

Hexahedral Mesh Generation using the Embedded Voronoi Graph

Alla Sheffer, Michal Etzion, Ari Rappoport, Michel Bercovier

Institute of Computer Science, The Hebrew University, Jerusalem 91904, Israel.
{sheffa|michals|arir|berco}@cs.huji.ac.il. <http://www.cs.huji.ac.il/~sheffa/~michals/~arir/~berco>).

Abstract

This work presents a new approach for automatic hexahedral meshing, based on the embedded Voronoi graph. The embedded Voronoi graph contains the full symbolic information of the Voronoi diagram and the medial axis of the object, and a geometric approximation to the real geometry. The embedded Voronoi graph is used for decomposing the object, with the guiding principle that resulting sub-volumes are sweepable. Sub-volumes are meshed independently, and the resulting meshes are easily combined and smoothed to yield the final mesh.

The approach presented here is general and automatic. It handles any volume, even if its medial axis is degenerate. The embedded Voronoi graph provides complete information regarding proximity and adjacency relationships between the entities of the volume. Hence, decomposition faces are determined unambiguously, without any further geometric computations. The sub-volumes computed by the algorithm are guaranteed to be well-defined and disjoint. The size of the decomposition is relatively small since every sub-volume contains a different Voronoi face. Mesh quality seems high since the decomposition avoids generation of sharp angles, and sweep and other basic methods are used to mesh the sub-volumes.

1 Introduction

Automatic generation of 3-D finite element meshes is essential for the automation of the analysis process. Research and development effort on new meshing algorithms has resulted in several automatic algorithms for 3-D meshing (see [7] for a recent review). Most of the research has focused on the generation of unstructured tetrahedral elements. This has resulted in several successful automatic mesh generation algorithms, which can be categorized according to the main approaches used: Delaunay triangulation [10, 28], octree-based methods [23, 2] and advancing front algorithms [8].

In many situations, hexahedral meshing is more attractive than tetrahedral meshing. The additional requirements of hexahedral meshing make the problem more demanding. Numerous approaches have been proposed and investigated: feature-based [13], medial surface subdivision [17, 18], plastering [3], grid-based [19, 24, 26], whisker weaving [27], etc. As with many difficult problems, each of the proposed solutions possesses both positive and negative attributes associated with the technique employed.

Several successful approaches were developed for different types of geometries of varying complexity. These include mappable and sub-mappable volumes [29], volumes meshable by midpoint subdivision [14], swept volumes and uni-axial combinations of swept volumes [4]. The relative success of these methods indicates that a promising approach towards the general problem is to decompose the volume into parts meshable by existing, well established techniques. This is the approach taken in [17, 18, 13, 12].

In [17, 18], the decomposition is guided by the medial axis of the object. The proposed algorithm gives a template based decomposition, building a template subdivision around each entity of the medial axis (its faces, edges and vertices). This results in an initial hexahedral mesh of the model, which can then be refined to the desired density.

The use of the medial surface for the decomposition provides a systematic and generic approach for all possible geometries. In addition, the decomposition process is directed by the medial surface, thus avoiding the computation

of decomposing surfaces and surface intersections. However, the described technique has several drawbacks. First, because the technique builds a template subdivision around each entity of the medial axis, it results in a fine subdivision of the model, even for simple cases. For example, for a brick (containing no medial surface degeneracies) the initial mesh contains 72 elements. Second, the algorithm used for computing the medial axis [20] is difficult to implement and not provenly correct. Third, for degenerate vertices and edges the suggested use of midpoint subdivision is likely to lead to poorly shaped elements.

In [13], a feature based decomposition of the object is generated. Features are recognized based on combinations of convex and concave edge loops. An attractive property of this approach is that it follows the intuition of manual subdivision. The approach is based on the model’s generic shape and is independent of the geometric parameters. Decomposition is in some sense minimal, because the process stops once the parts are convex or can be meshed by sweeping. However, the method possesses several drawbacks. Features with interacting geometry pose a difficult recognition problem. In many cases there are ambiguities when it is not clear which decomposition the algorithm should choose, and decomposition surfaces can cut each other. The result depends upon the order of performing the decompositions. The method requires computations of surface-surface intersections. Finally, convex shapes are not decomposed at all, even when both sweep or midpoint are not suitable.

Contribution. This work presents a new approach for automatic hexahedral meshing, based on the embedded Voronoi graph. The embedded Voronoi graph contains the full symbolic information of the Voronoi diagram and the medial axis of the object, and a geometric approximation to the real geometry. The embedded Voronoi graph is used for decomposing the object, with the guiding principle that resulting sub-volumes are sweepable. Sub-volumes are meshed independently, and the resulting meshes are easily combined and smoothed to yield the final mesh.

The presented approach possesses several advantages: (1) the algorithm for computing the embedded Voronoi graph is provenly correct, stable and easy to implement; (2) the approach is well defined and valid on shapes of any geometry, including shapes whose medial axis is degenerate and complex convex shapes; (3) the decomposition is order independent and prevents intersections between decomposition surfaces; (4) the number of sub-volumes generated is not large; (5) since the directions and entities involved in each decomposition are defined by the medial axis, there are no intersection computations; and (6) since a decomposition is used, as opposed to template, there is only a minimal need for medial axis geometry.

The main drawback of following the medial axis for the decomposition is that it sometimes can result in over-decomposition of the volume and often is not intuitive.

The paper is organized as follows. In Section 2 we briefly discuss the definition of the embedded Voronoi graph. Section 3 gives an overview of our approach. The main step, shape decomposition using the Voronoi graph, is detailed in Section 4. Generation of the mesh itself is described in Section 5. The application of the algorithm is demonstrated on several examples in Section 6. Section 7 discusses the advantages and drawbacks of the algorithm and suggests topics for future work.

2 The Embedded Voronoi Graph

In this section we briefly describe the embedded Voronoi graph, which is an approximation of the Voronoi diagram and the medial axis of the object. It is used here for decomposing the volume into simple parts. Full details on its definition and construction are given in [6].

Let Q be a volume. The *entities* of Q are the vertices, edges and faces of Q . An entity a is *incident* on an entity b iff one of the following is true: (1) a is an end vertex of the edge b , (2) a is a vertex of the face b , or (3) a is a bounding edge of the face b . Two entities a and b are *adjacent* if they are not incident one on the other, and there is another entity incident on both of them. For example a face and an edge that share a vertex. A Voronoi region R_a is the locus of points that are closer to entity a than to any other entity of Q . The boundaries of the Voronoi regions comprise the *Voronoi diagram* of Q , $VD(Q)$. $VD(Q)$ is comprised of *Voronoi faces*, *Voronoi edges*, and *Voronoi*

vertices. The *governors* of a Voronoi element are its nearest entities. In the non-degenerate diagram, a Voronoi face has two governors, a Voronoi edge has three governors, and a Voronoi vertex has four governors. In a degenerate diagram, edges and vertices can possess more governors. When the faces of the volume are linear, its medial axis can be easily obtained from the Voronoi diagram by deleting Voronoi faces and edges leading to concave vertices and edges of the volume. Constructing the medial axis from the Voronoi diagram is more problematic when the faces of the volume are not linear, since in this case the medial axis might contain elements governed by a single entity. Such elements do not belong to the Voronoi diagram.

The Voronoi diagram of a volume is attractive as a computational tool for geometric applications [1, 16, 25]. The Voronoi diagram is useful due to several reasons. First, its shape is closely related to the shape of the object, thus it can serve as a shape analysis tool. Second, it is of lower dimensionality than the object, thus is easier to deal with. Finally, it explicitly carries proximity information. However, the combination of a complex connectivity structure and the high algebraic degree of its geometric elements makes the construction of the Voronoi diagram of a volume a difficult problem.

Etzion et al [6] defined a set of Voronoi skeletons that approximate the Voronoi diagram of a polyhedron, and provided a simple algorithm to construct them. These skeletons provide information regarding both the symbolic structure of the Voronoi diagram and the geometric location of the diagram's elements. The symbolic structure of the Voronoi diagram contains the complete connectivity structure of the Voronoi diagram together with the governing entities of each Voronoi element. This paper uses the embedded Voronoi graph, which is defined below.

1. **Voronoi Graph.** The Voronoi graph contains all the symbolic information present in the Voronoi diagram, without containing any geometry. The Voronoi graph is a graph whose nodes correspond to elements of the Voronoi diagram. Each node corresponds either to a Voronoi vertex, or to a Voronoi edge, or to a Voronoi face. The label of a node is the set of governors of the corresponding element in the Voronoi diagram. An arc exists between two nodes of the graph if and only if there is an incidence relationship between the corresponding elements in the Voronoi diagram.
2. **Approximate Voronoi Graph.** An approximate Voronoi graph approximates the **Voronoi graph** of Q to a tolerance of ϵ in the sense that a connected subgraph of the Voronoi graph that lies in a region of space of size smaller than ϵ is replaced by a single graph node. For example, a Voronoi edge whose length is smaller than ϵ might be represented together with its two vertices by a single graph node. The label of this node contains the governors of the edge, together with the governors of its vertices.
3. **Embedded Voronoi Graph.** The embedded Voronoi graph is a **Voronoi graph** that also provides geometric approximation of specific elements of $VD(Q)$. The embedded Voronoi graph of Q with a parameter δ is a Voronoi graph (or approximate Voronoi graph) of Q s.t. some of the nodes of the Voronoi graph carry also a geometric approximation (of the appropriate type) to the corresponding element in $VD(Q)$, to an accuracy of δ . For example, a graph node that corresponds to a Voronoi edge, contains a polyline that approximates the Voronoi edge to an accuracy of δ .

An algorithm for constructing the embedded Voronoi graph of a polyhedron is given in [6]. Initially, a space subdivision whose cells are labeled according to their proximity to polyhedron entities is constructed. Subsequently, information regarding the symbolic structure of $VD(Q)$ is extracted from the subdivision. This information includes the Voronoi elements of $VD(Q)$, their governors, and their adjacency relationships. If it is known that $VD(Q)$ is not degenerate, then the Voronoi graph is constructed. Otherwise a tolerance parameter ϵ is determined, and the approximate Voronoi graph is constructed. In the final step, a geometric approximation of Voronoi elements of interest is extracted. Because the algorithm is based on space subdivision, a geometric approximation of a specific Voronoi element can be computed locally. The most complex geometric operation performed by the algorithm is intersecting two conic curves. The algorithm is easy to implement and robust. Its convergence and correctness are proven in [6].

We use the embedded Voronoi graph for decomposing the volume into easily meshable parts. Voronoi faces are used to identify pairs of volume faces that enclose a sweepable volume. Thus, the exact geometry of the Voronoi

faces is not important, only their existence. The Voronoi graph is sufficient for identifying such faces. The volume is actually decomposed by surfaces created by projecting Voronoi vertices and edges on their governors. In order to decide whether a Voronoi element should be projected, only symbolic information is needed, and therefore the **Voronoi graph** (or the **approximate Voronoi graph**) is used. This decision is based on the types of the governors of the Voronoi element, and on the adjacency relationships between the governors. If a specific Voronoi element has to be projected, then the geometric location of the element is approximated by the **embedded Voronoi graph**. The exact location of the Voronoi element is of little importance, since the projection surfaces are not part of the final mesh; their only role is to decompose the polyhedron into simple parts. Thus, the embedded Voronoi graph is suitable for our purposes, and there is no need to compute the exact Voronoi diagram, which is much more difficult.

3 Algorithm Overview

In this section a general overview of the meshing algorithm is presented. The algorithm consists of three main stages: (1) construction of the embedded Voronoi graph of the object, (2) decomposition of the object into simple parts, and (3) actual meshing.

Embedded Voronoi Graph. In the first stage of the algorithm, an embedded Voronoi graph of the input volume is computed (Figure 1(a)). The embedded Voronoi graph displayed in this figure is not symmetric since it gives an *approximation* to the location of the Voronoi elements. This approximation is sufficient for the meshing algorithm. However a finer approximation can be computed if needed. The algorithm for constructing the embedded Voronoi graph operates on linear polyhedra. Therefore, if the input object is not linear, its faces and edges must first be linearly approximated. During this approximation, for each approximation entity its originating entity is stored. After an embedded Voronoi graph is computed, the Voronoi entities whose governors are approximation entities originating from the same model entity are united.

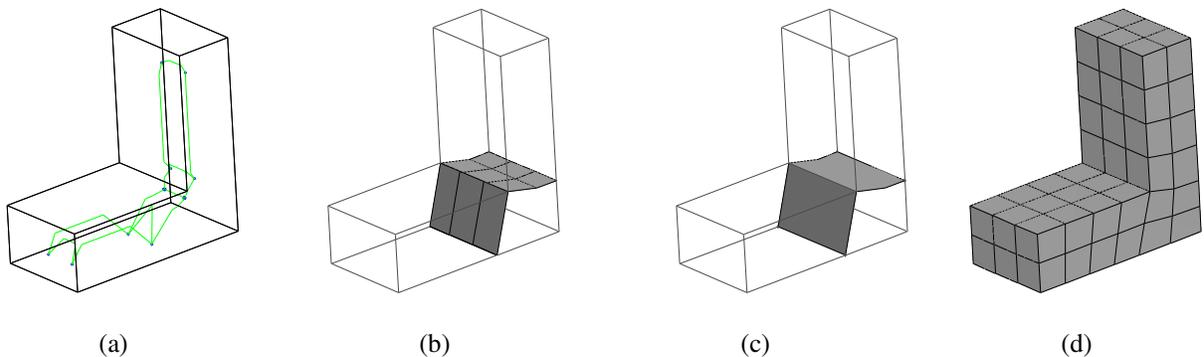


Figure 1: The stages of the meshing algorithm applied on an ‘L’ shaped volume: (a) the embedded Voronoi graph; (b) the decomposition faces, generated by the projection of Voronoi edges and vertices; (c) the decomposition of the volume into three sub-volumes, showing the decomposition faces after they were merged; (d) the resulting volume mesh (before smoothing).

Object decomposition. This is the major step of the algorithm. The main observation on which this stage is based is as follows. Consider a Voronoi face f governed by two faces a and b of V , where a and b are not adjacent. Consider the volume V_{ab} defined by a , b , and the projection of f to a and b . The volume V_{ab} possesses the following attractive properties:

- V_{ab} is wholly governed by a and b [6], and therefore does not intersect any other entity of the object nor any other element of its Voronoi diagram.

- V_{ab} does not intersect any other volume defined similarly by two faces of the object. Therefore V_{ab} can be meshed independently of other sub-volumes.
- V_{ab} is a sweep from a to b , hence can be meshed by a standard method to mesh sweep volumes.

Denote by f_s a Voronoi face governed by two unconnected faces (the ‘s’ subscript stands for ‘sweep’), and by V_s a sub-volume containing a Voronoi face of type f_s . Unfortunately, the union of all volumes of type V_s does not cover the whole object V . Denote by V_e a connected volume with no such face (it is a connected component of $V \setminus \cup V_s$).

The embedded Voronoi graph of a volume V thus provides a natural decomposition of the volume into simple parts. V is decomposed into sub-volumes of two types, V_s and V_e . The decomposition is achieved by projecting Voronoi edges that bound faces of type f_s on their respective governors, and creating decomposition faces between the edges and their projections. These faces are then merged to create the faces separating between the sub-volumes. Figure 1(b) shows the decomposition faces resulting from the projections, and Figure 1(c) shows the decomposition faces after they were merged.

In Section 4 the decomposition will be defined such that the following requirements are satisfied:

1. The faces decompose the volume into disjoint sub-volumes, i.e., it is a valid decomposition. This requires the decomposition to be defined at the Voronoi vertices, so that the decompositions induced by the edges that meet at the vertex will be connected correctly.
2. The parts after the decomposition can be meshed by basic algorithms (sweep, mapping or mid-point subdivision): this includes verifying that the decomposition faces are quadrilaterals, and that the angles between adjacent decomposition faces and between the decomposition faces and the original volume faces are not too sharp.
3. The number of resulting sub-volumes is not large.

The decomposition is performed using virtual topology operators, avoiding the complex computations required for actual geometric decomposition and allowing the easy removal of the decomposition surfaces later at the smoothing stage. Virtual topology operators are briefly reviewed in the next section.

Meshing. Finally the volume mesh is constructed by meshing the set of sub-volumes produced by the decomposition. After the sub-volumes are meshed, the decomposition boundaries are removed and partial meshes are united and reassigned to the original volume (Figure 1(d)). A smoothing procedure is then applied on the volume mesh as a whole. The global smoothing relaxes the constraints imposed on the mesh by the geometric positioning of the decomposition faces.

The result of the algorithm is a hexahedral mesh of the volume. The decomposition procedure and the mesh generation that follows are described in detail in the following sections.

4 Volume Decomposition

After the embedded Voronoi graph of the volume is constructed, the volume is decomposed into a set of meshable sub-volumes, by projecting Voronoi edges that bound faces of type f_s on their respective governors and creating decomposition faces between the edges and their projections. Below we discuss which edges and vertices of the Voronoi diagram are of interest for the decomposition and how they should be projected on their respective governors.

The section is organized as follows. In 4.1 a brief overview is given of the virtual topology operators used to perform the editing operations required by the decomposition. Sections 4.2 and 4.3 describe how the projection of the Voronoi edges and vertices is performed, and demonstrate it on all the combinations of Voronoi edge and vertex governors that can occur for non-degenerate Voronoi entities. Section 4.4 explains how the actual volume decomposition is done using virtual topology operators.

4.1 Virtual Topology Operators

To perform the volume decomposition, multiple editing operations are required (projection of Voronoi edges, construction of decomposition faces, splitting of the original model edges, faces and volume, etc.). The geometric computations required by those operations might add a significant overhead to the meshing procedure. This overhead is avoided by using the virtual topology tools reviewed here.

The virtual topology enhancement of the standard boundary representation (B-Rep) was introduced in [22]. It allows a large set of editing operations on the model topology, without changing the actual geometry.

In the standard B-Rep, there is a one-to-one correspondence between topological and geometrical objects, such that any adjustment of the topology requires changing the geometry as well in order to preserve the unique correspondence of topology and geometry.

The virtual topology enhances the B-Rep by uncoupling the topology from the geometry. In order to do this in addition to standard *real* entities, which contain as the description the exact, mathematical definition of the entity's geometry, another type of *virtual* entity is added. The *virtual* topological entities have no exact geometric definition of their own; they rely on other topological entities from which the entity geometric description is to be derived. The structure of this reliance varies depending on the editing operations applied, and the nature of the topology being edited. The virtual topology operators are particularly suitable for model editing performed for mesh generation [21].

4.2 Edge Projection

In order to create the volumes, Voronoi edges are projected on their governing entities in V . For each edge the decision of whether to project it on a specific governor and how to do this projection depends on the type of the governor entity and on the relations between the governors. As observed above, edges that are of interest for the decomposition are those that bound Voronoi faces of type f_s . These are exactly the edges having at least two unconnected face governors. Based on the decomposition goal of creating sub-volumes with one face f_s or none, the decomposition should separate the region governed by f_s from the rest of the volume.

Applying this concept on all types of non-degenerate Voronoi edges (edges governed by three entities in V), we obtain a finite set of cases described below. This classification is an extension of the one given in [17] for Voronoi edges governed only by faces. Types [0], [1] and [2] are defined as in [17], and include Voronoi edges whose governors are all faces. Types [0e] and [1e] are defined for Voronoi edges which have vertex or edge governors as well. The different types of Voronoi edges are defined as follows, and the projection performed for each case is shown in Figure 2.

[0] An edge with three face governors s.t. no pair of governors shares an edge.

[1] An edge with three face governors s.t. a single pair of governors shares an edge. Here a distinction is made between the case [1] where the angle along the shared edge is not sharp, and the case [1s] where the angle is sharp. The distinction between 'sharp' and 'not sharp' depends on the desired mesh quality.

[2] An edge with three face governors s.t. two pairs of governors share edges.

[1e] An edge with two face governors and an edge governor that is incident on one of the face governors.

[0e] An edge with two face governors and either a vertex governor or an edge governor that is not incident on any of the face governors. Note that this type contains the cases where a face governor is adjacent to the edge governor, but does not contain it.

Other Voronoi edges are of no interest for the decomposition, since they are not incident on Voronoi faces governed by two faces of the volume. An example of such an edge is the edge directly below the concave edge of the 'L' shape in Figure 1(a). This edge is governed by the concave edge and the bottom and right faces of the 'L'.

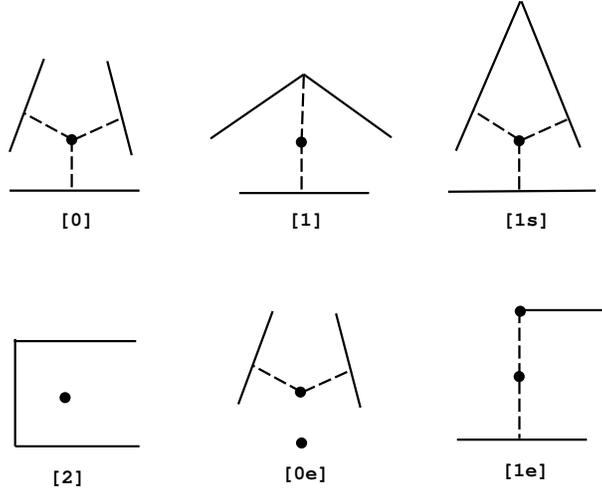


Figure 2: *Projection of the different types of Voronoi edges. The entities of the volume are shown in bold lines, and the decomposition faces added are shown in dashed lines.*

In order to separate the region governed by a pair of unconnected face governors from the rest of the model, the decomposition faces should be added as shown in Figure 2. The intuition behind this projection is the following. The simplest approach would be to project the Voronoi edge on all its governing faces (as done for types [0] and [0e]), but since we want to reduce the number of decompositions and create simpler volumes, the number of decomposition faces can often be reduced. Type [2] is the most common type of a volume boundary region. For example, in Figure 3(a) all the displayed Voronoi edges are of this type. Projection of Voronoi edges of this types will result in a redundant decomposition. Therefore no projection of these edges is done, and as a result, the sub-volumes adjacent to these edges are extended towards the boundary of the volume. In this case additional decomposition faces are sometimes added at the vertices of these edges, to guarantee valid sub-volumes (Section 4.3).

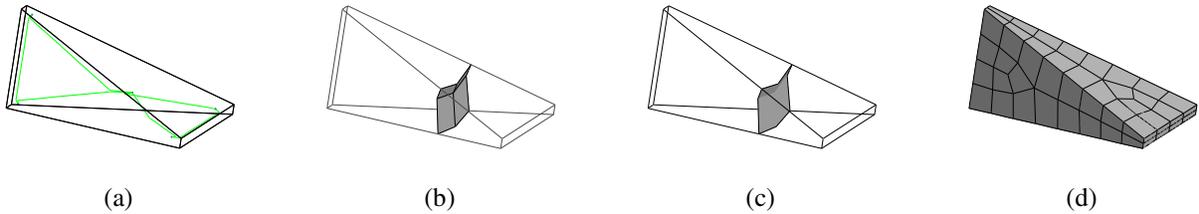


Figure 3: The stages of the meshing algorithm applied on a tapered brick (used in [17]): (a) the embedded Voronoi graph of the volume; (b) the decomposition faces as generated by the edges and vertices projection. In this example the displayed edges are of type [2] and hence the decomposition is performed only at the central vertex, which is of type [2,2,2,2]; (c) the merged decomposition faces and the two resulting sub-volumes; (d) the final mesh before smoothing.

For a Voronoi edge of type [1] to separate the two Voronoi faces of interest (e.g. the Voronoi face governed by the bottom and right faces and the Voronoi face governed by the bottom and left faces), it is sufficient to project the edge on the bottom face and on the common edge of the two upper faces. However, in case of a Voronoi edge of type [1s], this will cause the generation of a low quality mesh and hence the decomposition is performed as shown in Figure 2[1s].

Projection on governors that are not faces does not separate any regions of interest and hence is unnecessary. However, in the case of a Voronoi edge of type [1e], the projection is done on the governing edge and not on the

governor face containing that edge, to simplify the shape of the region governed by the edge and the second face. For example, in Figure 1(a) the type [1e] edges are the one below and the one on the right of the concave edge of the 'L' shape. The projection can be seen in Figure 1(b). The projections on the bounding edges of the face (cases [1], [1e]) are viewed as the projections to the relevant governor face.

The same strategy is extended to degenerate edges (i.e. edges with more than three governor entities) as well. Clearly in such cases no finite set of governor combinations exists and the edge projection is determined for each governor face based on its relationship with other governors and their types. The application of the strategy to a degenerate Voronoi edge is shown in Figure 4. The Voronoi edge has four governors: the two vertical (lower) faces and the two edges on top of them. Thus, it is a kind of a [1e] edge, and it is projected on the two governing edges.

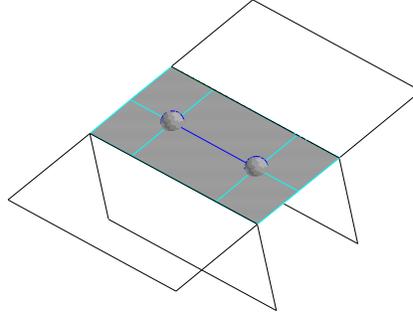


Figure 4: *Decomposition at degenerate Voronoi elements. The dark line shows the degenerate Voronoi edge. This edge appears at the top of the rod of Figure 7. It has four governors: the two vertical (lower) faces, and the two edges on top of them.*

There are two additional types of Voronoi edges that should be considered, although they are not incident on Voronoi faces of type f_s .

[3] An edge with three face governors s.t. three pairs of governors share edges.

[C] An edge whose governors share a common vertex. Such edges are not displayed in the figures showing the embedded Voronoi graph, in order to make the figures less cluttered.

These Voronoi edges are added as special cases to simplify the treatment of Voronoi vertices shared by them and some of the edges addressed above, which are quite common. A Voronoi edge of type [3] indicates a sub-volume that can be meshed by a sweep along the edge. However, in order to simplify the decomposition scheme at the vertices, such edges are treated as edges of type [0], i.e. they are projected on their face governors. A Voronoi edge of type [C] is not projected, but it affects the projection performed at its end vertices as described below.

Clearly, the projection of an edge involves the projection of its end vertices onto the same model entity. In order to decompose the volume using the decomposition along the Voronoi edges, the decomposition has to be extended at the edge end vertices, either by continuing it towards the volume envelope or by combining it with the decompositions defined by other Voronoi edges that share this vertex. The way this is done is explained in the next section.

4.3 Handling of Vertices

In order to decompose the volume along Voronoi edges of interest as described above, the decomposition needs to be defined at the end vertices of the edges.

The vertex treatment includes projecting the vertex on its governor entities, based on the desired projection of the vertex edges on the appropriate governor. Sometimes additional decomposition faces need to be constructed at a vertex to close gaps in the decomposition defined by the edges.

To prevent creation of triangular faces and volumes and avoid unnecessary decompositions, a vertex is projected only on a single location on each governor face. For many edge combinations the projections of the edges match at the vertex with no extra intervention. In the other cases, this involves changing the positions of the end vertex projections as defined by the edges, and sometimes changing the edge projection as well.

When one of the Voronoi edges emanating from the vertex was not projected onto the volume, the decomposition defined by the other edges often needs to be extended towards the volume envelope in order to close gaps.

Below we explain how these operations are applied to the different types of Voronoi vertices. The Voronoi vertices considered are vertices that are incident on Voronoi edges of the types enumerated in Section 4.2. Other vertices are of no interest for the decomposition. The Voronoi vertices are classified according to the Voronoi edges they are incident on. In the non-degenerate case, every Voronoi vertex is incident on four Voronoi edges.

Consider first Voronoi vertices all of whose governors are faces. The edges emanating from these vertices are of one of the following types: [0], [1] ([1s]), [2], [3] or [C]. In this case only a limited number of edge combinations at a vertex exists [17]; the vertex treatment for each of those is shown in Figure 5. The figure shows the decomposition faces at each vertex, including the faces created along each Voronoi edge emanating at the vertex, and the additional faces, added at the vertex, when necessary. Note that the figure is only for visualization of the cases; all the geometry shown is schematic and does not have any relationship to the final mesh.

If all the Voronoi edges emanating from the vertex are of types [0], [3] or [1s], then the projection of the vertex on each governor, as defined by the edge projections, is the same, and no extra adjustments are required. For vertex with edges of type [1], the projection to governor faces needs to be moved to the common edges shared by the governors, to avoid the duplicate projection.

If some of the edges are of type [0] (or [3]), and some are of type [1], then a decision should be taken whether to project the Voronoi vertex according to the [0] (or [3]) edges, or according to the [1] edge. That is, whether to project the vertex on the face (like in [1s] case), or on the edge of the face (like in [1] case). In each specific case (Figure 5) a decision was done as to which option to take s.t. (1) no triangular sub-volumes are created, and (2) no redundant decompositions are done. If the projection at the vertex has to be handled as a [1s] case, then the edge projection will also be performed as [1s].

If the set of edges emanating from a Voronoi vertex includes edges of type [2] and [C], then additional decomposition faces should sometimes be created.

While a Voronoi edge of type [2] is incident on one face of type f_s , an end vertex of such an edge may be incident on other f_s faces, hence additional decomposition at the vertex might be required to separate the regions governed by those faces. If such faces exist (the vertex includes [0],[1], or [3] edges, or four [2] edges), then the decomposition is done as follows. Suppose that an edge of type [2] is governed by faces a, b, c s.t. b is adjacent to a and c . Then, in order to separate the region of the Voronoi face governed by a and c from the rest of the volume, additional decomposition faces are generated between a and b , and between c and b . A decomposition face between a and b is the rectangle $v, \pi_a(v), \pi_{ab}(v), \pi_b(v)$ where v is the Voronoi vertex, and ab is the edge between faces a and b . The Voronoi diagram of the tapered brick in Figure 3 includes a vertex (the central one) of type [2222]. Therefore, to separate the regions governed by the two f_s faces containing it, closing rectangles are generated between the four governors of the vertex.

Voronoi edges of type [C] may also require special treatment at their end vertices. Let a, b, c be the faces governing an edge of type [C]. Since the [C] edge is not projected to its governors, if an edge governed by a and b (which is attached to the same vertex) is projected to a there will be a gap between the decomposition face added and the face c . Such gaps can be handled either by moving the projection points to the shared edge ac , or by creating additional decomposition faces between the vertex and pairs of governors (a and b , a and c , and b and c). Moving is preferable, since it generates fewer decomposition faces. The decision of which approach to take

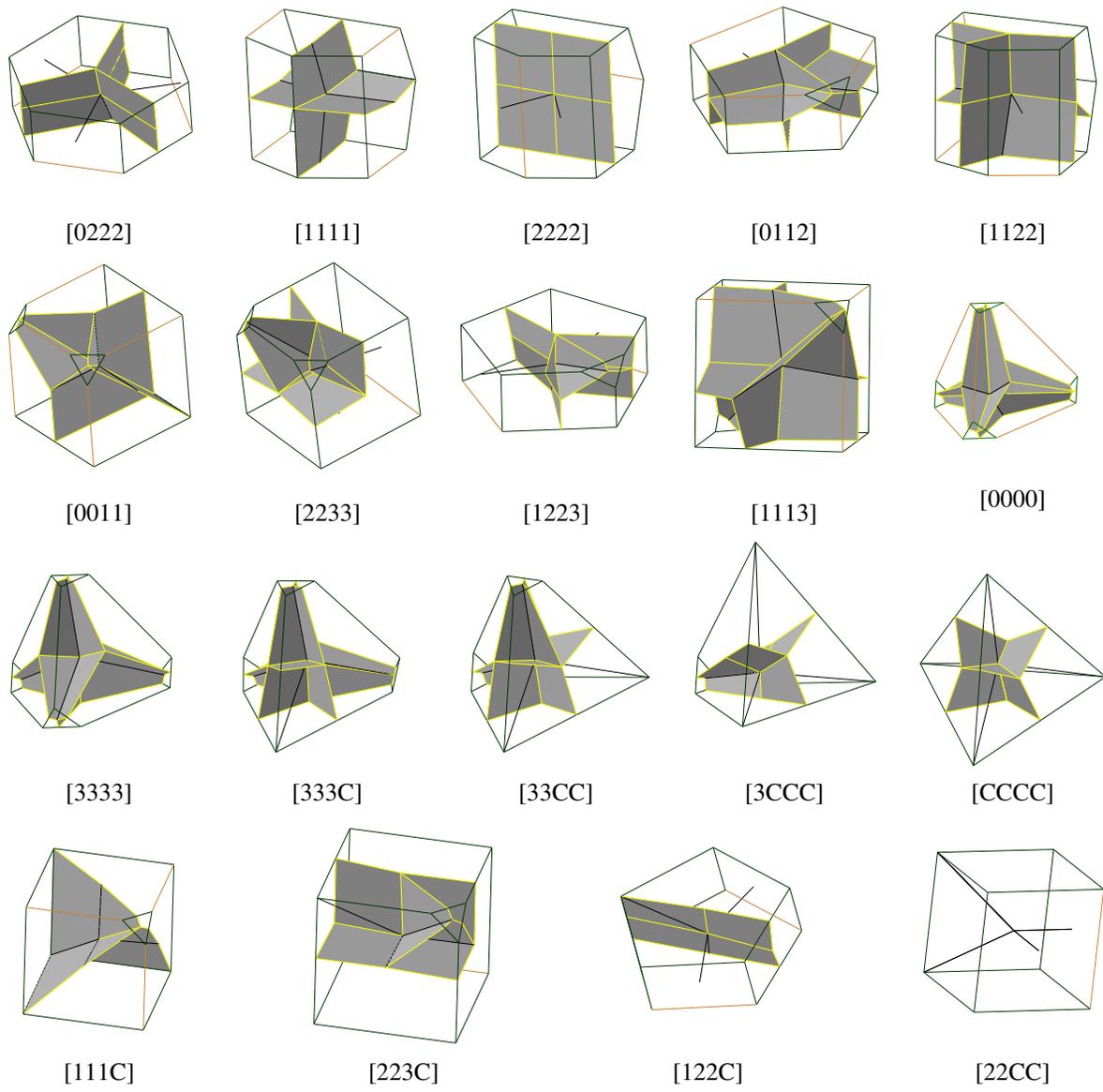


Figure 5: Vertex treatment for non-degenerate vertices with face governors. The decomposition faces are shaded. The Voronoi edges are shown in black. The Voronoi vertex is at the meeting point of the edges. The governor faces boundaries are dark-green. Note: this is only a schematic visualization of the vertices, with no correspondence to the actual sizes of the entities involved.

depends on the type of other edges at the vertex and is shown for each combination in Figure 5.

The treatment of Voronoi vertices with edge or vertex governors is based on the same considerations. We will not detail all types of Voronoi vertices in this case, but rather demonstrate the vertex treatment on a specific example. Consider Figure 1(a) and (b). The internal part of the Voronoi diagram of the volume contains ten Voronoi vertices. Four of them (at the four volume corners) are of type [22CC] and hence no projection or decomposition is performed at them. Two vertices (below the concave edge) are not incident on faces of type f_s and hence no decomposition is performed at them. The remaining four are symmetric and are governed each by three faces and an edge. Consider the vertex v governed by the back-right face (a), the front face (b), the concave edge (d) and the vertical face containing it (c). The Voronoi edge governed by abc is of type [2], the Voronoi edge governed by acd is of type [1e], the Voronoi edge governed by bcd is of type [C], and the edge governed by abd is not incident on a face of type f_s , and hence is not projected anywhere. Based on the edge acd the vertex is projected on a and d . As described for a vertex containing a [2] type edge (abc), an additional projection is performed on b and decomposition faces between a and b and b and c are created (the projection on d is viewed as the projection on c as described above). For the edge bcd to prevent the gaps in projections on b and c/d a rectangle between b and c should be created, but it is already in place.

Voronoi vertices that are degenerate, i.e. vertices with more than four governors, are treated using a similar strategy. An example is shown in Figure 4. The Voronoi edge has four governors: the two vertical faces and the two edges between the vertical and the horizontal faces. Each of its vertices is governed also by the front/back faces of the rod. Therefore each Voronoi vertex has four emanating edges: a degenerate edge projected on the two governing edges, and edges of type [2] and [C]. Therefore rectangles are added between the governing edges and the front/back face.

4.4 The Actual Decomposition

After the set of decomposition faces is generated at the edges and vertices of the embedded Voronoi graph, the original volume needs to be split into a set of volumes using those faces. This part of the decomposition procedure is demonstrated in the examples in Figures 3 and 1 (b) and (c). This is achieved using the following sequence of operations:

- An automatic merging procedure is applied on the set of the decomposition faces, uniting sets of faces that can be united topologically and have no sharp angles between them. An extra constraint that can be added to simplify the meshing is that faces are not merged if the resulting face is not quadrilateral. For example, in Figure 3 (b) and (c) for the [2222] type vertex this results in merging the four faces created at the vertex into a single face.

After the faces are merged, the face edges are merged as well using a similar procedure.

- A connect operation is now applied on the edges of the new faces and the original model, to connect together coincident edges. For example, in Figure 1(b) and (c) after the generation of the decomposition faces and the merges, there are two edges (from two decomposition faces) coincident to the concave edge of the model. Often this procedure includes also t-vertex connection as edges might coincide only partially.
- After all edges are connected, the decomposition faces need to be incorporated into the volume topology, by splitting the volume edges by decomposition faces vertices lying on them, and then splitting the volume faces by the appropriate edges.
- After all the topological structures are well connected, the volume is split into sub-volumes by the set of faces.

Note that in Figure 1(a) the approximated location of the Voronoi edges and vertices as given by the embedded Voronoi graph is quite far from their actual location. Therefore three of the decomposition faces are very small

(almost collapse into an edge). However the quality of the decomposition, and as a result, the quality of the mesh, are not impaired.

Both the construction of the decomposition faces at the Voronoi edges and vertices and all the operations above are implemented using the appropriate virtual topology operators. This way all the operations are performed only on the model topology and complex geometric computations are avoided. Another advantage is that since the original model geometry remains unchanged, after the set of sub-volumes is meshed the decomposition faces can be removed, and a smoothing procedure can be applied on the whole volume, without the constraints imposed by the geometry of the decomposition faces. The whole procedure can be fully automated.

5 Generation of the Mesh

The mesh of the volume is built by meshing the sub-volumes resulting from the decomposition, and then reassigning the mesh to the original volume and smoothing it. The two stages are described below.

5.1 Meshing the Parts

After the decomposition procedure, two types of sub-volumes are created: volumes of type V_s , i.e., containing a single Voronoi face governed by two unconnected faces (f_i and f_j) of the original volume, and volumes of type V_e , i.e., containing no such surface.

Volumes of type V_s can be meshed by a sweep mesh between f_i and f_j , since all the side faces between the two are quadrilateral:

- The faces generated by the decomposition are always quadrilateral.
- Based on the definition of the decomposition procedure, the only faces (partial faces) of the original volume that can take part in such a volume (besides f_i and f_j) are faces sharing a [2] type edge with f_i and f_j . The treatment of Voronoi vertices incident on a [2] type edge as described above guarantees that those faces are also quadrilateral.

The V_e type volumes can be classified into two types:

1. Volumes resulting from extending the decomposition surfaces at the Voronoi vertices towards the volume faces. Such volumes are always hexahedral, as demonstrated above (Figure 5).
2. Volumes resulting from decomposition along one or more Voronoi edges. From our experience so far such volumes can be meshed by sweep in the direction of the Voronoi edges. However, a formal study of the nature of such volumes must still be done.

In the tapered brick example (Figure 3(d)) the two sub-volumes are meshed by sweeping between the governors of the Voronoi faces present in each. The sub-volume on the left is meshed by sweeping between the front and back faces, and the volume on the right by sweeping between top and bottom faces.

In the L-shape example (Figure 1(d)) there are two V_s type sub-volumes in the two ends of the shape, and a V_e volume at the L-shape corner. The two end sub-volumes are meshed by sweeping between the governor faces, and the corner volume is meshed by sweeping in the direction of the decomposition edges along it.

When meshing the sub-volumes, mesh conformity has to be maintained between the adjacent sub-volumes. This can be achieved by using an interval assignment algorithm on the set of the sub-volume faces [15].

5.2 Smoothing

After the sub-volumes are meshed, the basic problem of generating a hexahedral mesh of the volume is solved. However, the restrictions imposed on the mesh by the geometry of the decomposition faces may affect the mesh

quality. The fact that the computation of the Voronoi diagram geometry is only approximate, also comes into account, resulting sometimes in non-intuitive positioning of the decomposition, and often in non-symmetric decomposition of symmetric volumes (Figure 7(d)). Since the constraints imposed on the mesh by the decomposition are artificial from the user point of view, it would be beneficial to the mesh quality to remove them.

Again, this can be done using virtual operators. The original topology is restored by merging all the entities split by the decomposition: first the sub-volumes are merged into a single volume, then volume faces split by the edges of the decomposition faces are merged back, and finally the volume edges split earlier are merged back. Throughout this procedure the mesh is preserved and is reassigned to the new, merged, entities.

After the original topology is restored, a smoothing procedure can be applied on the mesh of the volume as a whole, thus improving the mesh quality without the decomposition constraints. There is a variety of techniques for performing the smoothing including several ‘classic’ ones as reviewed in [11], and more recent ones such as [5].

6 Results

Most of the algorithm has been implemented. The embedded Voronoi graph part is a stand-alone application running under Unix. The volume decomposition part is implemented using the commercial mesh generator GAMBIT [9], used for sweeping and mapped meshing of the sub-parts. The implementation of virtual topology in GAMBIT was used for the multiple editing and decomposition operations required. The run time of the algorithm is comparable to other mesh methods.

The full implementation of the algorithm is still underway. Currently it lacks the preprocessing and postprocessing stages. The preprocessing stage includes linearization of non-polyhedral volumes. The postprocessing stage includes smoothing the volume mesh.

The algorithm is demonstrated on two complex real life examples. The example models include many of the possible decomposition types described above. The Voronoi diagrams of the models include multiple degenerate edges and vertices, which the algorithm has no difficulty to handle, since the decomposition approach is generic, and is not limited to non-degenerate entities.

The first example is shown in Figure 6. The volume is decomposed into eighteen swept volumes: The six triangular protrusions, the six main faces, and six volumes connecting the main faces. Each of the six triangular protrusions, and the six main faces, contain a single Voronoi face of type f_s , and therefore a sweep is done between the faces of the volumes governing the Voronoi face. The six volumes connecting the main faces are swept along their Voronoi edges. The final mesh is shown in Figure 6(d).

In Figure 7, the volume can not be meshed by sweep or uni-axial combination of sweeps, but requires the use of either decomposition or generic meshing algorithms. The Voronoi diagram of the volume contains many degenerate edges and vertices. The degenerate edge at the top of the part is shown in Figure 4. The volume is decomposed into swept volumes (the six parts of the bottom wheel, and the side parts on the top) and mappable volumes (Figure 7(c)). The final mesh is shown in Figure 7(d). As can be observed, even without the final smoothing the mesh quality is very high.

This example shows a drawback of the use of the medial axis for mesh generation: its sensitivity to scaling. The decomposition faces inside the wheel are redundant, since the wheel as a whole is a sweepable volume. These decomposition faces would have been avoided if the wheel had been narrower. In that case, there would have been a single Voronoi face in the wheel, a face governed by the two sides of the wheel (as occurring in the two sides at the top of the volume). In this case the whole wheel would have been a single sub-volume in the decomposition.

7 Discussion

In this paper we presented a hexahedral mesh generation algorithm. The algorithm uses the embedded Voronoi graph of the volume to decompose the volume into simple parts that can be meshed using basic meshing methods.

The approach presented here is general and automatic. It handles any volume, even if its medial axis is degenerate. The embedded Voronoi graph provides complete information regarding proximity and adjacency relationships between the entities of the volume. Hence, decomposition faces are determined unambiguously, without any further geometric computations. The sub-volumes computed by the algorithm are guaranteed to be well-defined and disjoint. The size of the decomposition is relatively small since every sub-volume contains a different Voronoi face. Mesh quality seems high since the decomposition avoids generation of sharp angles, and sweep and other basic methods are used to mesh the sub-volumes. The decomposition directions depend on the object and not on an arbitrary external coordinate system, as in octree-based methods.

Hexahedral 3-D mesh generation using the medial axis has been presented in [17, 18]. The present work builds upon that work, and uses a classification of Voronoi elements similar to that defined in [17]. The advantages of the present algorithm over the one of [17, 18] are the following.

1. The number of pieces created by the present algorithm is much smaller, because the decomposition here is performed only in order to separate regions governed by faces of type f_s from regions governed by other Voronoi diagram faces. For example, a rectangular volume will not be decomposed at all (since it either contains a single f_s face, or in the degenerate case of a cube no faces at all), as opposed to a subdivision into 72 hex sub-volumes given by [17].
2. The algorithm uses the embedded Voronoi graph instead of the medial axis (or Voronoi diagram) of the volume. Computing the exact medial axis of a linear polyhedron requires solving systems of tri-variate quadratic equations, resulting in algorithms that are not robust, difficult to implement, and difficult to prove correct. The most complex geometric operation performed by the algorithm to construct the embedded Voronoi graph is intersection of two conic curves. The algorithm is simple to implement, and proven correct. The information given by the embedded Voronoi graph is sufficient for the present algorithm. The embedded Voronoi graph gives correct symbolic information; there is no need for exact geometric information, since the projection faces emanating from Voronoi elements are not part of the final mesh. Their only role is to decompose the volume into simple parts that are later meshed using basic meshing methods. A smoothing procedure is then applied to the volume mesh as a whole, making the exact location of the decomposition faces even less important. This stands in contrast to [17, 18], where exact information of the Voronoi elements locations and radii is used to define the mesh.
3. The algorithm handles volumes with a degenerate medial axis. The same strategy is applied to degenerate and non-degenerate volumes. The determination of decomposition faces is identical, and thus also the decomposition into sweepable sub-volumes. In [17, 18], in the non-degenerate case each primitive obtained is one of the 13 types defined, and an appropriate mesh procedure can be applied according to the type of the primitive. In the degenerate case the method creates primitives that are not part of the existing set. Therefore a midpoint subdivision is applied to these primitives. However, in the degenerate case there might be many elements meeting at the primitive center, resulting in poorly shaped elements.

The approach presented in this paper has some drawbacks, whose removal should be investigated in the future. The most conspicuous drawback is that the medial axis is sensitive to scaling, a fact which can result in over-decomposition of the volume. For example, in Figure 7 there are two locations where the decomposition faces obtained are redundant: the wheel (which is a sweepable volume as a whole), and the two corners at the top. In both cases a scaling of the volume would eliminate the redundant decomposition faces (perhaps creating new redundant faces in other places.) Another direction which should be investigated in order to minimize over-decomposition is a post-processing stage that merges adjacent sub-volumes into larger sweepable volumes.

The algorithm presented in this paper decomposes the volume into sub-volumes that are later meshed by basic meshing methods. We have shown that most of the sub-volumes are sweepable or hexahedral (Section 5.1). However, there is one set of sub-volumes that has not been proven to be either sweepable or hexahedral. These are the sub-volumes of type V_e that result from decomposition along one or more Voronoi edges (Section 5.1, item 2).

From our experience so far, such volumes can be meshed by sweep in the direction of the Voronoi edges. However, further study should take place to either prove this conjecture or to define further decomposition of such volumes to ensure that the volumes obtained can be meshed by available basic algorithms.

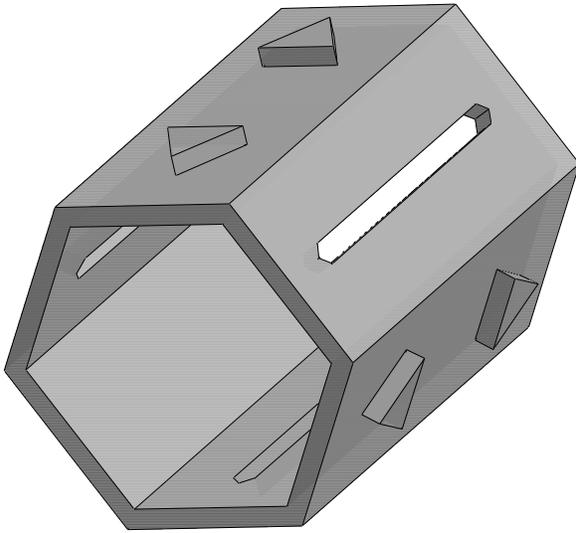
An important extension to the algorithm presented in this paper is handling of non-polyhedral volumes. If the medial axis of the non-polyhedral volume does not include points that have multiple projection points on a single volume entity, then it probably suffices to implement the procedure described in Section 3 (part 1). This includes approximating the entities of the volume by linear entities, computing the embedded Voronoi graph of the linear approximation, and then uniting Voronoi elements that are governed by the same original entities. In order to tackle the complete domain of non-polyhedral volumes, volumes with Voronoi elements with multiple projection points on an entity should be handled as well.

Acknowledgments. Part of this research was supported by the Israeli Ministry of Science, the program for development of scientific and technological infrastructures.

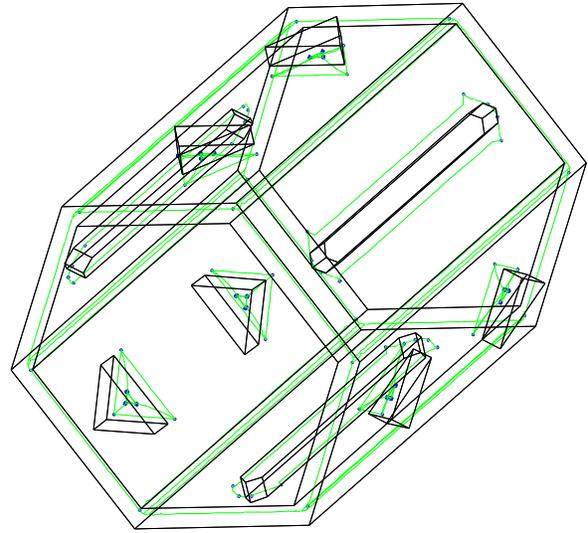
References

- [1] Armstrong C.G., Modeling requirements for finite-element analysis, *Computer-Aided Design*, 26(7):573–578, 1994.
- [2] Bern, M. ., Eppstein, D., Gilbert, J.R., Provably Good Mesh Generation, *Journal of Computer System Sciences*, vol 48, pp. 384-409, 1994.
- [3] Blacker T.D., Meyers R.J., Seams and wedges in plastering: a 3D hexahedral mesh generation algorithm, *Engineering with computers*, 9, 1993, pp.83-93.
- [4] Blacker, T. D., 1996, The Cooper Tool, 5th International Meshing Roundtable, SAND 95-2130, Sandia National Laboratories.
- [5] Canann S. A., Stephenson M. B., Blacker T., Optismoothing: An optimization-driven approach to mesh smoothing, *Finite Elements in Analysis and Design*, Elsevier Science Publishers, Num 13, pp.185-190, 1993
- [6] Etzion, M., Rappoport, A., Computing Voronoi Skeletons of a 3-D Polyhedron by Space Subdivision, Technical Report TR-8-97, Institute of Computer Science, The Hebrew University of Jerusalem, 1997.
- [7] Field, D. A., The legacy of automatic mesh generation from solid modeling, *Computer Aided Geometric Design*, 12(7), November 1995.
- [8] Frey, P.J., Borouchaki, H., George P.L., Delaunay Tetrahedralization using an Advancing-Front Approach, 5th International Meshing Roundtable, SAND 95-2130, Sandia National Laboratories. 1996.
- [9] GAMBIT 1.0 CFD preprocessor, User's Guide, Fluent Inc., 1998.
- [10] George, P.L., Improvements on Delaunay-based three-dimensional automatic mesh generator, *Finite Elements in Analysis and Design*, 25, 1997.
- [11] Ho-Le K., Finite Element Mesh Generation Methods: A Review and Classification, *Computer Aided Design*, Butterworth & Co. Ltd, Vol 1, Num 20, pp.27-38, Jan/Feb 1988
- [12] Holmes, D.I., 1995, Generalized Method of Decomposing Solid Geometry into Hexahedron Finite Elements, proceedings, 4th International Meshing Roundtable, Sandia National Laboratories.
- [13] Liu, S. S., Gadh, R., Automatic Hexahedral Mesh Generation by Recursive Convex and Swept Volume Decomposition, 6th International Meshing Roundtable, Sandia National Laboratories, pp. 217-231, 1997.

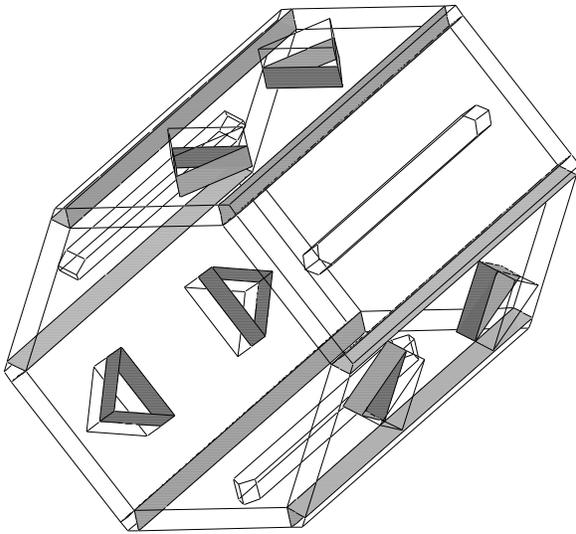
- [14] Li, T.S., McKeag, R.M., Armstrong, C.G., Hexahedral Meshing using Midpoint Subdivision and Integer Programming, *Computer Meth. App. Mech. Eng.*, Vol 124, pp. 171-193, 1995.
- [15] Mitchel S., High Fidelity Interval Assignment, 6th International Meshing Roundtable, Sandia National Laboratories, pp. 33-44, 1997.
- [16] Patrikalakis, N.M., Gursoy, H.N., Shape interrogation by medial axis transform, *Advances in Design Automation*, Vol. 1: Computer Aided and Computational Design, B. Ravani (ed.), ASME, 1990.
- [17] Price, M. A., Sabin, M. A., Armstrong, C. G., 1995, Hexahedral Mesh Generation by Medial Axis Subdivision: Part I. Solids with Convex Edges, *Int. J. of Numerical Methods in Eng.*, Vol.38.
- [18] Price, M. A. and Armstrong, C. G., 1997, Hexahedral Mesh Generation by Medial Axis Subdivision: Part II. Solids with Flat and Concave Edges, *Int. J. of Numerical Methods in Eng.*, Vol 40, pp. 111-136.
- [19] Schneiders, R., 1996, "A Grid Based Algorithm for the Generation of Hexahedral Element Meshes", *Engineering with Computers*, Vol. 12, No. 3-4, pp. 168-177.
- [20] Sheehy D.J., Armstrong C.G., Robinson D.J., 1995, Computing the medial surface of a solid from a domain Delaunay triangulation, *ACM Symposium on Solid Modeling Foundations and Applications*, Utah, pp. 201-212.
- [21] A. Sheffer, T. Blacker and M. Bercovier, Virtual Topology Operators for Meshing, *Proc. 6th Intl. Meshing Roundtable*, pp.49-66, October 1997. To appear in *International Journal of Computational Geometry and Applications*.
- [22] Sheffer, A., Blacker, T. D., Clements, J., Bercovier, M., *Virtual Topology Construction and Applications, Geometric Modeling Theory and practice*. pp. 247-259, Springer-Verlag, 1997.
- [23] Shephard, M.S. and Georges, M.K., Automatic Three Dimensional Mesh Generation by the Finite Octree Technique, *International Journal of Numerical Methods in Engineering*, vol 32, pp 7-9-749, 1991.
- [24] Smith, R. J., Leschziner, M., A., A Novel Cartesian Grid Method for Complex Aerodynamic CFD Applications, 5th International Conference on Numerical Grid Generation in Computational Field Simulations, Ed B. K. Soni, J. F. Thompson, J. Hauser, P. Eiseman, Mississippi State University, 1996.
- [25] Storti D.W., Turkiyyah G.M., Ganter M.A., Lim C.T., Stal D.M., Skeleton-based modeling operations on solids, *ACM Symposium on Solid Modeling Foundations and Applications*, Atlanta, May 1997, pp. 141-154.
- [26] Taghavi, R., Automatic, Parallel and Fault Tolerant Mesh Generation from CAD, *Engineering with Computers*, Springer-Verlag, Vol 12, pp. 178-185, December 1996.
- [27] Tautges, T. J., Blacker, T. D., Mitchell, S. A., 1996, The Whisker Weaving Algorithm: A Connectivity-Based Method for Constructing All-Hexahedral Finite Element Meshes, *Int. J. of Numerical Methods in Eng.*, Vol.39 No 19.
- [28] Weatherill, N. P. and Hassan, Efficient Three-dimensional Delaunay triangulation with Automatic Point Creation and Imposed Boundary Constraints, *Int. J. Numerical Methods in Engineering*, vol 37, pp. 2005-2039, 1994.
- [29] White, D.R., Lai, M., Benzley, S.E., Sjaardema, G.D., 1995, Automates Hexahedral Mesh Generation by Virtual Decomposition, 4th International Meshing Roundtable, SAND95-2130, Sandia National Laboratories.



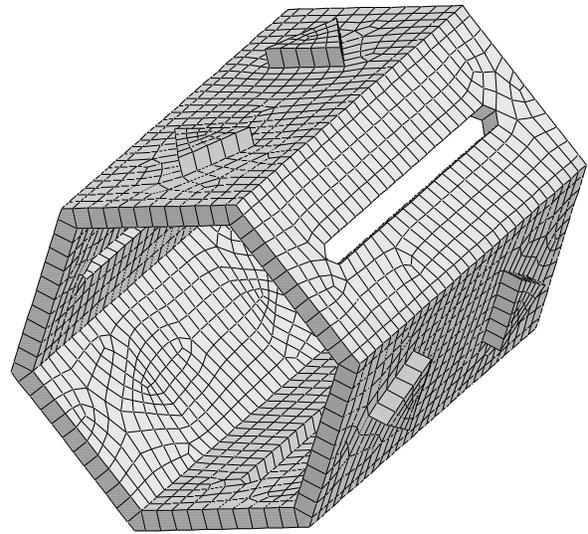
(a)



(b)

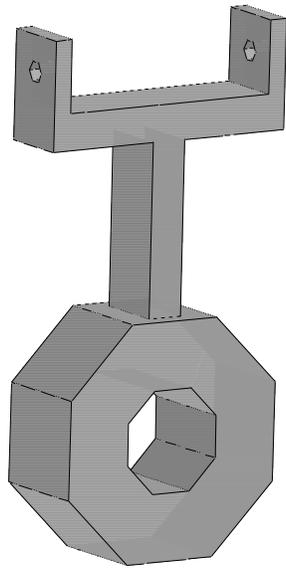


(c)

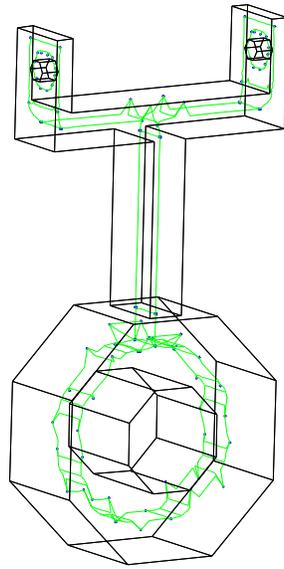


(d)

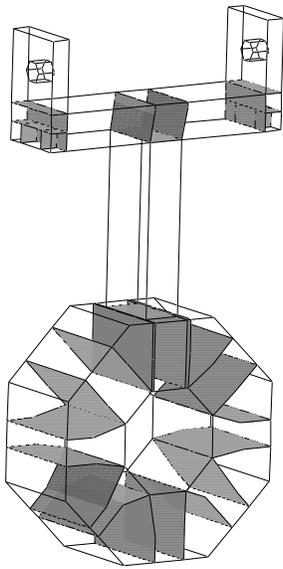
Figure 6: A meshing example: (a) the initial volume; (b) the embedded Voronoi graph of the model; (c) the decomposition faces generated based on the embedded Voronoi graph; (d) the final mesh before smoothing.



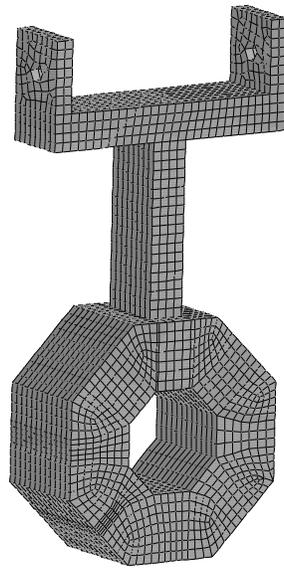
(a)



(b)



(c)



(d)

Figure 7: An additional example: (a) the initial volume; (b) the embedded Voronoi graph of the volume; (c) the decomposition faces generated based on the embedded Voronoi graph; (d) the final mesh before smoothing.