

# Construction of Models and Meshes of Heterogeneous Material Microstructures from Image Data

Ottmar Klaas, Mark W. Beall, and Mark S. Shephard

**Abstract** This chapter presents a set of procedures that start from image data to construct a non-manifold geometric model that supports the effective generation of meshes with the types of mesh configurations and gradations needed for efficient simulations. The types of operations needed to process the image information before and during the creation of the non-manifold geometric domains are outlined, with emphasis on those methods that are most appropriate for the analysis of materials system's behavior.

## 1 Introduction

One of the most important scales quantifying the behavior of materials is the mesoscale at which the mechanics of grain interfaces, voids and inclusions can be modeled. Often neglected in the development of mesoscale simulation technologies are the tools needed to support the accurate definition of the heterogeneous mesoscale geometry and automatic generation of meshes for the accurate prediction of the critical solution parameters. The accurate representation of the mesoscale geometry requires a statistically accurate representation of the grains, interfaces, voids, and inclusions as they exist in the as processed material systems.

Imaging technologies, such as X-ray computed microtomography (XCMT), have continued to develop to the point that they can provide a voxel level description of many important materials systems with sufficient resolution to construct the needed mesoscale geometries.

---

O. Klaas (✉) · M.W. Beall  
Simmetrix Inc., 10 Executive Park Drive, Clifton Park, NY 12065, USA  
e-mail: [oklaas@simmetrix.com](mailto:oklaas@simmetrix.com)

M.W. Beall  
e-mail: [mbeall@simmetrix.com](mailto:mbeall@simmetrix.com)

M.S. Shephard  
Rensselaer Polytechnic Institute, Troy, NY 12180, USA  
e-mail: [shephard@rpi.edu](mailto:shephard@rpi.edu)

This paper presents a set of components that take 3D voxel data, originating from either a statistically generated 3D description of a polycrystal material or a set of image slices, as input and, in combination with knowledge of the material systems involved and simulation methods to be applied, construct proper geometric representations and associated simulation discretizations (meshes) suitable for use in mesoscale analyzes. The methods used in the generation of these representations and discretizations include (i) tools to deal with irregularities at the level of the voxel data set (elimination of physically impossible voxel constellations and small features introduced by scanning or segmentation errors) (ii) tools to treat quantization artifacts at the level of the geometric model, and (iii) methods to enforce periodic boundaries when desired. With a geometric model as one of the outcomes of the presented process, the user has the ability to set mesh control such that the meshes generated support the simulation requirements in terms of parameters to be predicted and accuracy required.

Examples are presented to show the effectiveness of the presented methods. They start from data represented as a series of images representing a 3D volume, or data sets generated by an electron back scatter diffraction method.

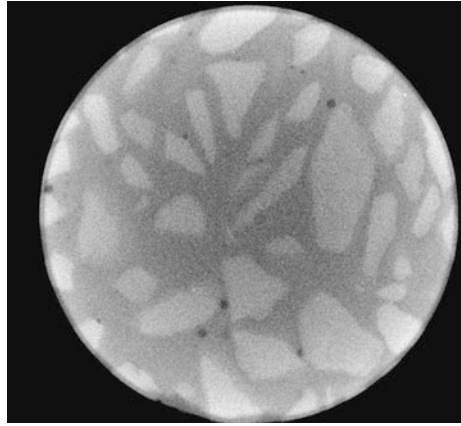
## 2 Image Data Input

The continued advancement of image techniques is providing “geometric information” in terms of discrete gray scale levels over a uniform grid of voxels. The use of such information in many medical related areas is common place with its use in combination with various simulation techniques increasing at a dramatic rate. Another area of application where image data is beginning to be used on a more regular basis is the quantification of as processed materials.

### 2.1 3D Voxel Using XCMT

Commercially available imaging systems such as X-ray computed microtomography (XCMT) [6] systems have advanced to the point that they can provide accurate microstructural information for many materials (for examples see e.g. [5]). The output of the imaging system is a set of image slices that, put together provide a set of voxels with image intensity data associated with each voxel. The first step in the process of constructing the geometric model is to convert that image data into voxel sets where each voxel is labeled such that voxels with similar characteristics receive the same label. The labeled voxel set is called a segmentation of the original data set. The characteristics to consider vary depending on the application. In medical imaging different distinct tissues may have the same gray-level appearance and thus the gray level alone is not a sufficient indicator to assign a voxel label. On the other hand, due to deficiencies of the imaging process, materials that ought to be identical

**Fig. 1** Image slice of a XCMT dataset of a concrete specimen



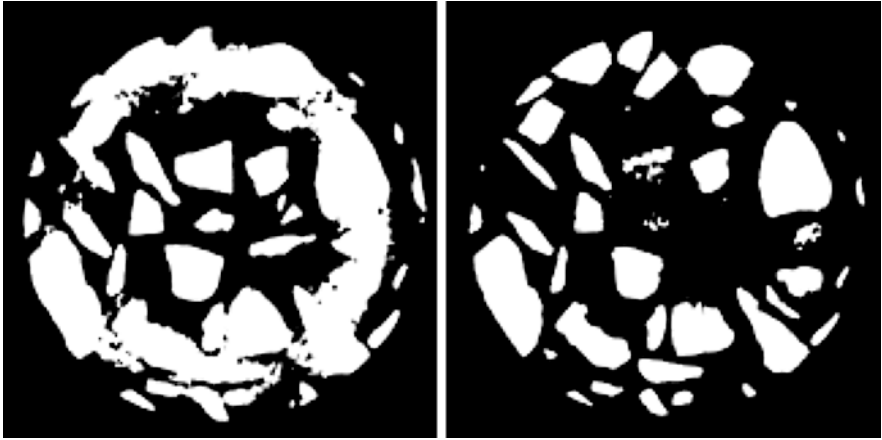
can be represented with different gray levels. This effect is often seen by comparing the gray level of objects in the center of the image with the gray level of objects closer to the boundary of the image.

There are a number of algorithmic approaches, and associated software, to carry out the segmentation process [3]. The open source Insight Segmentation and Registration Toolkit (ITK) [12] includes a wide variety of these algorithms and is used in this work. 3DSlicer is a graphical front end to a subset of the ITK algorithms [8, 22, 23, 31].

The basic segmentation algorithms are thresholding procedures, which group gray values into buckets based on a threshold value and their location in the histogram. Edge detection algorithms try to find connected voxels forming regions by locating rapid changes in the image. Region growing methods [30] start from a set of seed points that are iteratively grown by comparing the points identified with the neighboring voxels and deciding whether they are to be included into the region or not based on information gathered from already segmented areas.

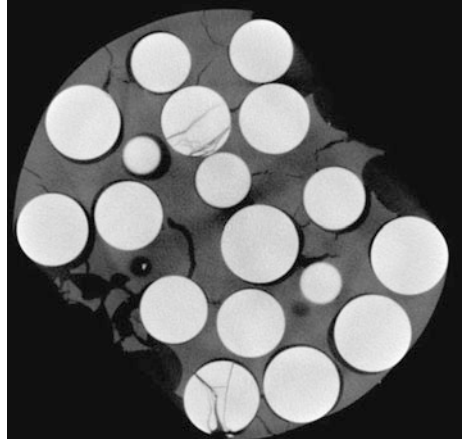
The complexity of the algorithms and the parameters to control the behavior of those algorithms that need to be applied is a function of the degree of contrast of the constituent materials to be detected as well as the level and type of noise present in the image data. When noise is present, this image needs to be preprocessed with noise filtering algorithms before it can be handed to the segmentation procedures. Again there are a wide variety of noise filters available, each with different characteristics and thus they have to be appropriately paired with the segmentation method to be used. As an example, many noise filters will by their very nature smear out distinct sharp jumps in gray level values, thus making their choice inadequate for segmentation algorithms that operate on detecting such features (e.g. edge detection segmentation procedures). Other noise filters, e.g. the so called anisotropic diffusion filter, are designed to reduce noise while preserving the features that edge detection algorithms rely on for the segmentation process.

Figure 1 shows an example of image deficiencies that the segmentation process has to contend with. The image shows significant randomly distributed noise, imaging artifacts presenting themselves as concentric circles, and a very visible change



**Fig. 2** Unacceptable segmentation results obtained using thresholding

**Fig. 3** Image slice of Large Glass beads suspended in hydroxyl-terminated polybutadiene



in the average gray level toward the boundary of the circular image domain. Trying to simply apply global thresholding to segment this data leads to inadequate results, capturing either the center area or the boundary area (Fig. 2 left and right, respectively) correctly, leaving some particles undetected or detecting the matrix material as particles as well.

Note that in the case of imaging of inanimate objects many of the problems may be overcome as the imaging process can use higher doses of radiation without having to worry about harming the object. Figure 3 shows an image slice of glass beads suspended in hydroxyl-terminated polybutadiene. One can see that the noise level is reduced and there are less artifacts from the imaging process (no visual concentric circles). However, there are still some issues with varying gray levels throughout the image, causing a fully automatic segmentation process to miss some parts of the glass beads on the boundary (see Fig. 4 at the bottom right). The Simple Region

**Fig. 4** Segmentation of Large Glass Beads



Growing algorithm, which is part of 3DSlicer was used for the segmentation in this example.

For imaging done on living subjects there are a wide variety of issues that typically make it harder to create images that are optimal for the segmentation process. Movement of the subject (e.g. due to breathing) can cause image artifacts and distortions. Some scanning methods rely on radiation (e.g. CT scans), and while for imaging purposes a high dose of radiation would be beneficial, the negative health effects require lower radiation doses that cause substantial noise and artifacts in the created images.

## ***2.2 2D Slices Plus Statistical Processing***

Electron back-scatter diffraction (EBSD) is a highly accurate method to provide a spatially resolved orientation map of a sample surface [1]. Its primary use has been focused on two-dimensional surfaces [1, 24], since the accuracy of the method is quite high when applied to a properly prepared surface. Efforts on methods to directly construct 3D representations are under consideration [17], but are not currently of high accuracy.

One method that has been developed for the automatic construction of fully three-dimensional representations is to combine statistical methods with a limited set of two-dimensional sections to construct statistically equivalent microstructures [17, 24]. The most well known tool for this type of construction is the Microstructure builder [19]. The input to the Microstructure builder consists of grain size and shape data as obtained from orthogonal images. The output is a 3-D voxel structure that matches the size and shape statistics provided as input. Microstructure builder is currently to create microstructures for (i) single-phase, equiaxised and non-equiaxised

microstructures, (ii) single-phase, variable grain shape (with certain limits), (iii) orientation distribution matching only, or both orientation and misorientation matching and (iv) two-phase with either high volume fraction of second phase particle or with smaller particles.

### 3 Construction of Geometric Model

There are a number of potential methods to go from the segmented voxel data to meshes that could be used in the simulation. However, these methods yield a fine uniform mesh that is over resolved in many areas while still providing what is typically a poor geometric representation of the material phases, which can violate known properties of the given material system. The alternative, used in this work, is to convert the image data into a multi-material non-manifold boundary representation [34] that accounts for both the specifics of the given image data and known properties of the material system.

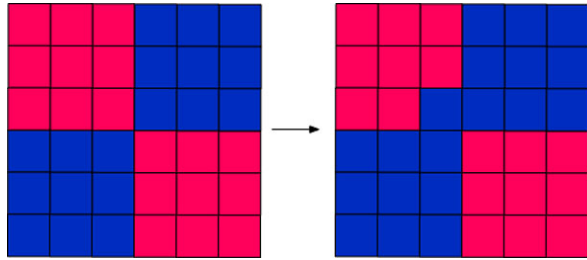
Given the finite size of the image voxels, the limited level of contrast produced, and the presence of noise in the results, any geometric representation produced from material microstructural image data is an approximation of the constituents present. The level of fidelity of the resulting representation with respect to the operations of interest, in this case the evaluation of mechanical behavior, can be further enhanced by accounting for known geometric properties of the constituents in the heterogeneous material system. Examples of properties of the material system can include: The material components (crystals or inclusions) are of a minimum size. The material components faces are flat with sharp edges or are rounded and are surrounded by filler. The interface between components will not be of dimension lower than two, or one. When known, accounting for such properties, which are material system dependent, can yield statistically more accurate geometric representations of these heterogeneous systems.

Taking account of knowledge of the accuracy limitation of the imaging modality in terms of voxel resolution relative to feature size, contrast level and/or consistency, is also important. When detected features are smaller than meaningful for the given resolution, their inclusion in the geometric model should be questioned, and in the case where those features are not consistent with known properties they should be eliminated.

#### 3.1 Voxel Data Processing

Voxel input data produced by the methods above can present situations that lead to topological inconsistencies that, in turn, prevent construction of non-manifold solid models that are physically consistent with the properties of the microstructure. The typical example is a situation where two voxels representing the same grain touch in

**Fig. 5** Elimination of corner singularity in voxel data set



such a manner that at a single common vertex or a single common edge is created between two components. Figure 5 illustrates the situation in 2D, where the voxels of the same material (indicated by the colors) touch through a single common vertex at the center. Although physically implausible with actual materials, these conditions can occur in datasets because they are a quantized representation. To eliminate these situations one of the local voxels is reassigned a different material label to correct it. For common vertices, the logic uses voting to select which voxel to reassign and what grain to reassign it to based on the neighboring voxels. The same approach is used for common-edge conditions, with a generalization allowing any one of the four voxels connected to the common edge to be selected for reassignment. In some cases, the reassignment can create another topological inconsistency in the dataset, so a multiple-pass algorithm is used to iteratively remove all of these conditions. Because these conditions are sparsely distributed, originate from inaccurate processing of the original data to begin with, and involve single-voxel modifications only, the impact on final model construction is negligible.

The process just described is the last to be executed before the non-manifold model is constructed. This is as important as other processes run on the voxel data set (e.g., small object removal, erosion/dilation) have the potential to introduce situations that would yield the generation of a non-realistic non-manifold model from the voxel data set.

As an example of the importance of processing a dataset, a synthetic volume with 5269 grains [10] was processed. In that dataset, which contains slightly more than 10 million voxels, the procedures above corrected 150,000 physically implausible conditions in order to create a valid geometric model. In some places, the voxel data exhibits a check board pattern, indicating that the quantization level was very close to the Nyquist limit for reliable reconstruction.

Another significant issue is the underlying quantization of the voxel data. In some cases, datasets contain very small individual grains on the order of 10 or 100 total voxels. Those voxel groups could have been generated for example because changes in gray level of the input image were large enough to cause the segmentation process to assume a different material at that location. Those small voxel groups lead to difficulties downstream as they are turned into small geometric objects in the non-manifold model. That not only leads to a highly refined mesh where none is needed, due to the small size of the geometric objects that need to be resolved, but often self-intersections occur in the resulting discretization. This leads to a discussion of what constitutes an acceptable quantization level and some evolution in

knowledge from recent experiences about the problem of representing microstructures with voxel data. When characterizing a specific grain boundary plane, the error can be readily quantified, and it appears that in 2-D at least ten voxels are required to capture a straight-line segment and maintain acceptable confidence that the feature is captured at the appropriate angle. If this holds for 3-D as well, a given flat grain interface should have a diameter of at least ten voxels, or an area of at least 79 square voxels. Extending this thinking to the volume, measured in terms of square voxels, we begin to lose confidence for a “spherical grain” with a diameter of less than ten voxels which translates to a volume of 523 voxels. A more liberal threshold of seven voxels/linear boundary section has also been suggested, which translates to a spherical volume of 180 voxels.

Recognizing that there is some minimum resolution at which grain geometry can no longer be accurately captured, an algorithm was developed that first filters the data to detect grains with sub-minimal volume followed by voxel reassignment by neighbor voting. For larger objects made up of hundreds of voxels, however, voxel reassignment can only be applied at the object boundaries, where there are one or more “external” voxel neighbors to use for voting. So the process works iteratively, reassigning voxels on the object boundary each pass, thereby making the object smaller and smaller, until the object is removed. The software takes a threshold value as input to specify the size of objects, in voxels, that are removed by this processing.

While the small object removal process can eliminate artifacts that represent themselves as disconnected groups of a small number of voxels making up the object, it will not remove objects that are small (on the order of one or two voxels) in one or two dimensions, but connected to a larger object. These objects significantly influence mesh generation since their small size requires extremely fine meshes so they can be resolved appropriately without causing mesh self-intersections. As they are often artifacts of the imaging and segmentation process, they don’t represent real data that needs to be resolved, and thus it is advisable to eliminate them before the non-manifold model is constructed. One method that can successfully be used for this purpose is an erosion/dilation procedure. Erosion/dilation algorithms belong to a set of algorithms developed as part of mathematical morphology [9, 32]. Morphology operations change the input voxel data set by applying a structuring element, effectively changing the shape of objects in the underlying voxel data set. In the case of erosion/dilation the erosion operation deletes voxels on the boundary of objects, while the dilation operation adds voxels. Together they have the effect to leave the boundary of larger smooth objects mostly untouched, while small outliers of one or two voxels thickness will be eliminated. The strength of the operation can be controlled by the shape and size of the structuring elements as well as how many passes the algorithms takes through the data set.

While originally developed for binary images, we have extended the algorithm to work on voxel data sets of multiple materials. Similar to the process of eliminating physically implausible situations from the voxel data set, for multi-material situations decisions have to be made during erosion to determine what material label to



give to the voxel to be eroded, and similarly during dilation. A careful implementation of the neighbor voxel voting process can make sure that a data set without any small artifacts does not get disturbed in the areas where several materials meet.

### 3.2 Construction of Non-manifold Model Topology

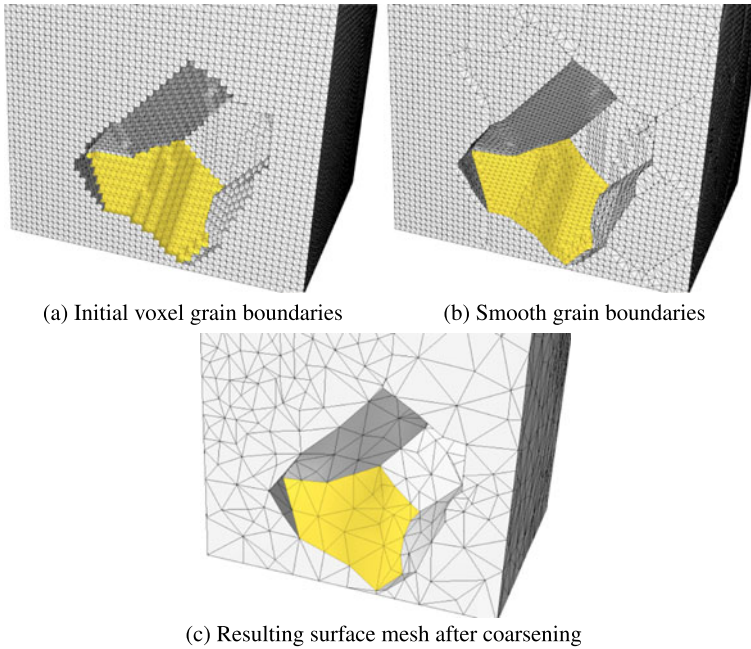
The most basic of the procedures to convert segmented voxel data to a form useful for the definition of a non-manifold geometric model is a marching cube type of procedure, originally defined for the extraction of a single triangulated isosurface [18]. Recent versions of this class of method are better able to account for information from neighboring voxels/octants to produce more controlled triangulations [13, 26] and to account for non-manifold multimaterial interaction [35]. A drawback of the marching cube type method is that the triangulations are at the resolution of the voxel/sampling size, which is typically higher than needed for a non-sampled representation, thus methods that reduce the size and complexity of these triangulations are typically applied [7, 11]. In the present work, an initial triangulation is created based on voxel level operations like those in [13, 26, 35] however, mesh coarsening is not preformed for this mesh until after the grain topology is determined and the surface triangulation is smoothed to get an accurate description of the geometry.

The process of defining the non-manifold boundary representation for a surface triangulation is focused on processes that group sets of triangles into faces and applying feature detection methods to define edges and vertices that bound faces [14, 21].

Due to the voxel nature of the initial geometry information the surface geometries created contain quantization artifacts on the scale of the individual voxels. Thus some form of surface smoothing method is needed to create more realistic shapes of the faces. Conventional data-smoothing methods, however, are not well suited to removing the quantization artifacts found in voxel datasets. Laplacian smoothing, for example, is only effective when the perturbations to be removed obey a smoothly varying statistical distribution. Volume-preserving smoothing filters [20] have been developed for medical imaging applications, however, such filters distort the surface geometry at grain/grain boundaries. Although overall grain volume is important, preserving the fidelity of the grain interfaces is much more important to obtaining the most accurate simulation results.

Therefore, a new algorithm for data smoothing designed specifically to remove quantization artifacts and recover the underlying surface geometry was developed. The algorithm is applied on a geometric face-by-face basis such that for each face the following steps are performed: (i) Calculate the surface normal at each mesh face based on the normal vector of its neighboring mesh faces. (ii) Smooth the surface normals to obtain the normals of the desired surface. (iii) Iteratively adjust the mesh vertex positions to create a surface matching the smoothed surface normals.

Figure 6 shows an example of this process. Image (a) shows the triangulation before normal smoothing and triangulation adjustment, while image (b) shows the



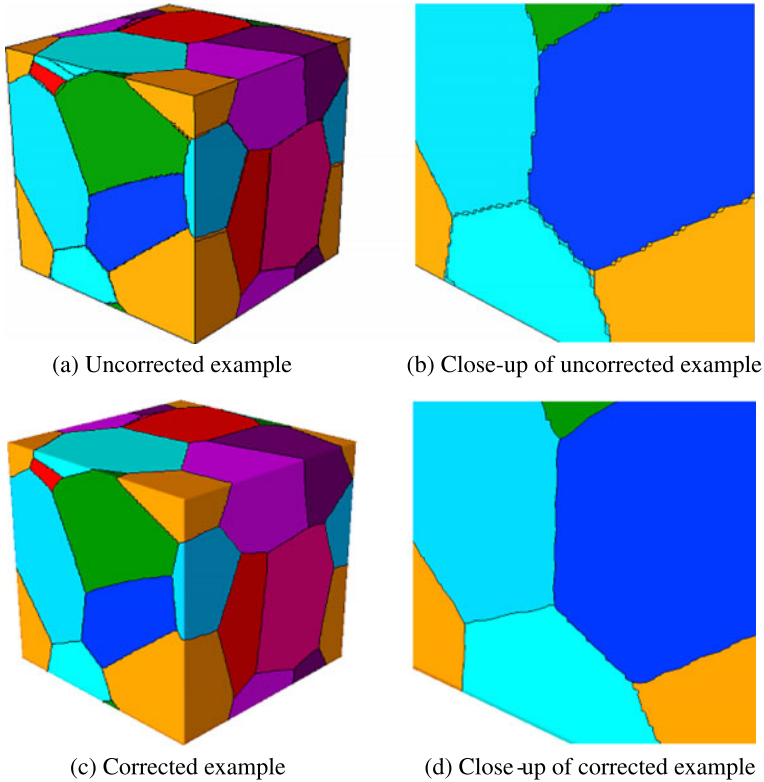
**Fig. 6** Geometry smoothing and mesh coarsening

triangulation after smoothing and triangulation adjustment. It is the surface mesh in image (b) that is then coarsened to the mesh to be used for simulations, shown in image (c) using general mesh modification operations [16].

### 3.3 *Non-manifold Models for Periodic Representative Volumes*

Tools, such as the Microstructure builder [19], that construct microstructural geometries using limited numbers of image slices, often support features to define periodic representative volume elements (RVEs). The use of such periodic RVEs is advantageous when the microstructure analysis is to calculate material properties that will be used in a macro-scale simulation. The use of periodic RVEs requires that the geometric model have matching geometry and topology on opposing sides.

When there is periodicity, the voxel description of the microstructure is such that moving to the right from the rightmost voxel you end up in a voxel that is equivalent to the leftmost voxel (as expected, this means that there is effectively a copy of the RVE repeated in space). However, this means that the boundaries of the entire RVE are not periodic in the sense that they can be a voxel off which leads to the introduction on one voxel thin sliver faces at the intersection of the RVE boundary with a face between two grains. The top two images of Fig. 7 show an example of



**Fig. 7** Elimination of boundary artifacts in RVE

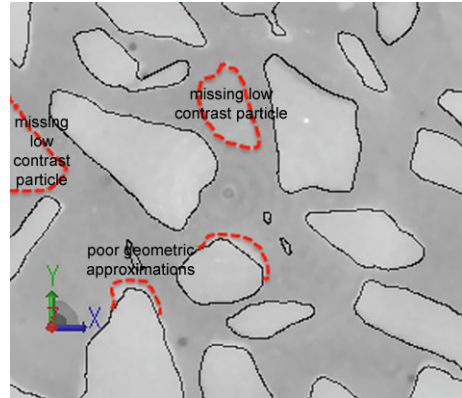
these artifacts. In downstream processes, these small model entities lead to reduced mesh quality and higher computational load in finite element simulation.

To eliminate this problem a procedure was developed that, in effect, splits the first voxel plane in half and wraps it around, so as to place the periodic surface in the middle of this voxel plane. Doing this removes grain-grain boundaries spanning the outside boundary, thereby removing these one voxel misalignments. The bottom two images of Fig. 7 show the results using this procedure.

Small features can still occur at the outside/periodic boundary, depending on how the boundary interacts with the grains. Since for periodic RVEs the location of the boundary is arbitrary, a preprocessing step is carried out that allows the positions of the periodic unit cell boundaries to be adjusted so that the least number of small features are created. This step leads to improved quality in the final mesh and reduces the computational size of the subsequent finite element simulation.

For domains that are periodic the final step in non-manifold model construction is to identify which grain corresponds to each region in the model, so that suitable material models and boundary conditions can be applied. Because the input data set is periodic, each grain can intersect the boundary of the RVE and therefore end

**Fig. 8** Example of the ability to overlay geometry on an image to identify problems and verify the results



up with portions in multiple parts of the model. These parts need to be properly matched so the same material properties are assigned to each portion of the grain. For example, a simple test case was produced by Microstructure builder comprising 14 grains. The corresponding non-manifold model produced contains 84 model regions, or an average of six model regions per grain.

The approach used to track grain assignments updates the model conversion software to tag each face (triangle) in the initial tessellation with the id numbers of the grains on either side. After the model is constructed, the software transfers these grain numbers from the mesh faces to the corresponding model faces. For model regions intersecting the RVE boundary, the grain number can be obtained from the data attached to its model faces. For internal regions, a multiple pass algorithm was implemented to derive the grain number from the data previously transferred to the model faces plus already assigned regions. Using the 14-grain, 84-region example mentioned above, all model regions were resolved in two passes, with 78 regions resolved on the first pass and the remaining 6 on the second pass.

### ***3.4 User Interface Functions to Support Image to Geometry Operations***

Even with the level of effort that has gone into the development of good algorithms for the automatic construction of the non-manifold geometric model, there are cases where the image data is not of sufficient quality for the fully automatic execution of this process. Thus a user interface is available for the user to compare image data to the extracted non-manifold model data. One specific interface function allows the user to overlay the geometry along planes representing each slice onto the original image data. Figure 8 shows an example of this functionality on a slice of data that has been partially segmented using some preliminary settings. One can see where particles are not included in the geometry and where the geometry does and does not match well with the image. In cases like that in Fig. 8, one has to ability to alter

the settings used by the segmentation procedures and can have them re-executed. While somewhat similar functionalities exist at the level of the segmentation tools to compare the segmentation with the image data, the tool here displays the intersection of the constructed geometric model with a plane at the location of the image slice. This opens up the possibility to make changes at the level of the geometry, for example through dragging control points for the surface until the geometry matches with the underlying image. Given a 3D visualization of the surface these operations can conceivably be done in three dimensional space, clearly superior to the manual editing of individual voxels for individual slices that segmentation tools provide.

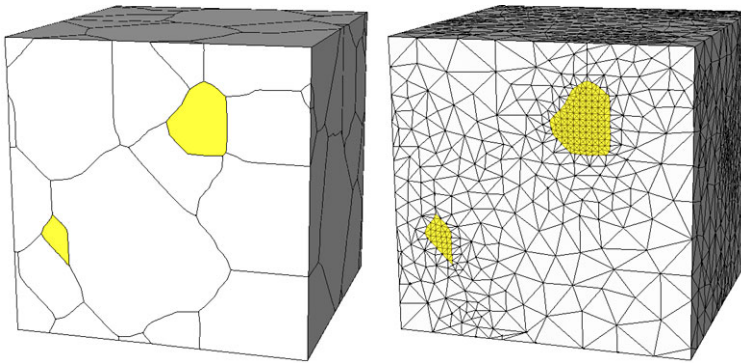
Additional functions allow one to perform local geometric modifications to the geometry by introducing additional geometric features. Boolean operations (intersections, union) are supported such that the final geometric model is a valid non-manifold model that includes the feature and can be meshed. Such tools can be applied where there are a small number of modifications needed, or where one or several items that are not part of the scan need to be inserted to evaluate its impact during simulation. Examples are individual parts like implants, or simple “what if” scenarios where the user might want to introduce e.g. a certain amount of void or other particles. Procedures to determine various geometric properties are also available. Since all geometric entities representing the particles have a known relation to the segmentation of the original scan, geometric quantities like determination of the surface area and/or volume of selected particles or voids are readily available.

## 4 Mesh Generation

Once the geometry is fully defined, generation of a mesh requires determination of appropriate mesh control information to ensure that the initial mesh has mesh sizes, configurations, and gradations appropriate for the execution of the simulations, and mesh generation algorithms to automatically create a mesh that matches the specified mesh control information.

Although there is reasonable a priori geometry-based knowledge of areas where meshes should be refined (e.g., material faces and sharp geometric features), it is often desirable to be able to adapt the mesh during the analysis process based on simulation results information obtained from the analysis steps performed to that point. Thus the mesh adaptivity procedures need to include mesh refinement and coarsening that can be driven by mesh discretization error control and can account for the influence of geometric changes including large deformations, fracture or fragmentation, or evolving contact.

The automatic mesh generator used has been designed to generate valid meshes for general non-manifold objects [27, 29] which includes multi-material geometric models with voids acting through a direct interface with the native system using the abstraction of topological entities and their adjacencies, and the modeling system libraries to support the geometric interrogations needed by the automatic mesh generator [4, 27]. The automatic mesh generation [27, 29], and adaptive mesh modification procedures [28] have been integrated with the geometric modeling kernels of



**Fig. 9** Model and mesh of a polycrystal with finer mesh in selected crystals

Siemens's NX, Dassault's CATIA V5, and PTC's Pro/Engineer CAD systems, and Spatial's ACIS, Siemens's Parasolid and PTC's Granite modeling kernels which are the basis for the majority of other CAD systems. The mesh generation and adaptation procedures also support discrete model representations such as mesh models for large deformation analyzes including multiple materials and fracture and classified voxel data for use in meshing microstructures.

A broad range of mesh control functions can be invoked for the generation of graded meshes where the mesh can be refined at critical geometric features and material interfaces and coarsened elsewhere. For example, Fig. 9 shows a mesh in which two particular grains are refined and the remaining mesh graded appropriately around those areas. The rate of gradation in mesh size can also be controlled.

Figure 10 shows some examples of the various types of mesh control available. These are close-up images of the concrete dataset showing where one of the aggregate pieces meets the boundary of the domain. These images show a range of mesh sizes applied to the entire domain and the interior faces as well as using boundary layer meshing to provide a graded mesh along the aggregate boundary. Anisotropic unstructured mesh gradation is supported when an anisotropic mesh metric field is specified. The definition of automatic procedures to set either gradations, boundary layers, anisotropic metrics or any combination of mesh specification at the material interfaces is easily supported by linking that information to the material interfaces in the non-manifold geometric model [29]. The mesh generators also include the ability to create periodic meshes from voxel data that is non-periodic. This can be used with procedures such as those in reference [15] to use non-periodic microstructural representations rather than requiring periodic unit cells.

Anisotropic mesh adaptation to account for discretization errors [16, 28] including procedures to maintain boundary layers [25], and/or evolving geometry [33] is based on local mesh modification of the model.

Figure 11 shows the overall workflow for an adaptive finite element analysis incorporating the described technologies. Note that besides the initial step of creating a discrete model from the voxel data input, the remaining part of the analysis workflow is identical to a process where the input is a CAD model. In both cases the



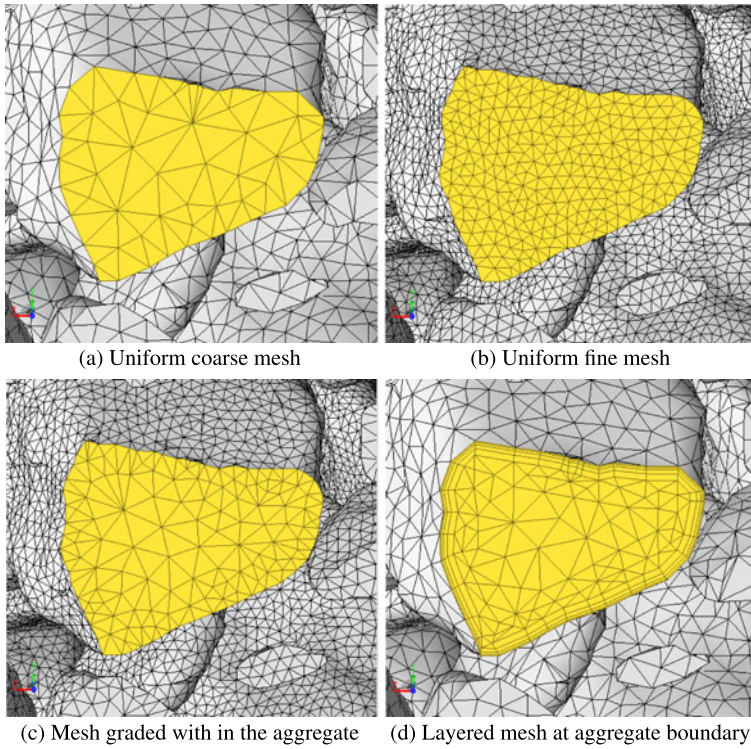


Fig. 10 Example mesh gradation for concrete data

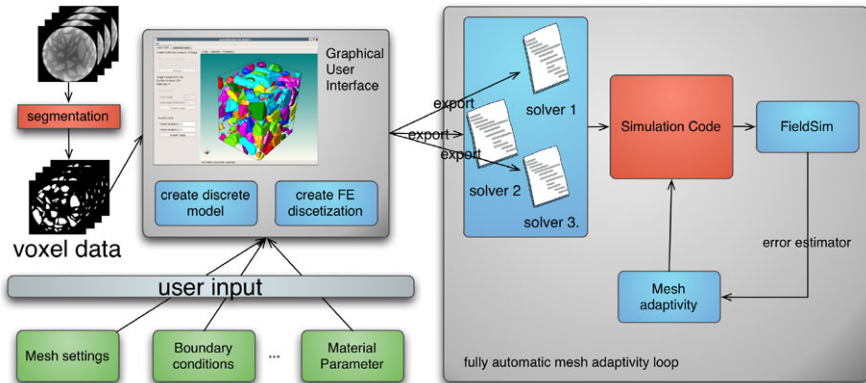
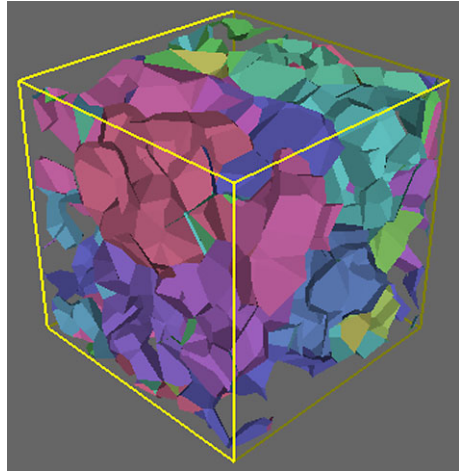


Fig. 11 Overall workflow

user will attribute the geometric model to apply non-geometric attribute information (meshing parameters, boundary conditions, material information etc.), export the problem setup to a Finite Element solver of their choice and perform simu-

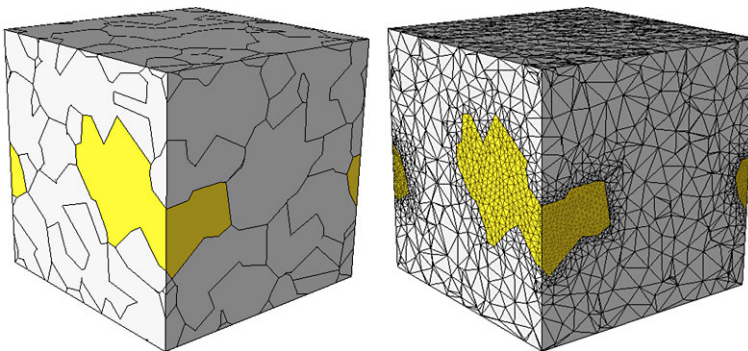
**Fig. 12** Grain data set produced by Microstructure Builder



lations followed by error estimation and the construction of an adaptively refined mesh. That loop continues until the desired solution accuracy is obtained. In the work flow shown in Fig. 11, all steps within the blue boxes indicate Simmetrix technology, green boxes indicate user input, and red boxes indicate third party software technologies.

## 5 Results

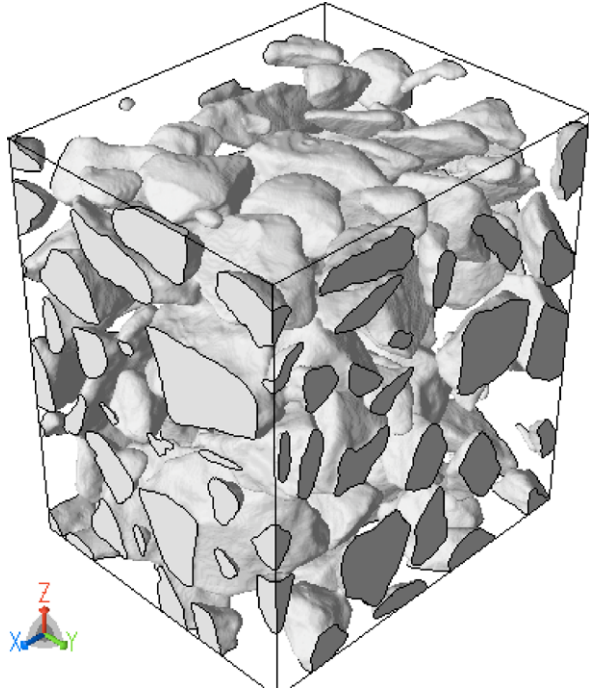
The first example demonstrates the model and mesh generation starting from a Microstructure Builder dataset consisting of 25 grains. Figure 12 shows the dataset after the grains constructed by the Microstructure Builder were clipped to a cubical domain. Figure 13 shows the model and a mesh that was generated from the model. One geometric face on the boundary of the model and the corresponding mesh were



**Fig. 13** Model and mesh generated from data set from Microstructure Builder



**Fig. 14** Non-manifold model of image data set of concrete

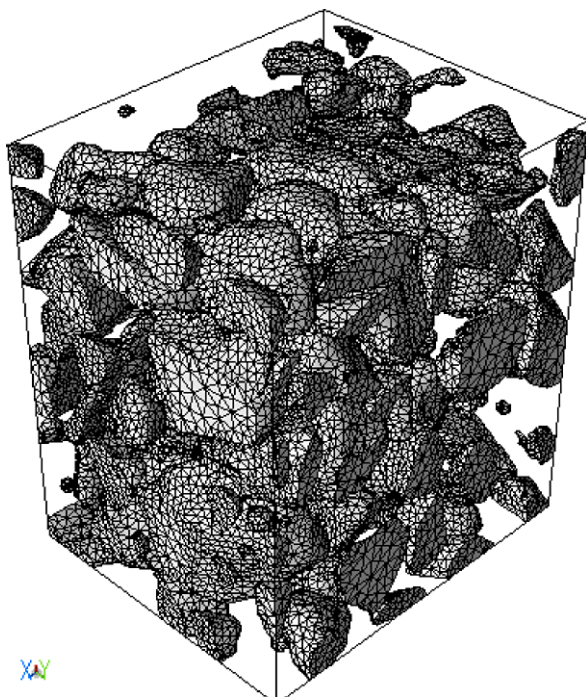


highlighted to make it easier to see the periodicity condition across the boundary of the RVE. The Microstructure Builder generates periodic grain structures, and thus the periodicity is maintained in the model and meshes generated.

The second example starts from segmented data stored in an Analyze 7.5 file format. Analyze 7.5 is a file format commonly used to store MRI data [2]. For this example, the original data was given as a set of 678 slices stored as 16 bit TIFF files of  $943 \times 943$  pixels resolution produced by a 3D scan of a cylindrical block of concrete. Figure 1 shows one slice of the input data set, from which a cubical domain was cut out from the center to avoid irregularities of the imaging data at the boundaries of the cylinder that was scanned. The segmentation of the data was performed using 3DSlicer [31]. Figure 14 shows the non-manifold model constructed for the cubical domain. At this stage no further processing has been done on the original voxel data set. In a next step the user removes the quantization artifacts using the discrete model smoothing operation described earlier by selecting the desired number of smoothing iterations (see previous section), and then proceeds to generate the finite element mesh (Fig. 15). It should be noted that the meshes shown for all examples in this paper are the surface meshes of otherwise full 3D tetrahedral meshes of the cubical domains. However, for the purpose of visualization only the surface mesh of the various geometric regions is shown.

The next image dataset was obtained from an XCMT scan of an aluminum foam sample. This dataset consists of 845 TIFF image files, each  $950 \times 950$  pixels. An initial review shows that the gray-level histogram for this data has two distinct regions,

**Fig. 15** Mesh for concrete image data



so that segmentation could be carried out using a single threshold value. Visual inspection also shows that this dataset has very high surface complexity relative to the sampling (voxel) size. Because of this the segmentation was directly used without additional processing which would have likely eliminated some of the small features, which in the case of such sharp image data do in fact exist. The model produced was a  $100 \times 100 \times 100$  subset of the aluminum foam data. Figure 16 shows the surface mesh generated including a close-up showing the level of detail contained in the model.

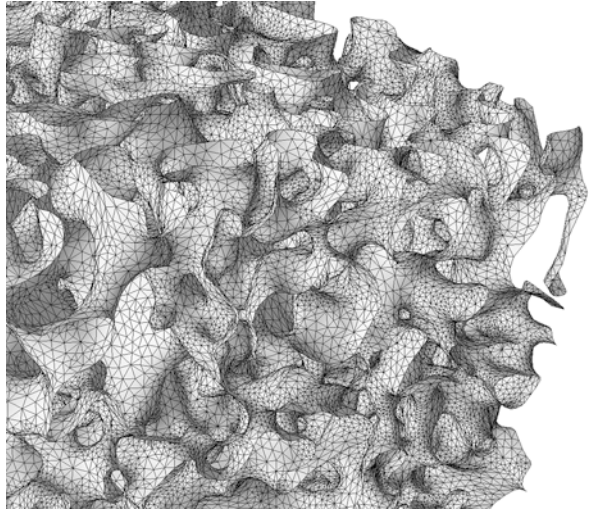
The last data set is a set of partially shattered glass beads enclosed in hydroxyl-terminated polybutadiene. The data set contained a total of 990 jpg images, each  $954 \times 915$  pixels. It was downsampled by a factor of two to a final data set consisting of 495 images, each  $477 \times 458$  pixels.

As it can be seen in Fig. 3 the contrast is very high between the glass beads and the material they were embedded in. For this example only the glass beads were segmented and it was assumed they are encapsulated in a homogenous material.

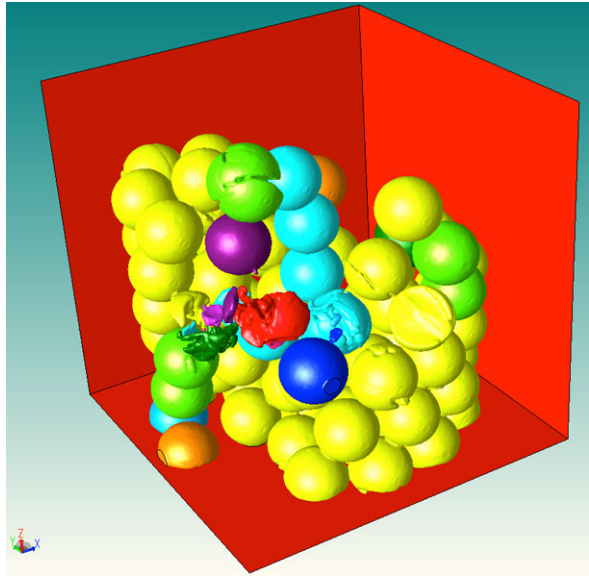
Given the high contrast of the pictures, the Simple Region Growing segmentation method available in 3DSlicer [31] was used to create the segmentation.

Figures 17 and 18 show the discrete model that was constructed from the segmentation. Note that the colors in those pictures indicate the separate regions that were detected, not the different materials found in the voxel data set (the glass beads were all segmented with just one label).

**Fig. 16** Surface mesh detail for the aluminum foam image data set

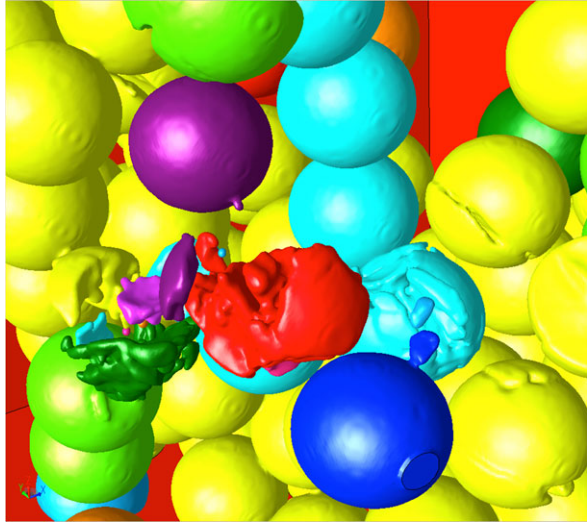


**Fig. 17** Discrete model created from Glass Beads image data set

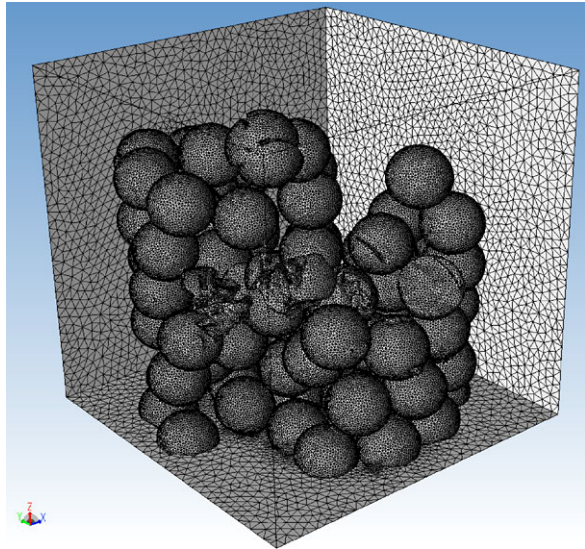


The discrete model was created by first eliminating small objects (any disconnected object with less than 180 voxels in volume), followed by an erosion/dilation step to eliminate small artifacts that are the size of a voxel in one or two dimensions. The discrete smoothing procedure described earlier was applied to the resulting discrete model to eliminate the voxel artifacts. At this point in the process the discrete model can be saved and, for the most part, treated as any other geometric model coming from CAD systems or geometric kernels for the purpose of creating simulation inputs.

**Fig. 18** Close up of discrete model create from Glass Beads image data set



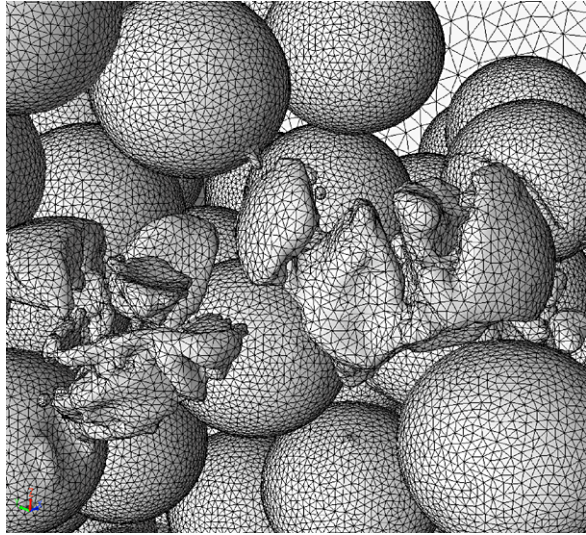
**Fig. 19** Mesh for Large Glass Beads image data set



The mesh created from the discrete model is shown in Figs. 19 and 20. Meshing attributes were chosen such that local mesh refinement are applied on the interface between the glass beads and the material they are embedded in. The difference can be seen in Fig. 19 when the mesh of the glass beads is compared to the mesh on the outer faces of the enclosing cube.



**Fig. 20** Close up of mesh for Large Glass Beads image data set



## 6 Closing Remarks

This paper has presented a set of procedures for creating simulation ready unstructured meshes from image data of the microstructure of material systems. Given the image data for a given microstructure the steps in the process include:

- Performing the segmentation of the image data to identify the material regions.
- Constructing a simulation appropriate non-manifold geometric model, including the application of automated procedures to correct the non-manifold model as needed to properly represent the material system and to make the model more appropriate for the creation of meshes to be used in simulations. There are also options for the user to interact with the non-manifold model and original image data to compare them and modify the non-manifold model if desired.
- Automatic and/or user controlled specification of mesh control information as desired for the simulations to be performed. This is followed by fully automatic mesh generation.
- Execution of the desired simulation, which can include the application of adaptive mesh control (not covered in the current paper).

The process of constructing a proper non-manifold geometric model from the image data is a critical technical step in this process. Image data segmentation and the interactions of the resulting data with the procedures to construct the non-manifold geometric model is the most complex step in the process and the one that can benefit the most from the continued development of improved methods.

**Acknowledgements** The research reported in this document was performed in connection with contract W911QX-06-C-0040 with the U.S. Army Research Laboratory and contract FA8651-11-C-0200 with the Air Force Research Laboratory (AFRL).

## References

1. Adams BL, Wright SI, Kunze K (1993) Orientation imaging—the emergence of a new microscopy. *Metall Trans A* 24:819–831
2. Analyze file format. <http://www.grahamwideman.com/gw/brain/analyze/formatdoc.htm>
3. Bankman I (ed) (2008) *Handbook of medical image processing and analysis*, 2nd edn. Elsevier, Amsterdam
4. Beall MW, Walsh J, Shephard MS (2004) A comparison of techniques for geometry access related to mesh generation. *Eng Comput* 20(3):210–221
5. Data set retrieved from <http://www.lungcanceralliance.org>
6. Flannery BP, Deckman HW, Roberge WG, D'Amico KL (1987) Three-dimensional X-ray microtomography. *Science* 237(4821):1439–1444
7. Garland M, Heckbert P (1997) Surface simplification using quadric error metrics. In: *Proceedings of ACM SIGGRAPH*, pp 209–216
8. Gering D, Nabavi A, Kikinis R, Grimson W, Hata N, Everett P, Jolesz F, Wells W (1999) An integrated visualization system for surgical planning and guidance using image fusion and interventional imaging. In: *International conference on medical image computing and computer assisted intervention*, pp 809–819
9. Gil JY, Kimmel R (2002) Efficient dilation, erosion, opening, and closing algorithms. *IEEE Trans Pattern Anal Mach Intell* 24(12):1606–1617
10. Groeber M, Ghosh S, Uchic MD, Dimidukc DM (2008) A framework for automated analysis and simulation of 3D polycrystalline microstructures. Part 1: statistical characterization. *Acta Mater* 56(6):1257–1273
11. Hoppe H (1996) Progressive meshes. In: *Proceedings of ACM SIGGRAPH*, pp 99–108
12. Ibanez L, Schroeder W, Ng L, Cates J (2005) *The ITK software guide: the insight segmentation and registration toolkit (version 1.4)*. Kitware, Inc., Clifton Park
13. Ju T, Losasso F, Schaefer S, Warren J (2002) Dual contouring of hermite data. In: *Proceedings of SIGGRAPH*, pp 339–346
14. Krysl P, Ortiz M (2001) Extraction of boundary representation from surface triangulations. *Int J Numer Methods Eng* 50:1737–1758
15. Kumar RS, Wang AJ, McDowell DL (2006) Effects of microstructure variability on intrinsic fatigue resistance of nickel-base superalloys—a computational mechanics approach. *Int J Fract* 137:173–210
16. Li X, Shephard MS, Beall MW (2005) 3-D anisotropic mesh adaptation by mesh modifications. *Comput Methods Appl Mech Eng* 194(48–49):4915–4950
17. Lin FX, Godfrey A, Juul Jensen D, Winther G (2010) 3D EBSD characterization of deformation structures in commercial purity aluminum. *Mater Charact* 61:1203–1210
18. Lorensen WE, Cline HE (1987) Marching cubes: a high resolution 3D surface construction algorithm. In: *Proceedings of SIGGRAPH*, pp 163–169
19. Microstructural Builder web page. <http://code.google.com/p/mbuilder>
20. Moore RH, Rohrer GS, Saigal S (2009) Reconstruction and simplification of high-quality multiple-region models from planar sections. *Eng Comput* 25(3):221–235
21. Owen SJ, White DR (2003) Mesh-based geometry. *Int J Numer Methods Eng* 58:375–395
22. Pieper S, Halle M, Kikinis R (2004) 3D Slicer. In: *Proceedings of the 1st IEEE international symposium on biomedical imaging: from nano to macro*, vol 1, pp 632–635
23. Pieper S, Lorensen B, Schroeder W, Kikinis R (2006) The NA-MIC Kit: ITK, VTK, pipelines, grids and 3D slicer as an open platform for the medical image computing community. In: *Proceedings of the 3rd IEEE international symposium on biomedical imaging: from nano to macro*, vol 1, pp 698–701
24. Rollett AD, Lee S-B, Campman R, Rohrer GS (2007) Three-dimensional characterization of microstructure by electron back-scatter diffraction. *Annu Rev Mater Res* 37:627–658
25. Sahni O, Jansen KE, Shephard MS, Taylor CA, Beall MW (2008) Adaptive boundary layer meshing for viscous flow simulations. *Eng Comput* 24(3):267–285

26. Schaefer S, Ju T, Warren J (2007) Manifold dual contouring. *IEEE Trans Vis Comput Graph* 13(3):610–619
27. Shephard MS (2000) Meshing environment for geometry-based analysis. *Int J Numer Methods Eng* 47(1–3):169–190
28. Shephard MS, Flaherty JE, Jansen KE, Li X, Luo X-J, Chevaugnon N, Remacle J-F, Beall MW, O'Bara RM (2005) Adaptive mesh generation for curved domains. *Appl Numer Math* 52(2–3):251–271
29. Simmetrix, Inc. <http://www.simmetrix.com/>
30. Simple Region Growing webpage. [http://www.slicer.org/slicerWiki/index.php/Modules:Simple\\_Region\\_Growing-Documentation-3.4](http://www.slicer.org/slicerWiki/index.php/Modules:Simple_Region_Growing-Documentation-3.4)
31. Slicer web page. <http://www.slicer.org>
32. Verdu-Monedero R, Angulo J, Serra J (2011) Anisotropic morphological filters with spatially-variant structuring elements based on image-dependent gradient fields. *IEEE Trans Image Process* 20(1):200–212
33. Wan J, Kocak S, Shephard MS (2005) Automated adaptive 3-D forming simulation process. *Eng Comput* 21(1):47–75
34. Weiler K (1988) The radial edge structure: a topological representation for non-manifold geometric modeling. In: *Geometric modeling for CAD applications*, pp 3–36
35. Zhang Y, Hughes TJR, Bajaj CL (2010) An automatic 3D mesh generation method for domains with multiple materials. *Comput Methods Appl Mech Eng* 199(5–8):405–415