Feature-Preserving, Multi-Material Mesh Generation using Hierarchical Oracles

Max Kahnt, Heiko Ramm, Hans Lamecker, and Stefan Zachow

Medical Planning Group, Zuse Institute Berlin,
Takustraße 7, D-14195 Berlin-Dahlem, Germany
<lastname>@zib.de
http://www.zib.de/en/visual/medical-planning.html

Abstract. This paper presents a method for meshing multi-material domains with additional features curves. This requirement arises for instance in situations where smooth objects (e.g. anatomical structures) are combined with technical objects (e.g. implants, surgical screws). Our approach avoids the tedious process of generating a single consistent input surface by means of an implicit representation, called *oracle*. Input features are preserved in the output mesh and termination of the algorithm is proved for certain conditions. We show that our method provides good element quality while at the same time keeping the number of elements in the output mesh low.

1 Introduction

Problems that can be described by partial differential equations (PDEs) are generally solved by finite element (FE) methods. Due to their geometric flexibility, tetrahedral FE meshes (tetmeshes) are well suited to discretise complex shapes. A challenge lies in generating tetmeshes with preferably few elements to keep the complexity of the FE computations low, while accurately approximating a given geometry. Concurrently, the elements must be of good shape in order to ensure stability [14]. Satisfying both criteria at a time is generally not possible, hence a compromise must be found.

In this paper we focus on meshing of multi-material domains that represent a combination of both smooth and non-smooth geometries. Applications comprise the combination of anatomical and mechanical structures like bone/implant compounds, e.g. to predict implant wear [10], or the necessity that feature curves are represented explicitly. In such scenarios, an accurate representation of the material interfaces and the non-smooth regions (sharp edges) is desirable.

Usually, the geometry of anatomical structures is derived via image segmentation as in Zachow et al. [16], while mechanical parts are represented analytically, e.g. as CAD models. In a first step, a boundary representation of the multi-material compound is generated, which is then “stuffed” with tetrahedra. This typically requires an explicit fusion of the objects to be discretised before meshing, for instance using a single voxel- or surface-representation. Inconsistencies resulting from object overlaps are resolved easily within a voxel grid like used by Zhang et al. [17]. But the resulting mesh typically suffers from artificially introduced inaccuracies, for instance lost information about sharp edges. Alternatively, a consistent surface triangulation that separates the different objects (called domains) can be computed. Regions where boundary intersections introduce small angles or narrow inter-boundary distances occur exhibit small or badly shaped triangles which are not suited for generating useful FE meshes.

An alternative approach, which does not require an explicit boundary representation, was introduced by Oudot et al. [11] and Pons et al. [12]. This Delaunay-based method requires only the implementation of a so-called "oracle" which returns the following information: (1) the material a point belongs to, (2) one intersection of a line segment with a boundary, if it exists. However, for geometries with non-smooth boundaries (sharp edges), this approach tends to introduce many small elements while still not recovering the feature curves adequately.

Boltcheva et al. [4] address feature curves that occur at the junctions of 3 or more materials in segmented image data. They sample them a-priori according to a user-given density parameter. Avoiding any point insertion too close to these samples, the junctions are preserved throughout the mesh generation process. In their approach, the point density along the feature curves is determined at the beginning of the algorithm which opposes the self-adjusting criteria-driven Delaunay refinement.

Very recently, Dey et al. [6] propose a method that meshes piecewise-smooth complexes approximately defined by multi-label datasets. They provide a method to extract the feature curves. These are incorporated in the mesh generation process as a set of line segments protected by spheres eventually to be refined on encroachment.

The resulting mesh quality of meshes generated by Delaunay refinement schemes can be improved in terms of dihedral angle. Foteinos and Chrisochoides [8] employ a refinement criterion that heuristically removes most slivers. Cheng et al. [5] and Edelsbrunner et al. [7] propose methods to post-process such meshes.

We propose a method to extend the oracle-based approach by Oudot et al. [11] to preserve a user-given set of line segments. We prove that our algorithm terminates while preserving any set of line segments not exhibiting angles below 60°. Our implementation provides a hierarchical oracle that allows for an intuitive setup if simple Boolean operations describe the mutual relations of separate input domains. As each input domain is handled transparently through the oracle, they can be of distinct type, e.g. voxel representation or triangular surface, and an explicit fusion step can be omitted. A study on bone/implant compounds demonstrates that our method simplifies the mesh generation process and provides high quality FE meshes.

2 Preserving 1D feature lines

Delaunay triangulations of an ε-sample have been shown to appropriately approximate smooth surfaces, both geometrically and topologically. Boissonnat and

Oudot [3] use these results to construct a surface mesh generator with provable properties. From an initial point set, successive refinements lead towards a reliable surface approximation. They point out that their method is general enough to handle various kinds of input, e.g. analytical implicit surface descriptions, level sets in 3D images, point set surfaces and polyhedra, as long as some simple properties can be fulfilled to build up *an oracle* and provide an appropriate sizing field. Oudot et al. [11] extend the method to the volumetric case. They add higher dimensional refinement rules to the surface meshing method. The resulting algorithm fully contains the surface mesh generation method and consecutively addresses the volumetric cells. Eventually the volumetric algorithm falls back on the surface refinement rules. Finally, Pons et al. [12] propose an approach for oracle-based multi-material volume meshing. A material is assigned to each Delaunay tetrahedron according to the corresponding Voronoi vertex, i.e. the tetrahedron circumcenter. Moreover, they propose parameters to tune element size and shape criteria globally or per material to simplify the application of their algorithm.

Our method extends the set of rules proposed by Oudot et al. [11] to preserve a set of constrained line segments, in the following called segments. The segments are maintained in the conforming sense, i.e. if the algorithm attempts to insert a point that conflicts with a constrained segment, it is split beforehand. Oudot et al. [11] append a meshing layer to the algorithm of Boissonnat and Oudout [3] to tackle the volumetric elements. Analogously, we prepend a meshing layer to the method of Pons et al. [12] that handles the constrained segments. Both, the surface and the volumetric layer of the algorithm are modified to eventually fall back on the constrained segment refinement rules. The geometric and topological guarantees given by a dense sampling of the material interfaces hold for smooth surfaces, see Amenta and Bern [2]. While this does not necessarily conflict with our extension, its application admittedly is most interesting at non-smoothnesses. We provide the application to non-smooth geometries in a similar manner as already is done for the original algorithm: For instance, the non-smoothness of polyhedral input is hidden within the oracle.

2.1 Strategy to preserve constrained segments in the oracle method

Let E a set of segments to be preserved with P_E the set of their endpoints. A segment s is called *constrained* if it is a segment to be preserved, i.e. iff $s \in E$. Note that the set E is modified during the course of the algorithm when a segment is split. Point insertions triggered by the necessity to split a constrained segment remove the respective segment $s = (a, b)$ with endpoints a, b and splitting point m from E. As a replacement the subsegments (a, m) and (m, b) resulting from the split are inserted into E. Their preservation, resp. the preservation of their subsegments, recursively guarantees the preservation of s in the conforming sense. We will denote $E_0 = E$ the initial set and E_i the result of the i-th split of a segment.

A segment s with endpoints a, b and midpoint m is said to be *encroached* by a point p if $\|p - m\| < \|a - m\|$, i.e. the point is in the open diameter ball of

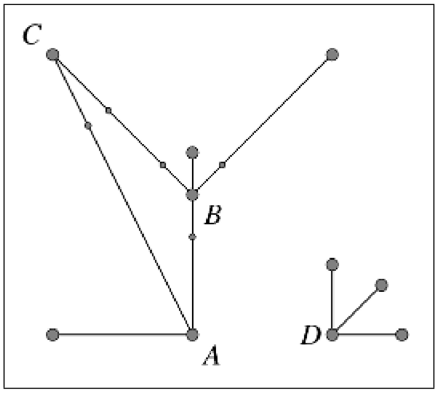


Fig. 1. The type of star vertices. Small points are added later. A, B, C, D are star vertices because they all have at least two adjacent points in the initial set of endpoints. The other points are non-star vertices.

the segment. This is a common notion when dealing with conforming triangulations [13]. This encroachment definition differs from the encroachment notions for tetrahedra and Delaunay facets, because it is triggered also on point insertions not necessarily extinguishing the segment from the Delaunay triangulation. The converse is true though: If there is no point within the diametral sphere, the corresponding segment is a simplex of the Delaunay triangulation.

The vertices $p_i \in P_E$ that are endpoints of at least two constrained segments on initialization, are called *star vertices*. It is a property of the input set of vertices, i.e. no refinement point added has this property and no *star vertex* can lose it during the course of the algorithm, For each *star vertex* $p_i \in P$ let $l_{\min,i}$ the length of its shortest incident constrained segment in E. A constrained segment is called *bad*, if (1) it is incident to a star vertex p_i and has length larger than $\frac{l_{\min,i}}{3}$, (2) it is incident to a star vertex and is longer than some other segment incident to the same star vertex, or (3) if it is encroached by some point.

We chose $\frac{l_{\min,i}}{3}$ heuristically, assuring that the initial splits do not create arbitrarily small subsegments. For an initial constrained segment s that is incident to two star vertices, the medial subsegment resulting from the splits at each of its ends according to (1) is not the shortest subsegment of s. We maintain the notation of Oudot et al. [11] where σ specifies the sizing field, α the angle bound of a facet and B the minimal radius-edge ratio.

2.2 Algorithm preserving constrained segments

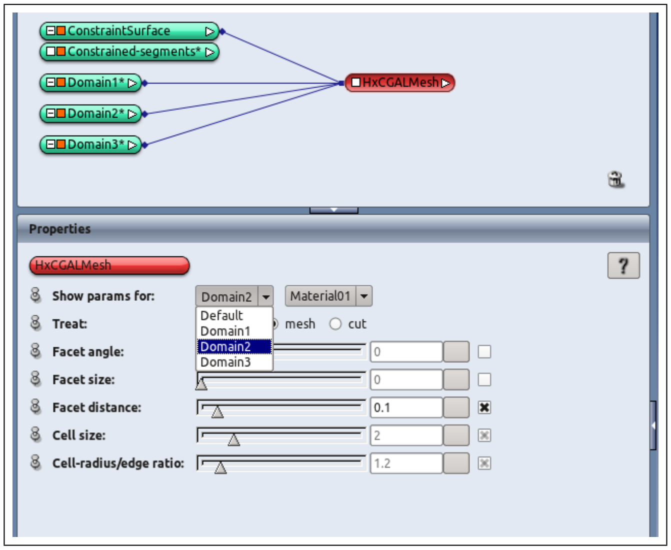
The first part of the algorithm aims at recovering the constrained segments in the Delaunay triangulation of the initial point set. To avoid non-terminating recursive insertions around star vertices, incident segments are trimmed to a uniform length on their first refinement triggering.

Rule set for algorithm preserving constrained segments

```
T1 If a constrained segment $e$ is bad
    T1.1 if $e$ is incident to a star vertex $p_i$ and has length $> \frac{l_{\min,i}}{3}$, insert the point
        at distance $\frac{l_{\min,i}}{3}$ from $p_i$ on e
    T1.2 else insert the midpoint of $e$
T2 If a facet $f$ does not have its three vertices on $\partial O$ or has a surface Delaunay ball
    $B(c,r)$ with ratio $\frac{r}{\sigma(c)} > \alpha$, then:
    T2.1 if $c$ is included in a segment diameter ball $B(c',r')$, insert $c'$
    T2.2 else insert $c$
T3 If a tetrahedron $t$ with circumcentre $c$ has a circumradius $r$ greater than $\sigma(c)$ or
    radius-edge ratio $\frac{r}{l_{\min}}$ greater than $B$, then
    T3.1 if $c$ is included in a segment diameter ball $B(c'',r'')$, insert its center $c''$
    T3.2 else if $c$ is included in a surface Delaunay ball $B(c',r')$ and $c'$ is included in
        a segment diameter ball $B(c'',r'')$, insert $c''$
    T3.3 else if $c$ is included in a surface Delaunay ball $B(c',r')$ insert $c'$
    T3.4 else insert $c$
```

Rules T2 and T3 are analogues to the meshing rules proposed by Oudot et al. [11]. They only differ in the fallback strategy eventually refining constrained segments on encroachment. This strategy assures the preservation of the constrained segments since they cannot be encroached by a point actually inserted into the triangulation whenever T1 is fulfilled.

We explicitly prove termination of the algorithm for a decomposed version of the Rule Set T:
The initial conformation step is independent of the actual geometry a mesh is to be generated for. After a finite number of point insertions all constrained segments are guaranteed to not being encroached in the Delaunay triangulation of the successively built point set.
In a second step, we generalise the termination proof of Oudot et al. [11] to our extended method. We follow the common idea of deriving an insertion radius lower bounding the mutual distance of any two points in the resulting mesh. In this context, lower bounds on the applicable parameters are deduced. Our proofs point out the existence of finite values for the minimal facet angle bound α and the minimal radius-edge ratio bound B, depending on the configuration of the constrained segments and the sizing field σ. Our proofs only apply to constrained segment configurations not exhibiting angles below $\frac{\pi}{3}$. There are no general negative proofs on termination though, hence motivating the use of the method also for a more practical choice of parameters.



Show params for: Domain2 Material01

Treat: ● mesh ○ cut

Facet angle: 0 □

Facet size: 0 □

Facet distance: 0.1 ☒

Cell size: 2 ☒

Cell-radius/edge ratio: 1.2 ☒

Fig. 2. Mesh generator module in ZIBAmira. Inputs (top): The separate domains represent distinct objects and a mesh is required for their fusion. They are connected sequentially to the mesh generator according their priority at regions of mutual overlap. Additionally a set of constrained segments is connected to the meshing module. In its current implementation, this set is a set of line segments constituting a piecewise linear surface path defined on the *ConstraintSurface* wich does not serve any further purposes; Parameters (bottom): quality criteria can be bounded by constants, defined separately for each material. It is possible to treat a material as a *cut geometry*: all points within this material are handled as points outside of any domain with same or lower priority.

3 Implementation

An implementation of Oudot et al. [11] and Pons et al. [12] is available in CGAL since version 3.5. We integrated CGAL as a module into ZIBAmira-2012.03 [15] (`http://amira.zib.de`). Our current implementation is based on CGAL 3.7 [1].

3.1 Hierarchical Oracle

An oracle has been implemented that defines the domain to be meshed. It allows to implicitly perform Boolean operations, because the mesh generation algorithm itself does only require the two kinds of query operations introduced by Boissonnat and Oudot [3]. The hierarchical oracle essentially maintains a priority list of input domains, such that point queries are passed to lower priority domains if they are outside all higher prioritized inputs. That way the oracle is a black box that performs the Boolean operations on the input domains without explicitly computing the intersections and implicitly resolving inconsistencies in data type and shape. The data types that can currently be handled are (1) watertight triangular surface meshes with exactly two triangles joining at an edge, (2) labeled uniform voxel grids, and (3) implicit spheres given by their resp. midpoint and radius.

The features that are to be preserved are defined by a connected set of piecewise linear segments. In ZIBAmira these can be defined as surface path sets on triangulated surfaces. If specific points of the domain boundary are known a-priori these can be added to the initial point set in order to speed up the mesh generation process or simply to preserve them. The ZIBAmira module offers a port to connect a geometry containing these points.

4 Experiments

We tested our method in a simulation study concerning the computation of strains occurring in bone/implant compounds. Galloway et al. [9] generated FE meshes for implanted tibiae (see Figure 3). We randomly selected one hundred virtual total knee replacement settings from that study, including geometric representations of the tibia and the tibial component. For each implant, the region where protruding bone has been removed was given and sharp edges were marked as line segments to be preserved. The one hundred datasets were then meshed with our implementation and the method of Pons et al. [12] (available in CGAL 3.7). For both methods the desired quality criteria were determined heuristically and chosen as follows: (1) maximal radius-edge ratio of 1.1 for all materials, (2) maximal circumradius of 1 mm for implant and 3 mm for bone, and (3) maximal facet distance of 0.1 mm to approximate the surfaces for implant and 2 mm for bone material. Despite lacking the theoretical guarantee of termination for this choice of parameters, all models were generated successfully without further tweaking. The results of both approaches were post-processed by the sliver removal methods available in CGAL 3.7, see [5], [7].

The tetrahedral meshes generated with our implementation met the quality criteria with 437,000 tetrahedra on average (range 295,000 to 707,992). The number of tetrahedra generated with Pons approach was significantly larger with an average number of 900,898 tetrahedra per mesh (range 614,234 to 1,533,011). For the meshes generated by our method, the average minimal dihedral angle was $8.72^\circ (\pm 1.31^\circ)$ after post-processing.

5 Discussion

As shown in the experiments, our method generates fewer elements than the method by Pons et al. with the same quality criteria applied. We attribute this to the fact that for the original method geometric accuracy is achieved by unguided refinement of the mesh in areas where the quality criteria are not met. Our method specifically aims at reconstructing the requested features first due to a more targeted point insertion during the refinement process in those areas (see fig. 5).

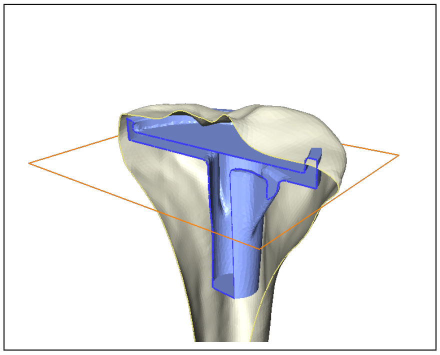


Fig. 3. Tibia implantation scenario. Above the orange rectangle the protruding bone shall be removed. Implant, cutting plane and bone enter the hierarchical oracle.

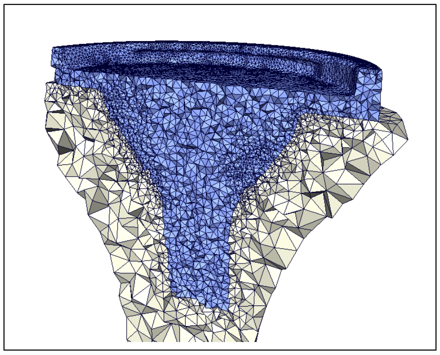


Fig. 4. Resulting mesh. High geometric accuracy at implant boundary and sharp features. Mesh size grades down where elements are far from the implant-bone material interface.

Termination of the algorithm has not been proven for arbitrary constrained segment configurations. The algorithm terminates if the minimal angle in the set of constrained segments α_0 is greater or equal to $\frac{\pi}{3}$ and parameters α, B have been chosen appropriately. Conversely our algorithm does not terminate if the specified quality parameter α, B directly contradict the preservation of the constrained segments, e.g. $B < \frac{1}{2\sin\alpha_0}$ where α_0 is the minimal angle in E. The termination proof in its current form does not yield lower bounds for α, B in case α_0 is too small.

If the surface is not sampled densely enough, constrained segments might not be present in the resulting mesh. They are guaranteed by our algorithm to be not encroached and hence are represented in the final Delaunay triangulation. But the resulting mesh is the union of all Delaunay cells restricted to the materials and an insufficiently dense surface sampling might cause a lack of surface elements also involving the local segments. This problem is more general and not exclusively related to constrained segments. But their explicit definition offers a way to develop further refinement criteria depending on their actual presence in the resulting mesh (i.e. we call a segment bad if it is not in the resulting mesh, or equivalently, if there is no incident tetrahedron assigned to a material).

A set of constrained segments is not suited to accurately represent smooth feature curves. In practice they are more likely to be approximations of the feature curves. If features are explicitly tagged but are not close enough to the material interface provided through the oracle w.r.t. the desired or self-induced element sizing, then they eventually are not part of the discretised boundary. In such settings the inaccurate constrained segments are inconsistent to the "oracled" surface.

Our approach pushes off the problem of localizing the features as we do not restrict to a certain input type. Moreover feature curves might address distinct

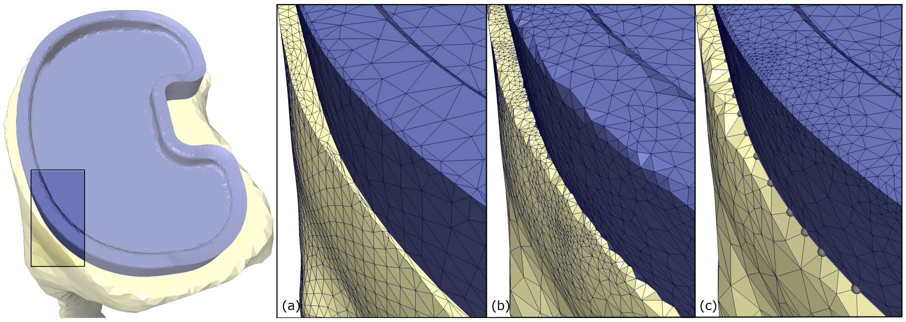


Fig. 5. (a) The advancing-front approach preserves the surface triangulation, which encloses the volumes to fill with tetrahedra. It generates distorted elements in regions where implant and bone surface leave small gaps. (b) The Pons approach inserts many elements in the vicinity of these gaps to accurately recover the geometry of the implant. With the radius-edge ratio specified, that even leads to smaller elements on the bone surface. (c) These problems do not occur with the *guided* point insertion in our method.

properties of the mesh. Among them are sharp edges (where small dihedral angles are ultimately introduced), multi- material interfaces where 3 or more materials meet and multi-material interfaces resulting from the implicit fusion process. Our approach only covers some of these cases as described above.

In the original oracle-based method, non-smoothness is hidden within the oracle e.g. when meshing polyhedral domains. It does not disturb the mesh generator if some quantities can be estimated/faked appropriately. If sharp features are given explicitly, the dihedral angles at these junctions might jeopardise the termination. We did not investigate such effects.

Our proof of termination is not optimal w.r.t. the lower bounds on the parameters α and B. Furthermore post-processing has not been adjusted to respect the constrained segments.

Lastly, the advantages of the preservation of features and the effortless fusion of different domains through the hierarchical oracle can only be used in combination if the features are not a result of the fusion process itself, i.e. they must be known a-priori.

6 Conclusion

In this work we extend the oracle-based meshing approach. It is based on an abstract domain representation and does not require time-consuming preprocessing of the input data. We coupled the ZIBAmira software with an extension of the oracle-based meshing approach. ZIBAmira allows to load, align and visualise the data describing the domains to generate a mesh for. Feature curves can be defined as so-called path sets. The hierarchical oracle currently allows triangle meshes, label images and implicit spheres as input. The overlaps and mutual inconsistencies of the data can be solved without the need to explicitly compute a consistent representation. The hierarchical oracle overrides the tedious and error-prone process deriving a single multi-material surface or comparable. Quality parameters such as element size and shape bounds can be applied material-wise. The resulting mesh will respect the desired criteria and additionally preserve the feature curves as prescribed.

Future work will be directed towards generalizing constraint configurations and finding improved termination bounds. Smooth feature curves shall be incorporated in the meshing approach. It would be nice to have an algorithm extracting the features with the current oracle methods only or with a new oracle assumption as simple as the existing. This would eventually enable to specifically address features that evolve from fusion implicitly. The recent method by Dey et al. [6] seems to be advantageous to our approach but also does not solve this issue that comes up when using our hierarchical oracle.

Acknowledgement. We thank the anonymous referees for their comments that helped improve the presentation of the paper. Heiko Ramm is supported by the EU Project MXL. Hans Lamecker is supported by the DFG-Matheon Project F2.

References

1. Alliez, P., Rineau, L., Tayeb, S., Tournois, J., Yvinec, M.: 3D mesh generation. In: CGAL User and Reference Manual. CGAL Editorial Board, 3.7 edn. (2010), http://www.cgal.org/Manual/3.7/doc_html/cgal_manual/packages.html#Pkg Mesh_3
2. Amenta, N., Bern, M.: Surface reconstruction by Voronoi filtering. In: SCG '98: Proceedings of the fourteenth annual symposium on Computational geometry. pp. 39–48. ACM, New York, NY, USA (1998)
3. Boissonnat, J., Oudot, S.: Provably good sampling and meshing of surfaces. Graphical Models 67(5), 405–451 (2005)
4. Boltcheva, D., Yvinec, M., Boissonnat, J.: Mesh generation from 3d multi-material images. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009 pp. 283–290 (2009)
5. Cheng, S., Dey, T., Edelsbrunner, H., Facello, M., Teng, S.: Silver exudation. Journal of the ACM (JACM) 47(5), 883–904 (2000)
6. Dey, T., Janoos, F., Levine, J.: Meshing interfaces of multi-label data with delaunay refinement. Engineering with Computers pp. 1–12 (2012)
7. Edelsbrunner, H., Li, X., Miller, G., Stathopoulos, A., Talmor, D., Teng, S., Üngör, A., Walkington, N.: Smoothing and cleaning up slivers. In: Proceedings of the thirty-second annual ACM symposium on Theory of computing. pp. 273–277. ACM (2000)
8. Foteinos, P., Chrisochoides, N.: High-quality multi-tissue mesh generation for finite element analysis. In: MeshMed, Workshop on Mesh Processing in Medical Image Analysis (MICCAI). pp. 18–28 (September 2011)
9. Galloway, F., Kahnt, M., Seim, H., Nair, P.B., Worsley, P., Taylor, M.: A large scale finite element study of an osseointegrated cementless tibial tray. In: 23th Annual Symposium International Society for Technology in Arthroplasty (2010)
10. Ong, K., Kurtz, S.: The use of modelling to predict implant behaviour. Medical device technology 19(5), 64–6 (Sep 2008)

11. Oudot, S., Rineau, L., Yvinec, M.: Meshing volumes bounded by smooth surfaces. In: Proceedings of the 14th International Meshing Roundtable. pp. 203–219. Springer (2005)
12. Pons, J., Ségonne, F., Boissonnat, J., Rineau, L., Yvinec, M., Keriven, R.: High-quality consistent meshing of multi-label datasets. In: Information Processing in Medical Imaging. pp. 198–210. Springer (2007)
13. Shewchuk, J.: Mesh generation for domains with small angles. In: Proceedings of the sixteenth annual symposium on Computational geometry. pp. 1–10. ACM (2000)
14. Shewchuk, J.: What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint). University of California at Berkeley (2002)
15. Stalling, D., Westerhoff, M., Hege, H.C.: Amira: A highly interactive system for visual data analysis. In: Hansen, C.D., Johnson, C.R. (eds.) The Visualization Handbook, pp. 749 – 767. Elsevier (2005)
16. Zachow, S., Zilske, M., Hege, H.: 3D reconstruction of individual anatomy from medical image data: Segmentation and geometry processing. Proceedings of 25. ANSYS Conference & CADFEM Users' Meeting, Dresden (2007)
17. Zhang, Y., Hughes, T.J.R., Bajaj, C.L.: An automatic 3D mesh generation method for domains with multiple materials. Computer Methods in Applied Mechanics and Engineering (2009)