig.1(a) with different colors representing different materials, and the generated Ray-rep is visualized by surfels at the endpoints of rays in Fig.1(b). Figure 1(c) shows the surface meshes extracted by our approach. It

*E-mail: cwang@mae.cuhk.edu.hk

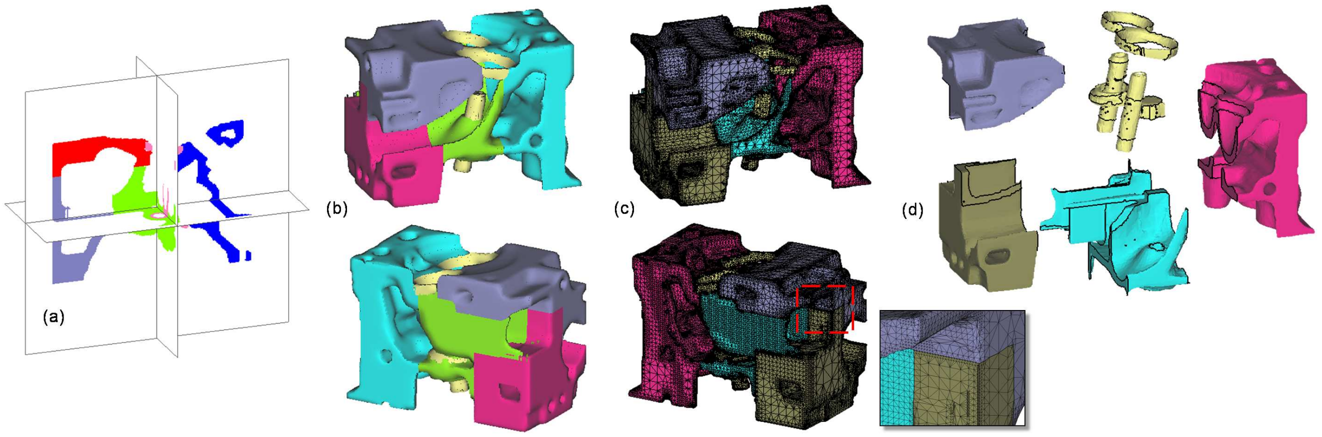


Figure 1: The major steps of our approach: (a) the originally given multi-material volumetric data $H$, (b) the generated Ray-rep of heterogeneous solid – visualized by the surfels at the samples of rays, (c) the reconstructed $\Gamma(H)$ as an assembly of two-manifold mesh surfaces, and (d) the reconstructed five homogeneous material regions bounded by surface meshes.

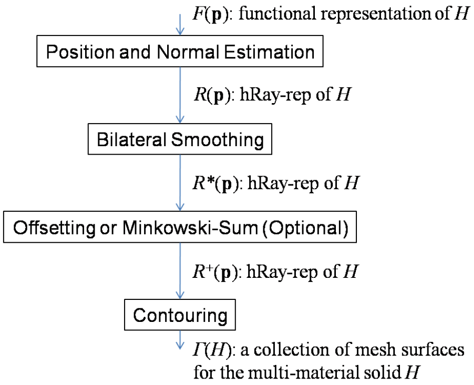
is easy to find that the meshes are compatible at boundaries and the sharp features are well-preserved. In other words, the nodes on the boundary of two linked patches are coincident, and sharp edges are generated on the result mesh surfaces (see the zoom-view of Fig.1(c)).

### 1.1 Related Work

The research of solid modeling for heterogeneous objects has been studied for many years and can be classified into the forward modeling strategy and the reversed modeling strategy, where the forward modeling strategy generates heterogenous objects according to simple input of design parameters and the reversed modeling reconstructs the heterogeneous objects from volumetric data sets. Unlike our approach that constructs the heterogeneous objects in a reversed modeling approach, most of the existing methods consider the forward modeling strategy (ref. [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]). A detail review on the modeling of heterogeneous object in the last decade was given in [19]. Hua et al. described a multiresolution heterogeneous solid modeling framework using simplex splines in [20]. Although a fitting algorithm was given, it is not clear how to apply their method to the multi-material volumetric data. Besides, they did not consider the reconstruction of sharp features either. The output model of our approach is a multi-volume model, which is in fact the one presented in [17]. This multi-volume model can be easily supported by current commercial CAD/CAM systems if such a heterogeneous model is represented as the assembly of several parts, where each part presents a volume with distinct material. The difficulty is how to construct the exactly same intermediate surfaces on two neighboring volumes, which is going to be solved by our research. The most recent researches that generate mesh surfaces from volumetric data are [1] and [2], where [1] concentrates on how to precisely classifying the volumetric model into component regions with homogeneous materials and [2] describes the strategy of modeling multi-component structure with the help of distance fields. None of them used the power of parallel computing which is nowadays available on consumer PCs.

There are dozens of algorithms in literature trying to generate two-manifold polygonal mesh surfaces from a volumetrically represented solid model. The Marching Cubes (MC) algorithm [21] is the first approach in the literature to generate a polygonal mesh surface from an implicit surface. As a variant of the original MC algorithm, the authors of [22, 23] modified the MC algorithm for single material to M3C, a multi-material marching cubes algorithm. However, as mentioned in [22], there are $8^8$ (i.e., 16,777,216) possible cases when the eight cube-vertices have eight different materials, which makes the task of making topology correct very tough and tedious. Compared with [22], the dual contouring approach conducted in [24] is simpler. Computing the dual contouring on the finest resolution of volumetric data will generate too many polygons like the M3C algorithm, thus it is impractical. Therefore, an octree structure is needed to make the dual contouring adaptive to the surfaces. In this paper, we will employ a dual-contouring like method to first construct an octree from the Ray-rep solid in parallel, and then generate the polygon soup that will be separated into a set of assembled two-manifold mesh surfaces finally.

The original ray representation (Ray-rep) of a solid was presented in [25] by Ellis et al. The researches followed include [26, 27, 28, 29]. Menon and Voelcker sampled the solid models into parallel rays tagged with h-tag (i.e., the information of half-space at the endpoints of rays) in [27] so that the completeness of Ray-rep can be generated. Benouamer and Michelucci employed triple Ray-rep to sample a solid in three orthogonal directions so that the solid modeling operations can be com-



$F(\mathbf{p})$: functional representation of $H$
Position and Normal Estimation
$R(\mathbf{p})$: hRay-rep of $H$
Bilateral Smoothing
$R^*(\mathbf{p})$: hRay-rep of $H$
Offsetting or Minkowski-Sum (Optional)
$R^+(\mathbf{p})$: hRay-rep of $H$
Contouring
$\Gamma(H)$: a collection of mesh surfaces for the multi-material solid $H$

Figure 2: Flow-chart of hRay-rep based mesh generation algorithm.

puted more robustly. As mentioned in [29], Ray-rep can make problem easy in the applications involving offsets, sweeps, and Minkowski operations. Stimulated by the work of [29], we will employ enriched triple Ray-rep as an intermediate to generate surface meshes from multi-material volume data. Another line of research related to our work is the so-called *Marching Intersections* (MI) approach [30, 31, 32]. However, none of the above approaches considered the modeling with multi-material objects, and it is not easy to enable their algorithms to take advantage of the parallel computing. To the best of our knowledge, this is the first parallel approach to process heterogenous solids in Ray-rep.

## 1.2 Main Result

Main results of the work presented in this paper include:

- A parallel approach is exploited to model mesh surfaces from multi-material volume data using Ray-rep as an intermediate, where every homogeneous region is enclosed by a set of two-manifold surface meshes on the resultant model.
- An extended Ray-rep, the *hRay-rep*, for heterogeneous solids is presented in this paper, which can store the volume data more compactly.
- Several parallel filters/operators are developed to process the samples on the rays of a solid in *hRay-rep*, which include not only the filters that keep the topology of model unchanged (like the smoothing filters) but also the operators that modify the topology (e.g., the offsetting and even the Minkowski sum).

The flow-chart of our mesh generation algorithm is given in Fig.2.

The hRay-rep will be introduced in Section 2. The parallel construction of a hRay-rep solid from multi-material volume data will be presented in Section 3 together with the parallel smoothing filters. After briefing the contouring method of a hRay-rep solid in Section 4, the parallel algorithm to compute general offset will be exploited in Section 5.

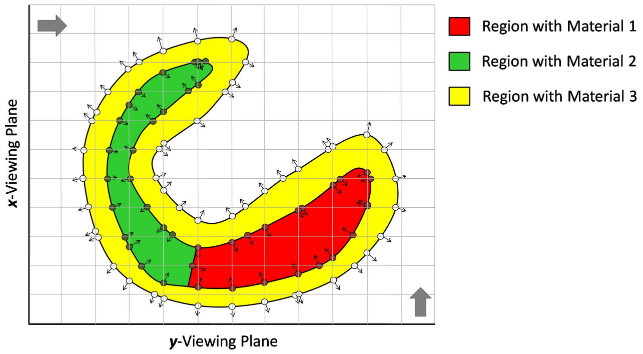


Figure 3: 2D illustration of hRay-rep – the extended Ray-rep for heterogeneous solids, where regions with different materials are presented in different colors. The depth of a sample on the intermediate surface is its distance to the viewing plane. The orientation of normal vector at the sample is specified by Definition 4.

# 2 Ray-rep for Heterogeneous Solids

In this section, we will define an extended Ray-rep, *hRay-rep*, which is used to describe the heterogeneous solids in our approach.

**Definition 1** Having a heterogeneous solid $H$ defined by a function $F(\mathbf{p})$ with the value of $F(\mathbf{p})$ being an integral index of material class between 0 and $n$, $\Gamma(H)$ defined in Eq.(1) gives the intermediate surfaces separating the homogeneous regions in $H$.

Figure 3 gives a 2D illustration of a heterogeneous solid $H$ and the homogeneous regions are illustrated in different colors.

**Definition 2** Without loss of generality, the hRay-rep of a heterogeneous solid $H$ along a specified direction $\mathbf{e}$, denoted as $R(\mathbf{e}, H)$, can be considered as a two-dimensional image with $w \times w$ rays, where each ray passes through the center of a pixel along the viewing direction $\mathbf{e}$.

**Definition 3** The samples on a ray $r^{\mathbf{e}}_{i,j}$ $(i, j = 1, \ldots, w)$ of $R(\mathbf{e}, H)$ are the intersections between the ray and the intermediate surface $\Gamma(H)$. The depth attribute at a sample gives the depth from the intersection to the viewing plane, and all samples on a ray are sorted by their depths in ascending order.

**Definition 4** The normal attribute is defined at each sample to specify the normal vector of the intermediate surface $\Gamma(H)$ at the sample; to maintain a consistent orientation, the normal vectors on the intermediate surface

$\Gamma_{i,j}$ always point from the region $\Omega_i$ to $\Omega_j$ if $i > j$.

**Definition 5** The material identification number, $ID$, is also defined at each sample on the rays of $R(\mathbf{e}, H)$, where $ID = i(n+1)+j$ if the ray is entering the homogeneous material region $\Omega_j$ from $\Omega_i$ at this sample, and $n$ is the number of different materials in $H$.

By Definition 5, if a point $\mathbf{p}_r$ on the ray is given, we can easily detect which region $\mathbf{p}_r$ belongs to by checking the $ID$s of two neighboring samples on the ray. Also, if the value of $ID$ at a sample is given, we can easily know that the sample is on the intermediate surface $\Gamma_{k,l}$ by $k = \llcorner ID/(n+1) \lrcorner$ and $l = ID - k(n+1)$.
Sampling a given model with $R(\mathbf{e}, H)$ along a single direction $\mathbf{e}$ will miss the surface regions that are nearly perpendicular to $\mathbf{e}$ (e.g., the bottom region of the surface shown in Fig.3 will be missed by the horizontal rays). As mentioned in [28], this miss-sampling can be improved by conducting another sampling along the direction perpendicular to $\mathbf{e}$.

**Definition 6** The complete hRay-rep $R(H)$ of a heterogeneous solid $H$ consists of $R(\mathbf{x}, H)$, $R(\mathbf{y}, H)$ and $R(\mathbf{z}, H)$ which have the same resolution and are sampled along $\mathbf{x}-$, $\mathbf{y}-$ and $\mathbf{z}-$axis respectively; the rays are arranged so that their intersections form $w \times w \times w$ nodes of uniform grids.

Figure 3 gives the 2D illustration of a complete hRay-rep. The sizes of information stored on a ray range from $O(1)$ to $O(k)$ where $k$ is the maximal number of layers of $\Gamma(H)$ on this ray. On most practical models, $k$ is a constant number that satisfies $k \ll w$; in the worst case, $k \to w$ on all rays, the upper bound of hRay-rep's memory complexity, $O(w^3)$, is reached. Therefore, we have the following proposition.

**Proposition 1** The memory complexity of hRay-rep is $O(w^2)$ on most practical models, and with $O(w^3)$ in the worst case.

In short, the memory cost of a hRay-rep is much less than a voxel-based representation of multi-material volume data, which is with $O(w^3)$ memory complexity. We will describe in the following section that the information from the multi-material volume data can be completely retained when converting it into a complete hRay-rep.

**Definition 7** The point set $S$ sampled from the intermediate surfaces $\Gamma(H)$ of a heterogeneous solid $H$ is defined as a $d$-covering of $H$ where any point $\mathbf{p}$ on $\Gamma(H)$ can find a point $\mathbf{q}$ that $\|\mathbf{p} - \mathbf{q}\| \leq d$.

**Proposition 2** The intermediate surfaces $\Gamma(H)$ sampled into a complete hRay-rep gives a $d$-covering of $H$ with $d$ bounded by $\sqrt{3}r$, where $r$ is the sampling distance between horizontal (or vertical) rays in a $R(\mathbf{e}, H)$.

*Proof.* See Appendix.

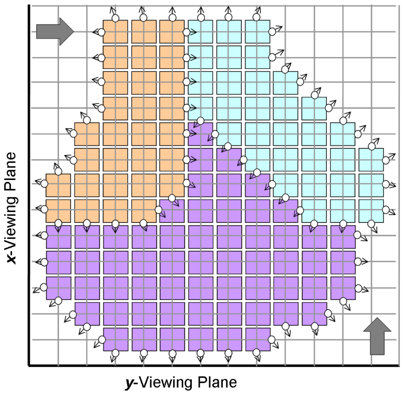


Figure 4: Converting multi-material volume data into a hRay-rep solid, where the boxes with different colors represent the voxels with different materials.

## 3 From Multi-Material Volume Data to Ray-rep

A multi-material volume model $H$ in the discrete domain, $\overline{H}$, is a set of integral vectors in $Z^3$. Elements $s_{i,j,k} = (i,j,k) \in \overline{H} \subset Z^3$ are called voxels and are thought of as unit cubes centered at $(i,j,k)$. As $\overline{H}$ is a multi-material model (i.e., a heterogeneous solid), each voxel also contains an integral number to specify its type of material. Therefore, we use $s^h_{i,j,k}$ to denote a voxel at $(i,j,k)$ with material index $h$. In this section, we will introduce a parallel algorithm to convert $\overline{H}$ into a hRay-rey, by which the meshes approximating the intermediate surfaces $\Gamma(H)$ can be efficiently extracted by the method in section 4. In order to recover the shape-edges on $H$ which are damaged by the uniform sampling in $Z^3$, two bilateral smoothing filters are introduced here as well.

### 3.1 Position and Normal Estimation

Without loss of generality, we can lay out a complete hRay-rep $R(H)$ to let its rays in all three directions pass though and intersect at the center of voxels in $\overline{H}$ (as shown in Fig.4), and the origin of the complete hRay-rep $R(H)$ (i.e., the intersection of $r^{\mathbf{x}}_{0,0}$, $r^{\mathbf{y}}_{0,0}$ and $r^{\mathbf{z}}_{0,0}$) is located at $(0,0,0) \in Z^3$. The samples on the hRay-rep $R(\mathbf{x}, H)$, $R(\mathbf{y}, H)$ and $R(\mathbf{z}, H)$ can then be obtained by the remarks below.

**Remark 1** A sample is added on the ray $r^{\mathbf{z}}_{i,j}$ if there are two neighboring voxels: $s^h_{i,j,k}$ and $s^m_{i,j,k+1}$ with $h \neq m$; $(k+\frac{1}{2})r$ is assigned as the depth of sample with $r$ being the width of a voxel.

Note that, a voxel $s^h_{i,j,k} \in (Z^3 \setminus \overline{H})$ can be considered as a voxel $s^h_{i,j,k} \in \overline{H}$ with $h = 0$. Similarly, the samples on $r^{\mathbf{x}}_{j,k}$ can be generated by checking $s^h_{i,j,k}$ and $s^m_{i+1,j,k}$, and checking $s^h_{i,j,k}$ and $s^m_{i,j+1,k}$ can obtain the samples on $r^{\mathbf{y}}_{k,i}$.

**Remark 2** The value of $ID$ at a sample on the intermediate surface of two different material regions is encoded by Definition 5.

**Remark 3** The normal vector at a sample on the intermediate surface $\Gamma_{i,j\ (i<j)}$ are computed by the *Principal Component Analysis* (PCA) of its neighboring samples (ref. [33]), where only the samples with the same value of $ID$ are considered and the orientation of normals is adjusted according to Definition 4.

**Details of Parallel Implementation** The operations in Remark 1 and 2 can be easily parallelized to run in multi-threads where each thread only considers the samples on its corresponding ray. For Remark 3, the neighboring samples of a sample on the voxel $s^h_{i,j,k}$ with distance less than $r\flat$ ($\flat \in Z$) will be searched on check all the possible samples among the voxels $s_{i',j',k'}$ with $i' \in [-(\flat+1)+i, (\flat+1)+i]$, $j' \in [-(\flat+1)+j, (\flat+1)+j]$ and $k' \in [-(\flat+1)+k, (\flat+1)+k]$. All tests in this paper choose $\flat = 3$ as the search range. This neighborhood search can also be implemented in parallel on mutex exclusion threads, where concurrent writing of the same variable by multiple threads does not happen. Moreover, if the set of volume data is very huge, it is easy to implement the above conversion algorithm in an out-of-core streaming mode to generate the complete hRay-rep.

## 3.2 Bilateral Smoothing Filters

When directly contouring the hRay-rep generated by Remarks 1-3 to generate mesh surfaces, the staircase artifacts can be found on the resultant surfaces (see Fig.5). To solve this problem, two feature-preserved bilateral filters similar to [34, 35, 36] are introduced to smooth the samples on a complete hRay-rep $R(H)$.

**Normal Filter** For a sample $s$ with the surface normal vector $\mathbf{n}_s$ and the position $\mathbf{p}_s$, the filtered normal $\overline{\mathbf{n}}_s$ is computed by

$$\overline{\mathbf{n}}_s = \frac{1}{k(s)} \sum_{q \in N(s)} W_c(\|\mathbf{p}_s - \mathbf{p}_q\|) W_s(I(s,q)) \mathbf{n}_q \tag{2}$$

where

$$k(s) = \sum_{q \in N(s)} W_c(\|\mathbf{p}_s - \mathbf{p}_q\|) W_s(I(s,q)), \tag{3}$$

and $N(s)$ is the neighboring samples of $s$ with the same $ID$

$$N(s) = \{q : \|\mathbf{p}_s - \mathbf{p}_q\| < 2\sigma_c \ (\forall ID(s) = ID(q))\}. \tag{4}$$

$$W_c(t) = e^{-t^2/2\sigma_c^2} \tag{5}$$

is the standard Gaussian filter with parameter $\sigma_c = 1.5r$, and

$$W_s(t) = e^{-t^2/2\sigma_s^2} \tag{6}$$

is a similarity weight function for feature-preserving with parameter $\sigma_s = 0.15$ that penalizes large variation in normals. $I(s,q)$ known as an intensity function defines the projection of the normal difference on the normal $\mathbf{n}_s$ as

$$I(s,q) = \mathbf{n}_s \cdot (\mathbf{n}_s - \mathbf{n}_q). \tag{7}$$

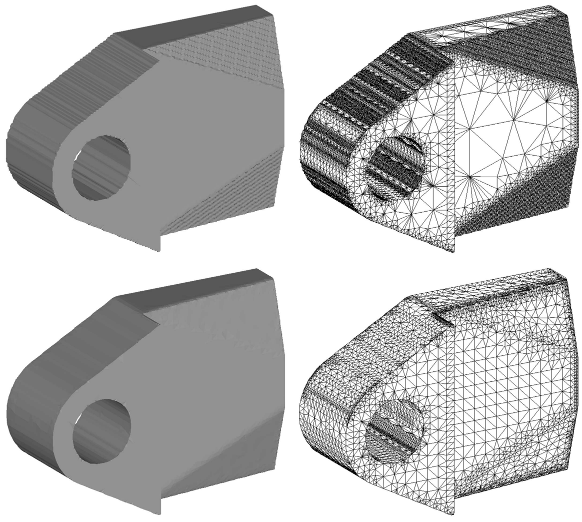


Figure 5: Function of smoothing filters: (top row) the staircase artifacts can be found on the slope surface generated from the samples produced by Remarks 1-3, and (bottom row) the staircase artifacts are eliminated by applying the normal and depth filters while the sharp features are preserved.

**Depth Filter** The bilateral filter on the depth $d_s$ of a sample $s$ on the ray in $\mathbf{e}$ direction is defined in a similar way. The update of depth value $d_s$ at the sample $s$ is computed by

$$\delta_{d_s} = \frac{1}{k(s)} \sum_{q \in N(s)} W_c(\|\mathbf{p}_s - \mathbf{p}_q\|) W_s(I(s,q)) \delta_q \tag{8}$$

where

$$\delta_q = (\mathbf{p}_q - \mathbf{p}_s) \cdot \mathbf{e} \tag{9}$$

gives the distance from $s$ to the projection of $q$ on the ray $\mathbf{e}$ holding the sample $s$.

**Remark 4** We let $d_s^{new} = d_s + \delta_{d_s}$ if $|\delta_{d_s}| < r/2$; otherwise, we truncate the depth update by $\delta_{d_s} = \frac{1}{2} r \delta_{d_s} / |\delta_{d_s}|$.

**Proposition 3** Applying the bilateral depth filter on the complete hRay-rep $R(H)$ will not break the consistency of inside/outside status at the intersection points of rays by the truncation defined in Remark 4.

*Proof.* The position of a sample determined by Remarks 1-3 is located on the boundary between two neighboring voxels; thus, the movement defined in Remark 4 will avoid it from moving beyond the center of the two voxels. As the rays of $R(H)$ intersect at the center of voxels, the update of depths on samples by Eq.(8) and Remark

4 will not change the inside/outsite status at the intersection points of rays. Therefore, the inconsistent cases on triple ray-reps discussed in [28] will not occur here. ⋄

In practice, we usually apply normal filter to the hRay-rep solid $R(H)$ first, and then followed by a run of depth filtering and another run of normal filtering.

**Details of Parallel Implementation** Since both these filters adopt the same support size $N(s)$ defined in Eq.(4), the computation can also be parallelized to run in multi-threads where each thread only considers the update of samples on one ray. For a sample $s$ on $r^{\mathbf{e}}(i,j)$ with depth value $d_s$, the samples on the following rays are detected to see if they are in $N(s)$:

- $r^{\mathbf{e}}(i',j')$ with both $(i'-i)$ and $(j'-j) \in [-\triangle, +\triangle]$;
- $r^{\mathbf{e}+1}(j',k')$ with $k = \llcorner d_s/r \lrcorner$, and both $(k'-k)$ and $(j'-j) \in [-\triangle, +\triangle]$;
- $r^{\mathbf{e}+2}(k',i')$ with both $(k'-k)$ and $(i'-i) \in [-\triangle, +\triangle]$.

Here, the discrete support size $\triangle = \ulcorner \sigma_c / r \urcorner = 2$ with $\sigma_c = 1.5r$, the directions of rays are changed in cycle from $\mathbf{x}$ to $\mathbf{y}$, and then $\mathbf{z}$. Again, the computation in these two filters can be mutex-free if we always retain the position before processing at each sample.

# 4 Contouring a Multi-Material Ray-rep Solid

We use a dual-contouring like algorithm to generate mesh surfaces to approximate the intermediate surfaces $\Gamma(H)$ of a heterogeneous solid $H$ from its hRay-rep $R(H)$. The algorithm consists of two steps: first, an Octree is constructed to span the space of $H$ so that the geometry and the topology in each leaf cell is simple; second, the polygons are constructed on the minimal edges to link the vertices in leaf cells.

## 4.1 Octree construction

By letting three edges of the root cell overlap with the rays $r^{\mathbf{x}}_{0,0}$, $r^{\mathbf{y}}_{0,0}$ and $r^{\mathbf{z}}_{0,0}$ of $R(H)$ respectively and the width of root cell be $(2^m+1)r$ with $m = \ulcorner \log_2 w \urcorner$, the intersected Hermite data points between the cell-edge and the solid model $H$ can be efficiently and easily detected on the hRay-rep $R(H)$. The material index on the eight nodes of a cell can also be detected easily. Starting from the root cell, the cells are recursively refined into eight sub-cells based on the condition of 1) the topology simplicity and 2) the geometry approximation error, which have been studied in [37, 38, 39]. Here, we extend to multi-material solids.

**Definition 8** For all Hermite samples in a cell $C$, a point $v_c$, the position of which minimizes the quadratic error function (QEF), is defined as the *error-minimizing point* of the cell.

The refinement of cells is stopped based on three criteria.

**Resolution Criterion** The width of cell $C$ is $r$ (i.e., the finest resolution of hRay-rep has arrived).

**Topology Criterion** The intermediate surfaces of the heterogeneous solid $H$ inside a cell $C$ have a disk-like topology.

**Geometry Criterion** The distance between the error-minimizing point $v_c$ and the planes defined by all Hermite samples in a cell $C$ is not greater than a user defined tolerance $\varepsilon_g$.

As discussed in [37, 39], both the face ambiguous configuration and the voxel ambiguous configuration will lead to complex topology inside a cell – here, these cases happen if the face diagonal nodes (or the volume diagonal nodes) on a cell $C$ are with the same material index. The multiple intersections between the ray and $\Gamma(H)$ (i.e., Hermite data) in the range of a cell edge will also make the topology of the cell complex. Moreover, it suppose to have no Hermite data in the solid (or complex) cell, which must be checked on the cells with the same material index on all their eight nodes. In the geometry criterion, $\varepsilon_g = 0.5r$ is used for all our tests.

## 4.2 Mesh Generation

The mesh generation on an octree follows the strategy given in [24]. On the minimal edges whose two endpoints are with different material indices, the polygonal faces are constructed by connecting the error-minimizing points in the cells neighboring to the minimal edges. The orientation of a face should be arranged in a way that its normal points to the homogeneous region with a smaller material index. For each face, we give it an $ID$ in the same way as Definition 5. Therefore, the polygon soup can be separated into a set of assembled two-manifold mesh patches, where each patch is formed by the polygons with the same $ID$. Details of this separation can be found in [9]. Figure 6 gives an example result of the mesh generation.

**Parallel Implementation** The step of octree construction takes the majority of computing time, which however can be processed in parallel easily. In our implementation, the root cell is firstly refined into 64 subcells in the third layer of the octree. Then, the sub-cells are assigned to different threads to be further refined for the octree construction. After this parallelization, the major time in contouring a hRay-rep solid is taken by filling the topology information of half-edge data structure instead.

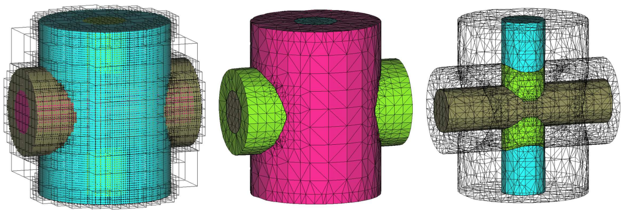


Figure 6: Example of mesh generation on hRay-rep of a heterogeneous solid: (left) octree is constructed from hRay-rep (only the non-solid/non-empty leaf cells are displayed), (middle) and (right) the assembly of mesh surface patches are generated.

# 5 Parallel Computing of General Offset Surface

Designing different filters will enable our paradigm to compute the resultant surface meshes of other solid modeling operations. We employ the computation of general offset surface as an example. General Offsetting is a very important operation in the applications of CAD/CAM [40] and biomedical engineering [2]. Unlike the uniform offset surface of a given model, the general offset surface needs to offset different thickness from the original surface. In the manufacturing applications (e.g., [40]), this leads to a more accurately machined part or can speed up the machining. In the biomedical engineering applications (e.g., [2]), this makes the simulation more realistic as the offset surface is usually employed to generate the muscle walls which in general is not uniform. Therefore, suppose we have a similar problem as [2] that given a heterogeneous solid $H = \cup_{i=0}^{n} \Omega_i$, the general offset surface of a homogeneous material region $\Omega_k$ is to be computed where different offset $\epsilon_j$ is adopted on different intermediate surface $\Gamma_{k,j}$ (or $\Gamma_{j,k}$).

It is well known that the uniform offset surface of a given model $M$ with an offset $\epsilon$ can be computed by the Minkowski sum of $M$ with a sphere centered at origin and with the radius $\epsilon$. Computing accurate Minkowski sum of a freeform model and a sphere is impractical and unnecessary. Here, an approximation of Minkowski sum is evaluated. For every sample $s$ on the intermediate surface $\Gamma_{k,j}$ (or $\Gamma_{j,k}$) of the hRay-rep $R(H)$, we place a sphere $S$ centered at $s$ and with the radius $|\epsilon_j|$. The union (or subtraction if $\epsilon_j < 0$) of $\Omega_k$ and $S$ is then computed, where the Boolean operation on Ray-rep has been well defined in [25]. Repeating the union (or subtraction) operations until all samples of the surface $\partial\Omega_k$ of $\Omega_k$ have been processed, we obtain a new hRay-rep solid $\Omega_k^\epsilon$ whose surface $\partial\Omega_k^\epsilon$ is the general offset of $\Omega_k$ and the mesh surface for which can be generated by the method in section 4.

However, computing the general offset in this way is very time-consuming (especially when processing models in high resolution of hRay-rep). A new algorithm is developed here to model the hRay-rep $R(\Omega_k^\epsilon)$ of the solid $\Omega_k^\epsilon$ in parallel. We separate the computation of $R(\Omega_k^\epsilon)$ into the computation of samples on each ray of $R(\Omega_k^\epsilon)$ in multiple-threads where the thread for the ray $r_{i,j}^{\mathbf{e}}(\Omega_k^\epsilon)$ will only generate samples on itself – thus it is mutex-free.

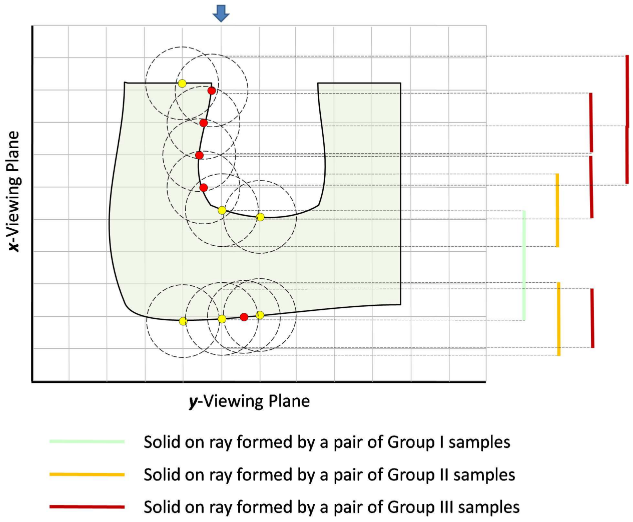

x-Viewing Plane
y-Viewing Plane
Solid on ray formed by a pair of Group I samples
Solid on ray formed by a pair of Group II samples
Solid on ray formed by a pair of Group III samples

Figure 7: The resultant samples for offset surface on a ray (pointed by blue arrow) are obtained by the samples from three different groups.

The samples on the ray $r_{i,j}^{\mathbf{e}}(\Omega_k^\epsilon)$ come from three different ways:

- Group I: The samples of the intermediate surface $\Gamma_{h,k\ (h\neq k)}$ on the ray $r_{i,j}^{\mathbf{e}}(H)$;
- Group II: Intersections between the ray $r_{i,j}^{\mathbf{e}}(H)$ and the spheres centered at the samples of $\Gamma_{h,k\ (h\neq k)}$ on the rays $r^{\mathbf{e}}(H)$;
- Group III: Intersections between the ray $r_{i,j}^{\mathbf{e}}(H)$ and the spheres centered at the samples of $\Gamma_{h,k\ (h\neq k)}$ on the rays $r^{\mathbf{e+1}}(H)$ and $r^{\mathbf{e+2}}(H)$ that are perpendicular to $r_{i,j}^{\mathbf{e}}(H)$.

The samples from each group are actually formed into pairs and the region between a pair of samples represents the solid on the ray $r_{i,j}^{\mathbf{e}}(H)$. Figure 7 gives an illustration of the samples in three groups. Usually, if $\epsilon \neq 0$, the samples in Group I will be eliminated. However, they are important to present the interval of solid before processing, so they are included in our computation. By computing the union (or substraction) of these pairs of samples, we can obtain the samples on the $r_{i,j}^{\mathbf{e}}(\Omega_k^\epsilon)$ for the hRay-rep $R(\Omega_k^\epsilon)$.

For the samples of Group II and III, it is unwise to check all $r^{\mathbf{e}}(H)$, $r^{\mathbf{e+1}}(H)$ and $r^{\mathbf{e+2}}(H)$ rays. If $\epsilon_{\max} = \max\{\epsilon_k\}$ and $\Delta_\epsilon = \ulcorner \epsilon_{\max}/r \urcorner$, only the samples on the rays of

$$r_{i',j'}^{\mathbf{e}}(H) \text{ with } ((i'-i)^2 + (j'-j)^2)r^2 < \epsilon_{\max}^2$$

are considered for generating Group II samples, and the samples on the rays of

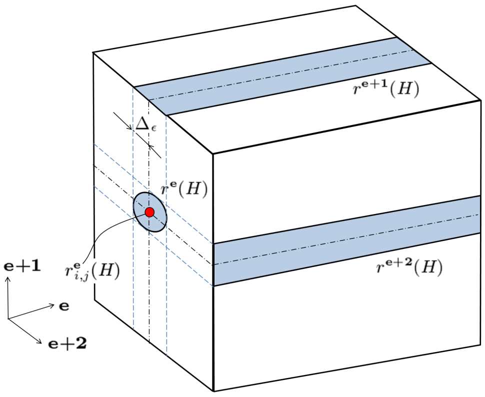


Figure 8: Valid regions of rays for obtaining Group II and Group III samples.

$$r^{\mathbf{e+1}}_{j'\in[j-\Delta_\epsilon,j+\Delta_\epsilon],k\in[1,w]}(H) \text{ and } r^{\mathbf{e+2}}_{k\in[1,w],i'\in[i-\Delta_\epsilon,i+\Delta_\epsilon]}(H)$$

are checked to generate Group III samples. Figure 8 gives an illustration of the necessary regions of rays to be considered.

Actually, the number of rays to generate Group II and III samples can be further reduced by excluding some rays. The rays that intersect with $r^{\mathbf{e}}_{i,j}(H)$ will be eliminated by the pairs of Group I samples during union operations. For example, if there is a pair of samples on $r^{\mathbf{e}}_{i,j}(H)$ with depth values $d_s$ and $d_e$, and the region on ray between $d_s$ and $d_e$ belongs to $\Omega_k$, the samples generated by the rays

$$r^{\mathbf{e+1}}_{j'\in[j-\Delta_\epsilon,j+\Delta_\epsilon],k'}(H) \text{ and } r^{\mathbf{e+2}}_{k',i'\in[i-\Delta_\epsilon,i+\Delta_\epsilon]}(H)$$

are removed from Group III if $k'r \in [d_s+\epsilon_{\max}, d_e-\epsilon_{\max}]$. Similarly, for inwards offsetting conducted by subtraction operations, if there is a pair of samples on $r^{\mathbf{e}}_{i,j}(H)$ with depth value $d_e$ and $d'_s$, and the region on ray between $d_e$ and $d'_s$ does not belong to $\Omega_k$, the samples on the above rays will be neglected if $k'r \in [d_e+\epsilon_{\max}, d'_s - \epsilon_{\max}]$ as the subtraction operations by the spheres centered at these samples will not contribute to the samples on $r^{\mathbf{e}}_{i,j}(\Omega_k)$.

By the above strategy of parallelization, the general offset surface for a homogeneous material region can be efficiently computed on multiple threads. For the general offset of the boundary surface of $H$ but not a region $\Omega_k \subset H$, it can be obtained by computing the inwards offset of material region $\Omega_0$. Figure 9 gives an example of general offset surface generated from a brain model, and the example of general offset surface generated on a Mechanical part is given in Fig.10.

## 6 Results and Discussion

We have implemented the proposed approach in C++ and tested various examples on both a consumer level PC with Intel Core 2 Quad CPU Q6600 2.4GHz + 4GB RAM and a workstation PC with two Intel Xeon Quad CPU E5440 2.83GHz + 8GB RAM. The parallel algorithms are implemented using OpenMP and Microsoft Visual Studio.

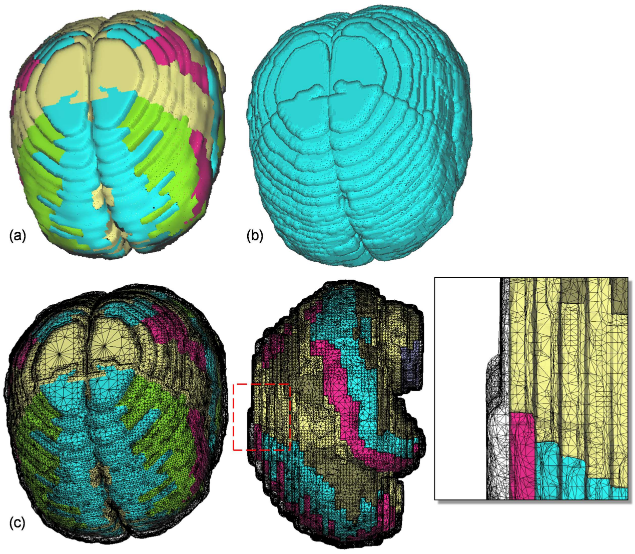


Figure 9: Generating general offset surface on a brain model: (a) the hRay-rep of the brain model, (b) the hRay-rep of the general offset surface, and (c) the meshes for the general offset surface and the brain model.

Our first example is a mechanical engine model with five different materials which has been given previously in Fig.1. The second example is a mechanical part with four cylinders in different materials (see Fig.6 and Fig.10). It is not difficult to find that the sharp features are well reconstructed on the model because of the bilateral normal and depth filters. The third example is a brain model with six different organ regions in different material labels (see Fig.9). More examples are given in Figs.11-14. The computational statistics are shown in Table 1 – all examples can be computed very efficiently. The input multi-material volumetric data of the biomedical models are obtained by segmenting the CT images using the publicly available Snap software at [41]. A more efficient segmentation can be conducted by using minimal-cut based segmentation methods (e.g., [1]).

Basically, the accuracy of final mesh surfaces is affected by the resolution of input volumetric data sets. As the generation of hRay-rep and the contouring for producing mesh surfaces are controlled by the coefficients relating to the resolution of input volume, an input with higher resolution results in a more accurate result. For example, if artifacts on the intersection lines shown in Fig.10 need to be further reduced, the best method is to ask for an input with higher resolution although it is difficult sometime.

The tests on parallel general offsetting also give encouraging results. Besides the examples given in Fig.9 and Fig.10, we have also computed the offset surface on a lion model with complex geometry (see Fig.15) with

Table 1: Computational Statistics

| Model | Figure | Voxel Resolution | Time (sec.) on a Quad-Core CPU | | | Time (sec.) on two Quad-Core CPUs | | |
|---|---|---|---|---|---|---|---|---|
| | | | hRay-rep Construct | Filtering | Meshing* | hRay-rep Construct | Filtering | Meshing* |
| Engine | 1 | $256 \times 256 \times 110$ | 0.597 | 4.61 | 2.31 (1.52) | 0.260 | 1.91 | 2.06 (1.39) |
| Cylinders | 5 & 10 | $256 \times 256 \times 256$ | 0.601 | 3.49 | 0.484 (0.211) | 0.345 | 1.46 | 0.365 (0.210) |
| Brain | 9 | $149 \times 181 \times 119$ | 0.496 | 4.28 | 3.73 (2.59) | 0.200 | 1.90 | 3.21 (2.40) |
| Fibers | 11 | $256 \times 256 \times 256$ | 0.915 | 7.78 | 0.810 (0.446) | 0.470 | 3.28 | 0.630 (0.415) |
| Knee | 12 | $193 \times 229 \times 305$ | 0.899 | 6.86 | 1.63 (0.982) | 0.400 | 2.74 | 1.36 (0.910) |
| Tooth | 13 | $512 \times 512 \times 360$ | 1.39 | 3.32 | 0.795 (0.468) | 1.17 | 1.94 | 0.720 (0.467) |
| Bone | 14 | $350 \times 350 \times 600$ | 2.02 | 12.8 | 6.31 (4.26) | 1.27 | 5.77 | 2.05 (1.43) |

Note that the time reported in the bracket is the time spent on the filling of topology in half-edge data-structure, the splitting of polygons into triangles, and the memory allocation and release management.

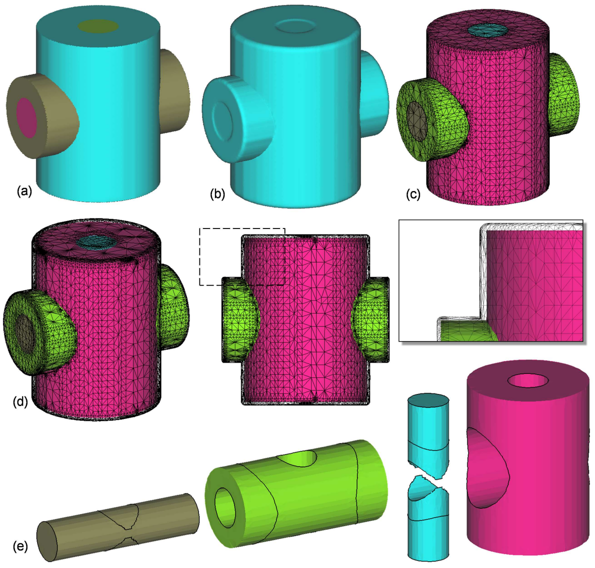


Figure 10: Generating general offset surface on a mechanical part: (a) the hRay-rep of the model, (b) the hRay-rep of the general offset surface, (c) the surfaces generated from hRay-rep – sharp features are well preserved, (d) the meshes for the general offset surface and the mechanical part, and (e) the assembly drawing of homogeneous regions.

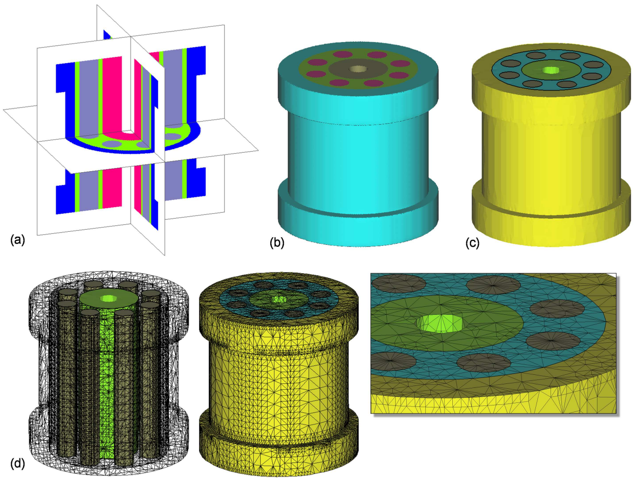


Figure 11: Example of a Fiber model: (a) the input volumetric data, (b) the hRay-rep of the model, (c) the reconstructed intermediate surfaces from the hRay-rep, and (d) the mesh surfaces.

Table 2: Computing Time on Offsetting

| Model | Res. | Offsets | Time* (sec) |
|---|---|---|---|
| Brain | $149 \times 181 \times 119$ | $0.25r \sim 5r$ | 2.55 |
| Cylinders | $256 \times 256 \times 256$ | $0.25r \sim 5r$ | 4.33 |
| Lion | $257 \times 257 \times 257$ | $-10r$ | 6.35 |

*Tested on the workstation with two Quad-Core CPUs.

a large offset value – $\epsilon = -10r$. The computing time on offsetting is listed in Table 2.

With a simple extension, our parallel algorithm of general offsetting can be modified to compute the Minkowski sum of a solid in hRay-rep with another solid in mathematical implicit representation (e.g., the superellipsoid [42] – more details can be found in Appendix B). As the intersection segments between a ray and a solid in mathematical implicit representation can be analytically computed, the Minkowski sum between a hRay-rep solid and an implicit solid can be generated by the same strategy of our algorithm in section 5. Figure 16 shows the Minkowski sum of the lion model and two superellipsoids, which can be efficiently computed in 14.46 and 6.93 seconds respectively. As the Minkowski sum has many applications in solid modeling, robotics path planning and CAD/CAM, our approach shows a good potential to benefit these areas.

## 7 Conclusions

A parallel approach is presented in this paper to compute mesh surfaces from multi-material volume data using hRay-rep (an extended Ray-rep) as an intermediate. After generating the hRay-rep of heterogeneous solid, a set of two-manifold surface meshes can be directly constructed from the hRay-rep where every homogeneous region is enclosed by an assembly of mesh patches to approximate the intermediate surfaces of heterogeneous

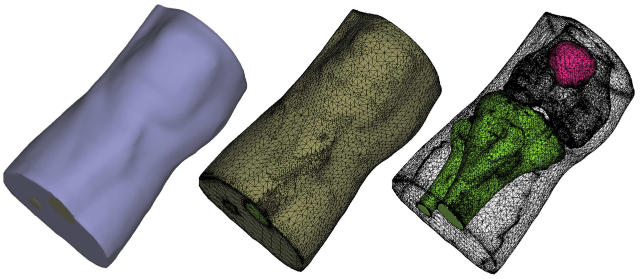


Figure 12: An example of Knee model.

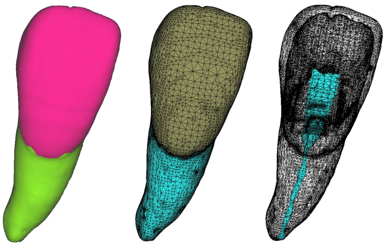


Figure 13: An example of Tooth model.

solids. Several parallel filters/operators have been developed to process the samples on the rays of a solid in hRay-rep. The experimental results show that our parallel approach can efficiently compute the surface meshes for various multi-material volumetric data sets. Moreover, as all the algorithms developed in this paper are mutex-free, it is very likely that they can be implemented on a more highly parallel platform — *Graphics Processing Unit* (GPU), which is considered as our work in the near future.

# Acknowledgments

The research presented in this paper is partially supported by CUHK Direct Grant CUHK/2050400 and Hong Kong RGC CERG Grant CUHK/416307 and CUHK/417508. The author would like to thank Mr.Francois Malan of LUMC Leiden (Orthopaedics) / TU Delft (Graphics & Visualization) who provides the volumetric data of the Bone example.

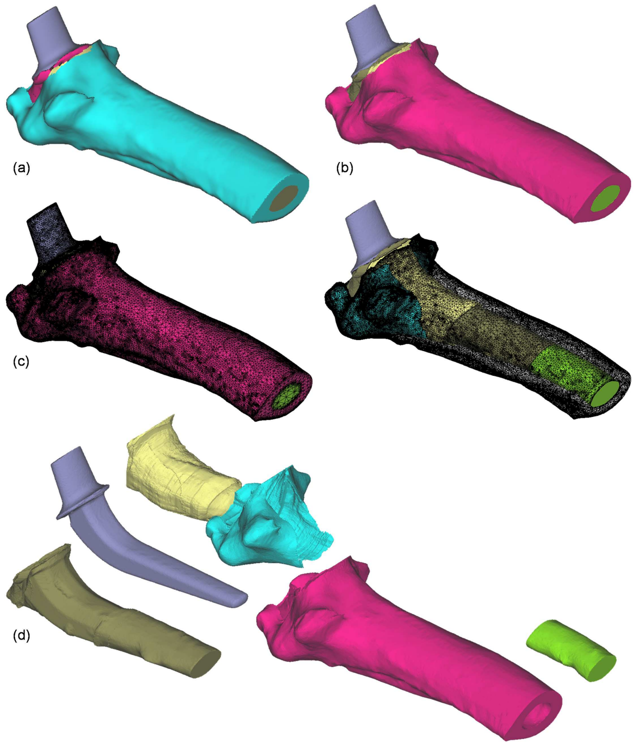


Figure 14: An example of Bone model with six different material regions: (a) the hRay-rep, (b) the reconstructed intermediate surfaces, (c) mesh representation, and (d) the six different material regions are well reconstructed into watertight mesh surfaces.

# References

[1] M. H. Shammaa, H. Suzuki, Y. Ohtake, Extraction of isosurfaces from multi-material ct volumetric data of mechanical parts, in: SPM '08: Proceedings of the 2008 ACM symposium on Solid and physical modeling, ACM, New York, NY, USA, 2008, pp. 213–220.

[2] C. Bajaj, S. Goswami, Multi-component heart reconstruction from volumetric imaging, in: SPM '08: Proceedings of the 2008 ACM symposium on Solid and physical modeling, ACM, New York, NY, USA, 2008, pp. 193–202.

[3] S. Muraki, T. Nakai, Y. Kita, K. Tsuda, An attempt for coloring multichannel mr imaging data, IEEE Transactions on Visualization and Computer Graphics 7 (3) (2001) 265–274.

[4] X. Guo, H. Qin, Real-time meshless deformation: Collision detection and deformable objects, Comput. Animat. Virtual Worlds 16 (3-4) (2005) 189–200.

[5] N. Kim, K. Choi, J.-S. Chen, M. Botkin, Meshfree analysis and design sensitivity analysis for shell structures, International Journal for Numerical Methods in Engineering 53 (9) (2002) 2087–2116.

[6] V. Shapiro, I. Tsukanov, Meshfree simulation of deforming domains, Computer-Aided Design 31 (7) (1999) 459–471.

[7] J. Peraire, J. Peiró, K. Morgan, Advancing front grid generation, in: Handbook of grid generation (edited by Thompson JF, Soni BK, Weatherill NP), CRC Press, 1999, pp. 17–1.

[8] H. Si, TetGen: A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator, http://tetgen.berlios.de/, 2007.

[9] C. Wang, Direct extraction of surface meshes from implicitly represented heterogeneous volumes, Comput. Aided Des. 39 (1) (2007) 35–50.

[10] A. Biswas, V. Shapiro, I. Tsukanov, Heterogeneous material modeling with distance fields, Comput. Aided Geom. Des. 21 (3) (2004) 215–242.

[11] M. Freytag, V. Shapiro, I. Tsukanov, Field modeling with sampled distances, Computer-Aided Design 38 (2) (2006) 87–100.

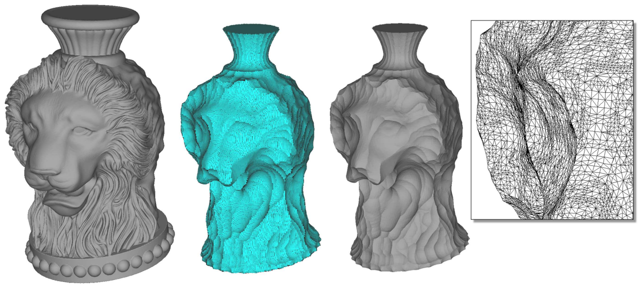


Figure 15: Parallel offsetting of a Lion model: (left) the given model, (middle) the hRay-rep of offset surface, and (right) the reconstructed mesh surface for the offset surface.

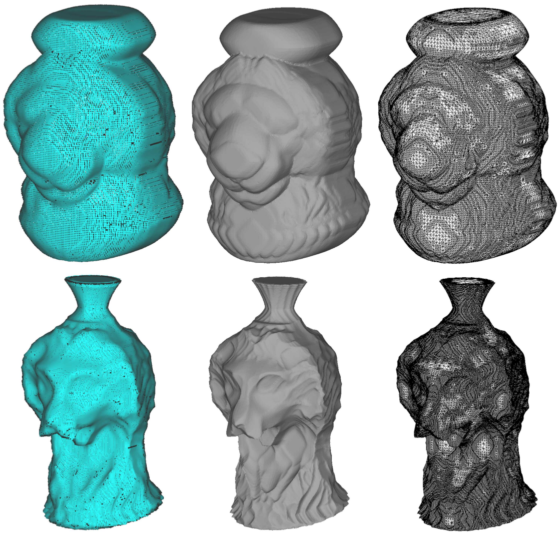


Figure 16: Minkowski sum of the Lion model with superellipsoids: (top) the superellipsoid is with $(n_1, n_2, r_x, r_y, r_z) = (1.5, 0.2, 10r, 10r, 10r)$ and (bottom) the superellipsoid is with $(n_1, n_2, r_x, r_y, r_z) = (1.5, 0.2, -5r, -5r, -5r)$.

[12] X. Qian, D. Dutta, Feature-based design for heterogeneous objects, Computer-Aided Design 36 (2004) 1263–1278.

[13] X. Qian, D. Dutta, Design of heterogenous turbine blade, Computer-Aided Design 35 (2003) 319–329.

[14] X. Qian, D. Dutta, Physics-based modeling for heterogeneous objects, Journal of Mechanical Design, Transactions of the ASME 125 (2003) 416–427.

[15] X. Qian, D. Dutta, Heterogeneous object modeling through direct face neighborhood alteration, Computers & Graphics 27 (2003) 943–961.

[16] V. Kumar, D. Dutta, An approach to modeling multi-material objects, in: Proceedings of ACM Symposium on Solid Modeling 97, ACM, 1997, pp. 336–345.

[17] W. Sun, X. Hu, Reasoning boolean operation based modeling for heterogeneous objects, Computer-Aided Design 34 (2002) 481–488.

[18] Y. Siu, S. Tan, 'source-based' heterogeneous solid modeling, Computer-Aided Design 34 (2002) 41–55.

[19] X. Kou, S. Tan, W. Sze, Modeling complex heterogeneous objects with non-manifold heterogeneous cells, Computer-Aided Design 38 (2006) 457–474.

[20] J. Hua, Y. He, H. Qin, Multiresolution heterogeneous solid modeling and visualization using trivariate simplex splines, in: SM '04: Proceedings of the ninth ACM symposium on Solid modeling and applications, Eurographics Association, 2004, pp. 47–58.

[21] W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3d surface construction algorithm, in: SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, ACM, New York, NY, USA, 1987, pp. 163–169.

[22] Z. Wu, J. J. Sullivan, Multiple material marching cubes algorith, International Journal for Numerical Methods in Engineering 58 (2003) 189–207.

[23] T. Fujimori, H. Suzuki, Surface extraction from multi-material ct data, in: Proceedings of the Ninth International Conference on Computer Aided Design and Computer Graphics, 2005, pp. 319–324.

[24] T. Ju, F. Losasso, S. Schaefer, J. Warren, Dual contouring of hermite data, in: SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, ACM, New York, NY, USA, 2002, pp. 339–346.

[25] J. L. Ellis, G. Kedem, T. C. Lyerly, D. G. Thielman, R. J. Marisa, J. P. Menon, H. B. Voelcker, The ray casting engine and ray representatives, in: SMA '91: Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications, ACM, New York, NY, USA, 1991, pp. 255–267.

[26] J. Menon, R. J. Marisa, J. Zagajac, More powerful solid modeling through ray representations, IEEE Comput. Graph. Appl. 14 (3) (1994) 22–35.

[27] J. P. Menon, H. B. Voelcker, On the completeness and conversion of ray representations of arbitrary solids, in: SMA '95: Proceedings of the third ACM symposium on Solid modeling and applications, ACM, New York, NY, USA, 1995, pp. 175–286.

[28] M. O. Benouamer, D. Michelucci, Bridging the gap between csg and brep via a triple ray representation, in: SMA '97: Proceedings of the fourth ACM symposium on Solid modeling and applications, ACM, New York, NY, USA, 1997, pp. 68–79.

[29] E. Hartquist, J. Menon, K. Suresh, H. Voelcker, J. Zagajac, A computing strategy for applications involving offsets, sweeps, and minkowski operations, Computer-Aided Design 31 (3) (1999) 175–183.

[30] C. Rocchini, P. Cignoni, F. Ganovelli, C. Montani, P. Pingi, R. Scopigno, Marching intersections: an efficient resampling algorithm for surface management, in: Proceedings of Shape Modeling International 2001, IEEE, 2001, pp. 296–305.

[31] M. Tarini, M. Callieri, C. Montani, C. Rocchini, K. Olsson, T. Persson, Marching intersections: an efficient approach to shape-from-silhouette, in: Proceedings of VMV 2002, 2002, pp. 283–290.

[32] C. Rocchini, P. Cignoni, F. Ganovelli, C. Montani, P. Pingi, R. Scopigno, The marching intersections algorithm for merging range images, The Visual Computer 20 (2004) 149–164.

[33] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Surface reconstruction from unorganized points, in: SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques, ACM, New York, NY, USA, 1992, pp. 71–78.

[34] S. Fleishman, I. Drori, D. Cohen-Or, Bilateral mesh denoising, ACM Trans. Graph. 22 (3) (2003) 950–953.

[35] T. R. Jones, F. Durand, M. Desbrun, Non-iterative, feature-preserving mesh smoothing, in: SIGGRAPH '03: ACM SIGGRAPH 2003 Papers, ACM, New York, NY, USA, 2003, pp. 943–949.

[36] C. Wang, Bilateral recovering of sharp edges on feature-insensitive sampled meshes, IEEE Trans. on Vis. and Comp. Graph. 12 (4) (2006) 629–639.

[37] A. van Gelder, J. Wilhelms, Topological considerations in isosurface generation, ACM Trans. Graph. 13 (4) (1994) 337–375.

[38] G. Varadhan, S. Krishnan, Y. J. Kim, D. Manocha, Feature-sensitive subdivision and isosurface reconstruction, in: VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03), IEEE Computer Society, Washington, DC, USA, 2003, p. 14.

[39] G. Varadhan, S. Krishnan, T. Sriram, D. Manocha, Topology preserving surface extraction using adaptive subdivision, in: SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, ACM, New York, NY, USA, 2004, pp. 235–244.

[40] B. Koc, Y.-S. Lee, Non-uniform offsetting and hollowing objects by using biarcs fitting for rapid prototyping processes, Comput. Ind. 47 (1) (2002) 1–23.

[41] Snap, http://www.itksnap.org/.

[42] A. H. Barr, Superquadrics and angle-preserving transformations, IEEE Comput. Graph. Appl. 1 (1) (1981) 11–23.

# A Proof of Proposition 2

**Proposition 2** The intermediate surfaces $\Gamma(H)$ sampled into a complete hRay-rep gives a $d$-covering of $H$ with $d$ bounded by $\sqrt{3}r$, where $r$ is the sampling distance between horizontal (or vertical) rays in a $R(\mathbf{e}, H)$.
*Proof.* The rays in a complete hRay-rep actually form many cubic cells. The samples of hRay-rep are located at the edges of the cells. After analyzing the possible configurations of the intermediate surfaces inside the cells, the configuration with the longest distance from a surface point $\mathbf{p}$ to the samples on hRay-rep is as shown in Fig.17. The distances from $\mathbf{p}$ to the intersections on the cell edges are $\sqrt{3}r$. Therefore, after sampling a given heterogeneous solid $H$ into hRay-rep with sampling distance $r$, the obtained point set $S$ gives a $d$-covering of $H$ with $d \leq \sqrt{3}r$.

◇

# B Superellipsoid

The superellipsoid gives a family of shapes formed from the spherical product of two superquadratric curves, whose implicit representation is

$$(|\frac{x}{r_x}|^{\frac{2}{n_2}} + |\frac{y}{r_y}|^{\frac{2}{n_2}})^{\frac{n_2}{n_1}} + |\frac{z}{r_z}|^{\frac{2}{n_1}} = 1. \qquad (10)$$

Figure 18 gives examples of superellipsoid with different parameters in $n_1$ and $n_2$.

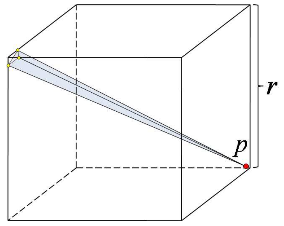


Figure 17: The configuration with the longest distance from a surface point $\mathbf{p}$ to the samples – the yellow ones on the edges of a cell.

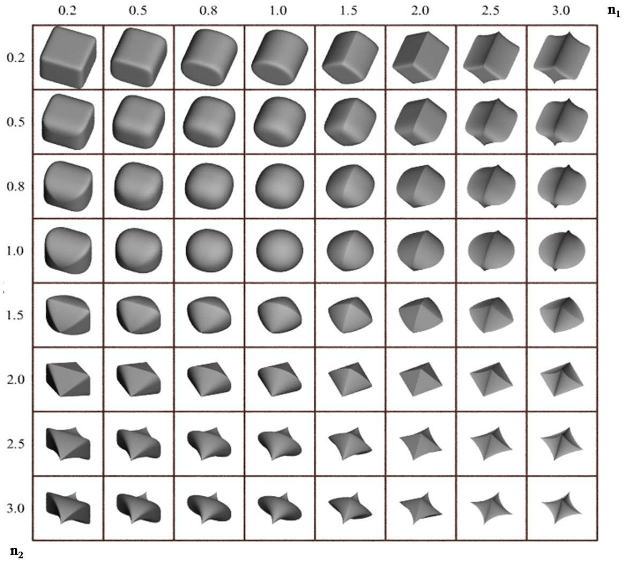


Figure 18: Choosing different $n_1$ and $n_2$ leads to different shapes which however are all bounded by the volume: $[-r_x, r_x] \times [-r_y, r_y] \times [-r_z, r_z]$.