

# **Voronoi-Based Surface Mesh (B-rep) Generation from Segmented Medical Imaging**

Omar Hafez  
Mark Rashid  
Department of Civil Engineering  
UC Davis

# 1 Motivation

The current state-of-the-art for image-based modeling and simulation in biomechanics involves the following series of steps:

1. **Image:** Begin with MRI or other imaging data, in the form of stacked two-dimensional slices of voxel intensity data (a voxel being a three-dimensional analog to a pixel).
2. **Mask:** Segment the image using a combination of level-set, flood fill, and manual fine-tuning techniques.
3. **Preliminary Finite Element Mesh:** Construct the computational mesh directly from the segmented image via octree subdivision on an initially-coarse structured hex mesh to generate a mesh consisting of conforming hex, pyramid, prism, and tet elements.
4. **Final Finite Element Mesh:** Extract the surface of the preliminary mesh, and apply a tetrahedral meshing algorithm (e.g., advancing front or Delaunay refinement) for a fully tetrahedral mesh.
5. **Analysis:** Apply boundary conditions, define material behavior, run simulation, and visualize.

Our proposed pipeline is as follows:

1. **Image:** Begin with MRI or other imaging data, in the form of stacked two-dimensional slices of voxel intensity data.
2. **Mask:** Segment the image using a combination of level-set, flood fill, and manual fine-tuning techniques.
3. **B-Rep:** Generate a watertight, facetized boundary representation (b-rep) from segmented image.
4. **Polyhedral Finite Element Mesh:** Construct domain-conforming polyhedral mesh from b-rep.
5. **Analysis:** Apply boundary conditions, define material behavior, run analysis, and visualize.

The proposed approach has a few important advantages over current segmented-image-to-mesh schemes. By decoupling the volume meshing from the image-based geometry definition, the mesh design is far less constrained: the mesh can, e.g., be graded and/or locally refined freely to reflect solution or geometry features. Sequences of meshes can even be generated to follow important solution features as they evolve with time. Also, generation of an explicit b-rep supports rigorous geometry verification and manual augmentation of the geometric model, e.g. by introduction of an internal interface between atria and ventricle. And, should manual manipulation of the image-segmentation process be necessary, a proper b-rep provides a more sound starting point for such intervention than does a volume mesh. The proposed approach also supports a high level of automation in the image-to-mesh process by sequestering the geometry definition from the mesh generation: once a valid, watertight b-rep exists, the meshing step is guaranteed to complete correctly, with no human “tweaking” required.

Lastly, the polyhedral meshing step separates the geometry resolution from the simulation resolution, such that one may run a coarse simulation on an object whose geometry is finely described. In biomechanics applications where the geometry may have substantial influence on the simulation solution, this may indeed be the most important benefit. For the same b-rep, one may run significantly coarser simulations with this polyhedral meshing approach than when using conventional meshing techniques. Thus, the system that is solved on each time step can be reduced by a factor

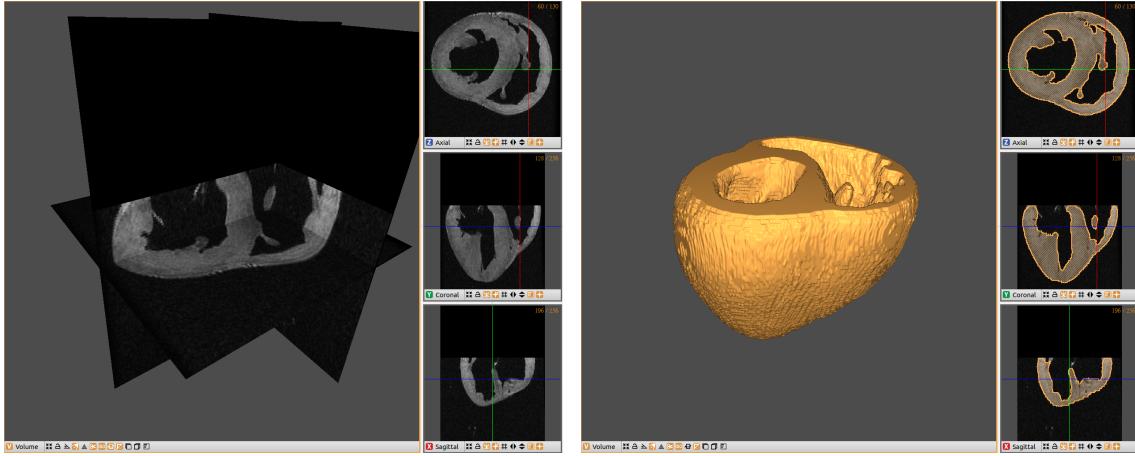
of 2, and in some cases by an order of magnitude.

The novelty in the generation of the b-rep and in the use of polyhedral finite elements both provide important benefits. A b-rep could be simply be extracted from the segmented image as the collection of voxel edges that define interfaces, but this requires a smoothing step that may lead to topological changes in the boundary and/or loss of watertightness. Conventional finite elements could be used as well, but not without serious sacrifice. Automatic meshing of hex meshing is still largely an open problem, and automatic tet meshing can lead to meshes with elements of poor quality, requiring additional processing prior to analysis. Additionally, linear tetrahedra are impractically inaccurate for solid mechanics applications, and quadratic tetrahedra can be prohibitively expensive, especially with the application of interest often requiring high-performance computing techniques.

This document focuses on step 3 of the proposed pipeline, i.e., generation of a watertight, facetized b-rep directly from a segmented image. For the time being, two-material scans will be considered. Specifically, each voxel in a scan either belongs to material  $m$  or is void material. The terms “surface mesh” and “boundary representation (b-rep)” will be used interchangeably herein.

## 2 Segmented Image Data

Given a raw image of voxel intensity values, to *segment* an image means to tag each voxel with a material type. The input for our purposes is a segmented image, often referred to as a *mask*. Segmented image data is simply an array of material assignment values for each voxel in the scan. Below is an example of a raw image and corresponding segmented image. The segmented image is used to produce a watertight, explicitly defined facetized boundary representation of the object of interest.



**Figure 1:** Raw image (left) and segmented image (right) for ex-vivo canine heart

### 3 B-Rep Generation

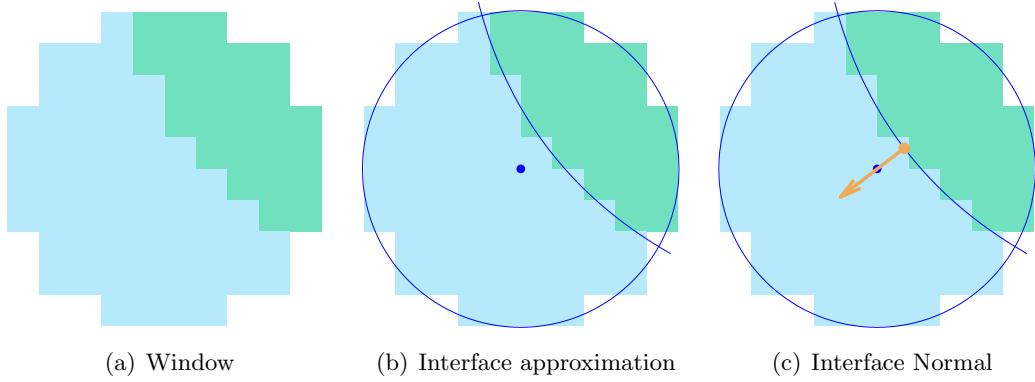
The proposed process from segmented image to watertight b-rep involves the following steps:

1. Generate an oriented point cloud defining the surface of the object. This is performed by sampling the image with overlapping “windows” to locally approximate material interfaces. For each window:
  - Determine whether the window involves an interface
  - Approximate the interface via functional minimization
  - Place interface a point and associated normal based on the interface
2. Generate Voronoi site set from oriented point cloud and perform Voronoi partition of image
3. Define b-rep as the set of facets that share Voronoi sites with different material type, and perform cleanup on that surface
4. Decimate, or coarsen, the surface to a more tractable mesh resolution

Each of these steps will be discussed in turn.

#### 3.1 Point Cloud Generation

The sequence to be described for point cloud generation is demonstrated in 2D in Figure 2. For this demonstration, green voxels belong to material  $m$  and blue voxels are void material.



**Figure 2:** Sequence of steps for point cloud generation

##### 3.1.1 Window Selection

Define the set of all voxels in a segmented image as  $\mathcal{I}$ . The segmented image  $\mathcal{I}$  is sampled with overlapping windows, each of which is a voxelated sphere with voxel radius  $R_{\mathcal{W}}$ . The sampling distance between adjacent windows is  $d_{\mathcal{W}}$ . The set of voxels in a particular window is defined as  $\mathcal{W}$ , and the number of voxels in that window is  $W$ . Define the set of voxels in  $\mathcal{W}$  that belong to material  $m$  to be  $\mathcal{M}$ , and the number of voxels in  $\mathcal{M}$  to be  $M$ . A window  $\mathcal{W}$  is deemed acceptable if it satisfies certain threshold requirements. Define  $k_{\mathcal{M}}$  as the ratio of voxels in  $\mathcal{M}$  to the voxels

in  $\mathcal{W}$ , i.e., the ratio  $M/W$ . For a threshold value  $\bar{k}_{\mathcal{M}}$ , the window is discarded if  $k_{\mathcal{M}} > \bar{k}_{\mathcal{M}}$  or  $k_{\mathcal{M}} < 1 - \bar{k}_{\mathcal{M}}$ .

Variable	Description
$\mathcal{I}$	set of all voxels in a segmented image
$\mathcal{W}$	set of all voxels in a window
$m$	material of interest
$k_{\mathcal{M}}$	ratio of voxels in $\mathcal{M}$ to voxels in $\mathcal{W}$
$\bar{k}_{\mathcal{M}}$	threshold value for $k_{\mathcal{M}}$
$R_{\mathcal{W}}$	window radius (in voxels)
$d_{\mathcal{W}}$	sampling distance between adjacent windows (in voxels)

**Table 1:** List of variables for window selection

### 3.1.2 Interface Approximation

For a particular window  $\mathcal{W}$ , we seek to approximate the interface between voxels belonging to  $\mathcal{M}$  and voxels belonging to  $\mathcal{W} \setminus \mathcal{M}$  by a curve  $\mathcal{C}$ . Assuming  $\mathcal{W}$  has been deemed acceptable, an approximating sphere  $\mathcal{D}$  of radius  $R$  is defined, such that the center and volume of this sphere match the center and volume of the window  $\mathcal{W}$ .

Define  $\Omega_p$  as the physical space covered by the voxels belonging to  $\mathcal{M}$ . Define  $\Omega$  as the physical space covered by an approximating lens to  $\Omega_p$ , determined by the space enclosed by  $\mathcal{D}$  and  $\mathcal{C}$ . We determine the curve  $\mathcal{C}$  by minimizing the error in geometric properties between  $\Omega$  and  $\Omega_p$ . See Figure 3.

The curve  $\mathcal{C}$  that approximates the interface may take many forms, but for these purposes the interface is approximated as a plane. If the center of the window is at the origin, the interface takes the form  $x' = d$ , where  $(x', y', z')$  are in a rotated coordinate system and  $d$  is the perpendicular distance from the plane to the origin. See Figure 4 below.

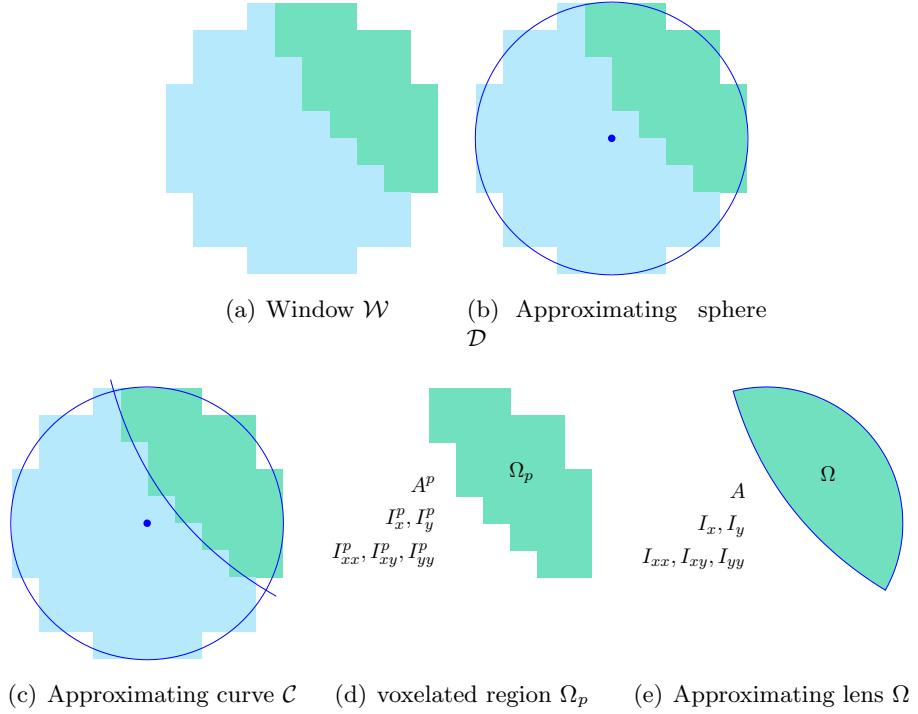
Define the following geometric quantities for  $\Omega_p$ :

$$V^p = \int_{\Omega_p} dv \quad (1)$$

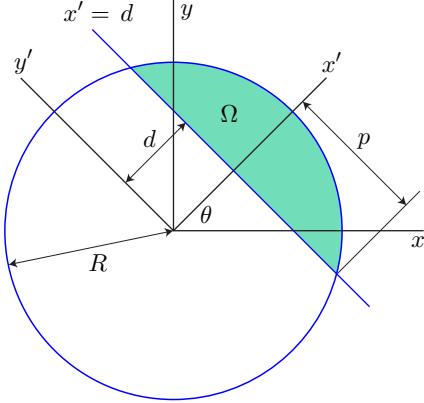
$$I_x^p = \int_{\Omega_p} x dv, \quad I_y^p = \int_{\Omega_p} y dv, \quad I_z^p = \int_{\Omega_p} z dv \quad (2)$$

$$I_{xy}^p = \int_{\Omega_p} xy dv, \quad I_{xz}^p = \int_{\Omega_p} xz dv, \quad I_{yz}^p = \int_{\Omega_p} yz dv \quad (3)$$

$$I_{xx}^p = \int_{\Omega_p} (y^2 + z^2) dv, \quad I_{yy}^p = \int_{\Omega_p} (x^2 + z^2) dv, \quad I_{zz}^p = \int_{\Omega_p} (x^2 + y^2) dv \quad (4)$$



**Figure 3:** Interface approximation demonstrated in 2D



**Figure 4:** Plane interface approximation demonstrated in 2D

And similarly for  $\Omega$ :

$$V = \int_{\Omega} dv \quad (5)$$

$$I_x = \int_{\Omega} x dv, \quad I_y = \int_{\Omega} y dv, \quad I_z = \int_{\Omega} z dv \quad (6)$$

$$I_{xy} = \int_{\Omega} xy dv, \quad I_{xz} = \int_{\Omega} xz dv, \quad I_{yz} = \int_{\Omega} yz dv \quad (7)$$

$$I_{xx} = \int_{\Omega} (y^2 + z^2) dv, \quad I_{yy} = \int_{\Omega} (x^2 + z^2) dv, \quad I_{zz} = \int_{\Omega} (x^2 + y^2) dv \quad (8)$$

Variable	Description
$\mathcal{D}$	approximating sphere to $\mathcal{W}$
$\mathcal{C}$	surface approximating interface of interest
$R$	radius of $\mathcal{D}$
$\Omega_p$	physical space covered by voxels belonging to $\mathcal{W}$
$\Omega$	physical space covered by approximating lens of $\Omega_p$
$(x, y, z)$	axes aligned with image scan, whose origin is at the center of window $\mathcal{W}$
$(x', y', z')$	orthogonal axes whose origin is at the center of window $\mathcal{W}$ , where $x'$ passes through the centroid of $\Omega$
$d$	perpendicular distance from center of $\mathcal{D}$ to plane $\mathcal{C}$
$\psi$	yaw angle of rotation between $(x, y, z)$ and $(x', y', z')$ axes
$\theta$	pitch angle of rotation between $(x, y, z)$ and $(x', y', z')$ axes
$V^p$	volume of $\Omega_p$
$I_x^p, I_y^p$	first moments of volume of $\Omega_p$
$I_{xy}^p, I_{xz}^p, I_{yz}^p$	products of volume of $\Omega_p$
$I_{xx}^p, I_{xy}^p, I_{yy}^p$	second moments of volume of $\Omega_p$
$V$	volume of $\Omega$
$I_x, I_y$	first moments of volume of $\Omega$
$I_{xy}, I_{xz}, I_{yz}$	products of volume of $\Omega$
$I_{xx}, I_{xy}, I_{yy}$	second moments of volume of $\Omega$
$e_0$	relative error in zeroth moment of volume
$e_1$	relative error in first moment of volume
$e_2$	relative error in second moment of volume
$\beta_0$	functional weighting coefficient for zeroth moment of volume error
$\beta_1$	functional weighting coefficient for first moment of volume error
$\beta_2$	functional weighting coefficient for second moment of volume error
$\varepsilon$	tolerance for minimization of functional $\mathcal{F}$
$\bar{\mathcal{F}}$	largest acceptable value of functional $\mathcal{F}$

**Table 2:** List of variables for interface approximation

For the purposes of computing the error functional, define the following quantities:

$$f^p = V^p, \quad f(d, \psi, \theta) = V \quad (9)$$

$$\mathbf{g}^p = \begin{bmatrix} I_x^p \\ I_y^p \\ I_z^p \end{bmatrix}, \quad \mathbf{g}(d, \psi, \theta) = \begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix} \quad (10)$$

$$\mathbf{h}^p = \begin{bmatrix} I_{xx}^p & -I_{xy}^p & -I_{xz}^p \\ -I_{xy}^p & I_{yy}^p & -I_{yz}^p \\ -I_{xz}^p & -I_{yz}^p & I_{zz}^p \end{bmatrix}, \quad \mathbf{h}(d, \psi, \theta) = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad (11)$$

Define the relative errors as:

$$e_0(d, \psi, \theta) = \sqrt{\frac{(f - f^p)^2}{(f^p)^2}} \quad (12)$$

$$e_1(d, \psi, \theta) = \sqrt{\frac{(g_i - g_i^p)(g_i - g_i^p)}{g_i^p g_i^p}} \quad (13)$$

$$e_2(d, \psi, \theta) = \sqrt{\frac{(h_{ij} - h_{ij}^p)(h_{ij} - h_{ij}^p)}{h_{ij}^p h_{ij}^p}} \quad (14)$$

Finally, we define the relative error functional as follows:

$$\mathcal{F}(d, \psi, \theta) = \beta_0 e_0 + \beta_1 e_1 + \beta_2 e_2 \quad (15)$$

where  $e_0$ ,  $e_1$ , and  $e_2$  are relative errors between the zeroth, first, and second order moments of volume of  $\Omega_p$  and  $\Omega$ , respectively. They are functions of  $d$ ,  $\psi$ , and  $\theta$  defined above. The values  $\beta_0$ ,  $\beta_1$ , and  $\beta_2$  are weighting coefficients, where  $\beta_2 = 1 - \beta_0 - \beta_1$ , and  $\beta_i \in [0, 1]$ . We seek the solution to the following unconstrained minimization problem:  $\min_{d, \psi, \theta} \mathcal{F}(d, \psi, \theta)$ , which can be solved by a variety of well-established techniques.

The simplex search method is currently used to determine the functional minimum. A convergence tolerance  $\varepsilon$  is defined such that on successive iterations,  $|\mathcal{F}_{i+1} - \mathcal{F}_i| < \epsilon$ , where the subscript corresponds to the iteration number. Additionally, the solution is only accepted if the error functional is below a specified value  $\bar{\mathcal{F}}$ , i.e.,  $\Omega$  approximates  $\Omega_p$  to a satisfactory degree. Specifically, we require that once convergence has been satisfied, the criterion  $\mathcal{F} < \bar{\mathcal{F}}$  must also be met for the point and normal to be retained.

The geometric quantities of  $\Omega_p$  are computed based on voxel dimensions and use of the parallel axis theorem. The geometric quantities of  $\Omega$  are computed based on  $d$ ,  $\psi$ , and  $\theta$  as follows:

$$p = \sqrt{R^2 - d^2} \quad (16)$$

$$V^* = 2\pi \left[ -\frac{1}{2}dp^2 + \frac{1}{3}R^3 - \frac{1}{3}(R^2 - p^2)^{3/2} \right] \quad (17)$$

$$I_{x'x'}^* = \frac{\pi}{30} \left[ -15dp^4 + \sqrt{R^2 - p^2} (12p^4 - 4p^2R^2 - 8R^4) + 8R^5 \right] \quad (18)$$

$$I_{y'y'}^* = \frac{\pi}{480} \left[ -160d^3p^2 + 128R^5 - \sqrt{R^2 - p^2} (128R^4 - 96p^2R^2 - 32p^4) - 120dp^4 \right] \quad (19)$$

$$V = \begin{cases} V^*, & \text{if } d \geq 0 \\ V^* + \frac{4\pi}{3} (R^2 - p^2)^{3/2}, & \text{otherwise} \end{cases} \quad (20)$$

$$I_{x'} = \frac{\pi}{4} [2p^2(R^2 - d^2) - p^4] \quad (21)$$

$$I_{y'} = I_{z'} = 0 \quad (22)$$

$$I_{x'y'} = I_{x'z'} = I_{y'z'} = 0 \quad (23)$$

$$I_{x'x'} = \begin{cases} I_{x'x'}^*, & \text{if } d \geq 0 \\ I_{x'x'}^* + \frac{4\pi}{15}(R^2 - p^2)^{3/2}(3p^2 + 2R^2), & \text{otherwise} \end{cases} \quad (24)$$

$$I_{y'y'} = \begin{cases} I_{y'y'}^*, & \text{if } d \geq 0 \\ I_{y'y'}^* + \frac{2\pi}{15}(R^2 - p^2)^{3/2}(p^2 + 4R^2), & \text{otherwise} \end{cases} \quad (25)$$

$$I_{z'z'} = I_{y'y'} \quad (26)$$

$$\mathbf{R} = \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi & \cos \psi \sin \theta \\ \sin \psi \cos \theta & \cos \psi & \sin \psi \sin \theta \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (27)$$

$$\begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix} = \mathbf{R} \begin{bmatrix} I_{x'} \\ I_{y'} \\ I_{z'} \end{bmatrix} \quad (28)$$

$$\mathbf{I}' = \begin{bmatrix} I_{x'x'} & -I_{x'y'} & -I_{x'z'} \\ -I_{x'y'} & I_{y'y'} & -I_{y'z'} \\ -I_{x'z'} & -I_{y'z'} & I_{z'z'} \end{bmatrix} \quad (29)$$

$$\mathbf{I} = \mathbf{R} \mathbf{I}' \mathbf{R}^T = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad (30)$$

The solution to the minimization problem yields the values  $d$ ,  $\psi$ , and  $\theta$  that determine the optimally oriented surface  $\mathcal{C}$ , from which the interface normal is defined.

### 3.1.3 Interface Point and Normal Placement

Once the parameters  $d$ ,  $\psi$ , and  $\theta$  are selected to fully define the surface that approximates the interface, a point location and outward normal are determined. The outward pointing normal is defined from the transformation of the  $x$  axis to the  $x'$  axis, i.e., the first column of matrix  $\mathbf{R}$ , or  $\mathbf{n} = -(\cos \psi \cos \theta, \sin \psi \cos \theta, -\sin \theta)$ . With the origin at the center of sphere  $\mathcal{D}$ , the location of the interface normal is then simply  $\mathbf{x}_p = -d\mathbf{n}$ .

## 3.2 Voronoi Site Generation and Voronoi Partitioning

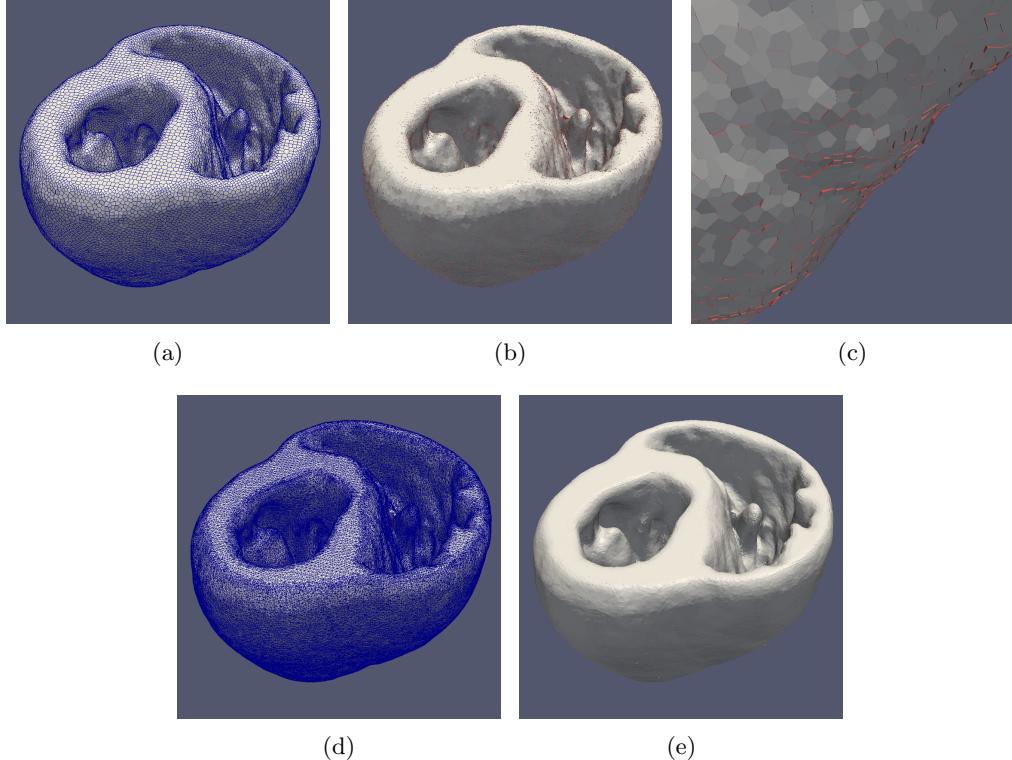
Given the orientation and location of the interface normal, a pair of Voronoi sites are placed on either side of the point along the line of action of the normal. They are separated from the point by a distance  $b$ . Thus, their locations are  $\mathbf{x}_p \pm b\mathbf{n}$ . The two Voronoi sites are assigned material types.

For the two-material case that is considered here, the site corresponding to the position  $\mathbf{x}_p - b\mathbf{n}$  is *inside*  $\Omega$ , and thus is assigned material  $m$ . The site corresponding to the position  $\mathbf{x}_p + b\mathbf{n}$  is *outside*  $\Omega$ , and thus is assigned a void material. Voronoi partitioning is performed using [Qhull](#).

### 3.3 B-Rep Extraction and Cleanup

The b-rep is extracted from the Voronoi partition as the set of facets that share Voronoi sites of differing material types. This b-rep is guaranteed to be watertight and topologically correct. “Good facets” are identified as those b-rep facets whose Voronoi sites belong to a site pair originating from the same point and normal, and “bad facets” are those b-rep facets whose Voronoi sites originate from different points in the point cloud.

“Bad edges” are identified as those that share two bad facets. Networks of bad edges are identified, and each network is collapsed to one point. Any good facet that is connected to a network of bad segs is modified to connect to the new collapsed point. The good facets are triangulated to maintain planarity following the removal of bad facets. This increases the data required to define the surface mesh, but the decimation step that follows resolves this issue immediately. Figure 5 shows the b-rep cleanup steps. Following the cleanup step, a small set of non-manifold regions arise, which are readily handled.



**Figure 5:** Voronoi-based surface reconstruction: (a) Extracted b-rep from Voronoi-partition, (b) “good” and “bad” facets identified, where bad facets are colored red, (c) detailed view of good and bad facets, (d), result of mesh cleanup, still requiring decimation step, (e) mesh cleanup result with edges hidden

### 3.4 Surface Decimation

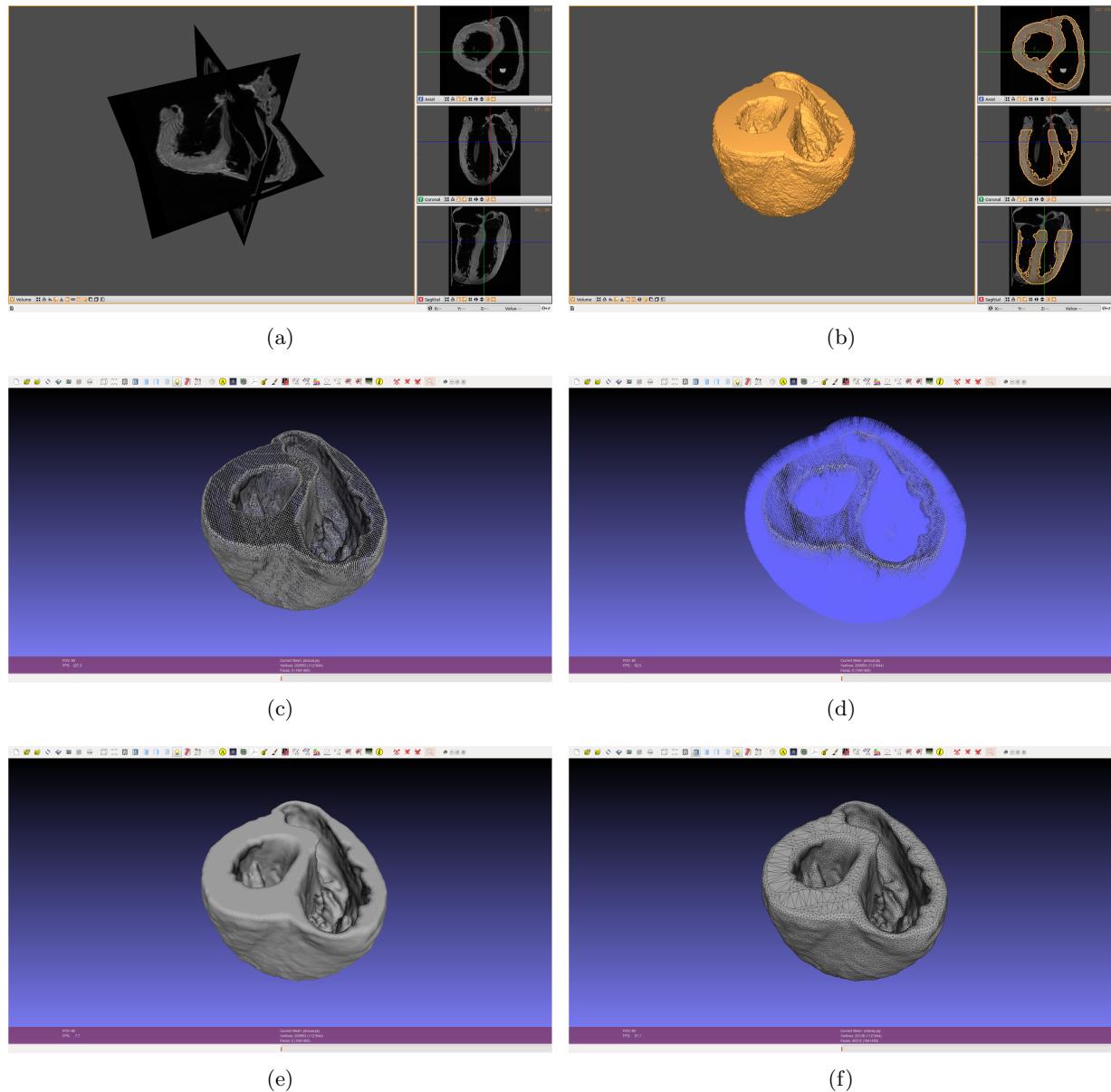
Surface decimation is performed to reduce the surface mesh to a practical size without significantly changing the geometry. Decimation is performed using [ACVD](#). Refer to the doc directory for their publication on surface decimation/coarsening.

## 4 Results

Results of the procedure are shown in Figure 6. The optimal parameters are shown in the table below.

Variable	Description	Value
$R_{\mathcal{W}}$	window radius (in voxels)	5
$d_{\mathcal{W}}$	sampling distance between adjacent windows (in voxels)	2
$\bar{k}_{\mathcal{M}}$	threshold for acceptable ratio of voxels in window $\mathcal{W}$ belonging to material $m$	0.925
$\beta_0$	functional weighting coefficient for zeroth moment of volume	0.5
$\beta_1$	functional weighting coefficient for first moment of volume	0.3
$\varepsilon$	tolerance for minimization of functional $\mathcal{F}$	$10^{-14}$
$\bar{\mathcal{F}}$	largest acceptable value of functional $\mathcal{F}$	0.15
$b$	distance from interface normal and corresponding Voronoi site pair (in voxels)	1.1

**Table 3:** Optimal parameter values for b-rep generation



**Figure 6:** Sequence of steps for b-rep generation for ex-vivo human heart: a) raw image, b) segmented image, c) point cloud, d) oriented point cloud, e) extracted b-rep from Voronoi partition, and f) final decimated surface