TESSELLATIONS OF CUBOIDS WITH STEINER POINTS

N. Hitschfeld[1,2], G. Navarro[2] and R. Farías[2],

[1] *Integrated Systems Laboratory, ETH-Zürich,*
nancy@iis.ee.ethz.ch
[2] *Department of Computer Sciences, University of Chile,*
gnavarro@dcc.uchile.cl,rfarias@dcc.uchile.cl

ABSTRACT

This paper presents a study of different 1-irregular cuboids (cuboids with at most one Steiner point on each edge) that can appear when meshes are generated using extensions of the modified octree approach [5], and then gives a recommendation how to handle them. The study is divided into two parts depending on the type of refinement used: First, for the bisection based approach (Steiner points are midpoints of the cuboid edges), the 1-irregular cuboids are classified into equivalence classes (each element of the class is partitioned in the same way) and the exact value of the number of equivalence classes is computed. As this value is not too big, all 1-irregular cuboids can be handled using a hash table, and then a tessellation can be always found in constant time. Second, for the intersection based approach (Steiner points can be located at any position along a cuboid edge), the total number of 1-irregular cuboids, and upper and lower bounds for the number of equivalence classes are computed . The lower bound is too big to handle all the equivalence classes in a hash table. In this case, a mixed approach, i.e., the use of an pattern-wise algorithm for 1-irregular elements with bisected edges and an algorithm that computes in real time the tessellation for the other 1-irregular cuboids, is recommended.

Keywords: control volume, cuboid tessellation, modified octrees

1. MOTIVATION

Since the last twenty years, modified octrees have been used very often in geometric modeling and mesh generation[11, 10]. The modified octree approach works as follows: The 3-D domain is enclosed in a cube, whose octants are repeatedly refined at their edge midpoints until the boundary and internal quantities are sufficiently approximated. Elements with and without edge midpoints are partitioned into tetrahedra. In case of using a octree based mesh generation for numerical methods, the final elements have to fulfill additional requirements.

Several aspects in the generation of meshes based on octrees and modified octrees have been already generalized in order to get a final domain representation that contains fewer basic elements than former approaches [5]: (1) The domain can be enclosed for a cuboid. A cuboid has rectangular faces. (2) The internal elements (macro-elements) can belong to a set of *well shaped elements*, such as pyramids, prisms and tetrahedra, and cuboids. The set of elements that is called *well-shaped* depends on the application. This set has to be closed under the *refinement operator*, i.e, each element can be refined in such a way that all newly generated elements belong to this set. The trees that can handle different element types as internal nodes are called *mixed element trees* [7]. (3) The refinement can be either bisection or what we have called the *intersection* based approach [6, 5]. Using the bisection based approach the refinement is always made at the edge midpoints. Using the *intersection* based approach the refinement is made at the most convenient edge point. The best point—the one whose associated refinement generates children with the smallest aspect ratio —is chosen from the available Steiner points (points generated by the refinement of the neighboring elements) and intersection points (points generated by the intersection between the object geometry and the current element). (4) Internal elements can be refined into a different number of elements and into elements of different type depending on the type of the internal node and on the refinement direction. For example, if a refinement is required along one, two, or three coordinate axes, cuboids, are subdivided into two halves, four quadrants, and eight octants, respectively. (5) The set of final elements is defined by the application. This set can be the set of macro-elements or a set composed of other element types. What we keep of the modified octree approach is the refinement parallel to the axes of the coordinate system.

The mesh generators known as Ω_{mebi} [6] and Ω_{mein} [5] have included several of the extensions mentioned above. Both mesh generators follow the same step sequence: (1) fit first exactly the object geometry (if possible) with a set of macro elements (cuboids, prisms, pyramids, and tetrahedra), (2) refine each element until the required mesh density is obtained, (3) generate a 1-irregular mesh (all the leaves are 1-irregular) that allows the generation of a Delaunay mesh by the union of the Delaunay tessellation of each leaf. (Note that in this case, the local computation of the 1-irregular elements must be done after the computation of the 2-D tessellations of the 1-irregular element faces, and after all these new faces fulfilled the empty sphere criterion.) and (4) generate the Delaunay mesh by computing the local tessellation of each leaf. The differences between both are that each macro-element is refined by bisecting its edges, while Ω_{mein} generates a nonconforming initial mesh where the macro-element edges can get several Steiner points at any position. The required density is obtained either bisecting the target edges or cutting the element at the position of one of the already inserted Steiner points.

The number of 1-irregular configurations depends on the element type (cuboid, prism, pyramid, etc) and on the refinement approach. The number of useful 1-irregular configurations, i.e, the ones that generate well-shaped final elements, depends also on the numerical method. In this case we consider that the final mesh is a Delaunay tessellation. Each co-circular(spherical) set of points is not divided into more simple elements, such that, tetrahedra, if it satisfies the Delaunay condition. An algorithm that tessellates any 1-irregular configuration into elements whose vertices are co-spherical was presented in [8]. That paper does not include any computation of the number of different 1-irregular configurations and equivalence classes that can be produced.

This paper presents a study of the number of different 1-irregular cuboids that can appear in mixed-element meshes generated by mesh generators Ω_{mebi} and Ω_{mein}, and recommends a way to handle them. It counts and finds all the equivalence classes for 1-irregular cuboids using a bisection based approach, and shows that is possible to find all the tessellations using a hash table (pre-computed tessellations). For the intersection based approach, it presents upper and lower bounds, and recommends the use of a mixed approach.

Whenever possible, the use of pre-computed tessellations as a method to find the tessellation of any 1-irregular element (independent of the algorithm used to generate them) should be preferred over other methods, because it is a robust method. It always computes the right tessellation and avoids precision problems.

2. BASIC CONCEPTS

Definition 1 *A d-cuboid is the notation for a cuboid of dimension d: 0-cuboid is point, 1-cuboid is a segment, 2-cuboid is a rectangle and a 3-cuboid is the cuboid (default).*

Definition 2 *A tessellation T of a set of points S is a* Delaunay tessellation *if there exists a point-free circumsphere for each final element.*

We use the term *Delaunay tessellation* and not *Delaunay triangulation* [3, 1, 4, 9] because our meshes include element types other than tetrahedra if their vertices are co-spherical. The most known of these elements are cuboids and some kinds of prisms and pyramids. Note that mesh generators based on octrees normally generate points that are not located in a general position, then it is possible to find many co-spherical configurations.

Delaunay tessellations are very useful in control volume methods that use the Voronoi region as integration volume. Co-spherical configurations (elements) that satisfy the Delaunay condition are not required to be tessellated into smaller elements because the numerical method only need the Delaunay edges with associated Voronoi edge in 2D (face in 3D) whose length (area) is not equal to 0.

The following definition introduces the concept of equivalence class and pattern type.

Definition 3 *Let c_1 and c_2 be two 1-irregular configurations, c_1 and c_2 belong to the same equivalence class if c_1 can be transformed to c_2 through rotations or reflections. The representative element of an equivalence class is called pattern type.*

1-irregular configurations that belong to the same equivalence class are partitioned in the same way. Each pattern type can have a bounded number of possible Delaunay tessellations depending on its edge length ratio. Figure 1(a) shows a 1-irregular rectangle where depending on ratio between w (its width) and h (its height), the vertices 5, 7 are connected (Figure 1 (b)), or vertices 4 and 6 are connected (Figure 1 (c)), or 4,5,6,7 are co-circular (Figure 1 (d)).

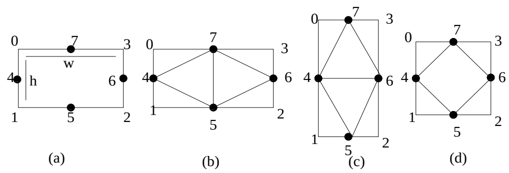


Figure 1. The tessellation of a 1-irregular rectangle with 4 Steiner points depends on the edge length ratio

2.1 Bisection Based Approach

Cuboids can be split into two halves, four quarters or eight octants as shown in Figure 2. The *Steiner* points defining a 1-irregular element are always located at the edge midpoints. In this case, the location of the *Steiner* points can be used to represent uniquely each 1-irregular cuboid.

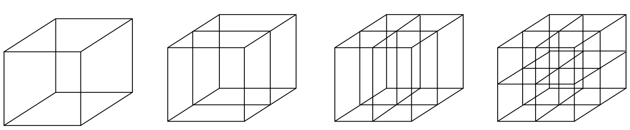


Figure 2. Bricks refined in one, two, or three directions generate two, four, and eight cuboids, respectively

Figure 3 shows several 1-irregular cuboids. The 1-irregular cuboid of Figure 3(b) can be transformed to the 1-irregular cuboid if Figure 3(a) by rotating it properly. We say then that the 1-irregular cuboids of Figure 3(a) and (b) belong to the same equivalence class. The two 1-irregular cuboids of Figure 3(c) and (d), respectively, have three bisected edges but they do not belong to the same equivalence class.

Conti [2] has already used the idea of equivalence classes in the implementation of a mesh generator based on modified octrees [11]. The information about the most common 1-irregular cuboids were stored in a hash table, whose hash function is a value obtained from a codification of the split edges. The edges are

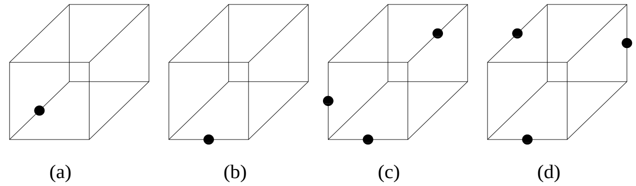


Figure 3. Different 1-irregular configurations

labeled in the order shown in Figure 4(a) and the vertices in the order shown in Figure 4(b). For each 1-irregular cuboid, the hash table stores the pattern type and the corner permutation to transform the current configuration to the configuration of the pattern type. For example, if Figure 4(c) is the pattern type for the 1-irregular elements with one split edge, the information stored in the hash table for the 1-irregular cuboid shown in Figure 4(d) is the bitcode of the pattern type (00000000001) and its corner permutation (1,4,0,5,2,7,3,6). Only the tessellation for the pattern type is computed and stored. The elements of the final tessellation were tetrahedra, pyramids, prisms and cuboids.

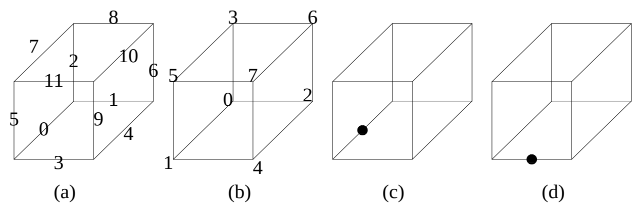


Figure 4. (a) Cuboid edge numeration, (b) cuboid vertex numeration, (c) one split edge pattern type, and (d) one split edge 1-irregular cuboid

The Conti's mesh generator only stored the twenty most used pattern types in a hash table. The time to find the tessellation of a 1-irregular cuboid that was stored was constant. But if the pattern type was not stored, new points were inserted until all the 1-irregular cuboids could be solved. This approach was extended for other element types in the implementation of a mixed element mesh generator [7].

2.2 Intersection Based Approach

Cuboids are split into two halves, four quarters or eight octants as before but edges are not necessary bisected. Figure 5 shows the different ways to split a cuboid using arbitrary refinement points. The only restriction is that parallel edges have to be split at the same relative position from their endpoints in order to generate cuboids and not general polyhedra.

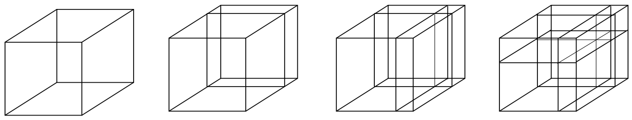


Figure 5. Bricks refined in one, two, or three directions generate two, four, and eight cuboids, respectively

During the tessellation of 1-irregular elements using a bisection-based approach, the type of the element, its aspect ratio, and the edges carrying a Steiner point are enough to identify uniquely a 1-irregular element. This condition does not hold for an intersection-based approach.

Figure 6 shows a set of 1-irregular cuboids with the same four split edges. Using a bisection-based approach, only the 1-irregular cuboid shown in Fig. 6(a) can occur. The four edges are bisected and the 1-irregular element is partitioned into two cuboids. Using an intersection based approach all these cases shown in Figure6 can occur.

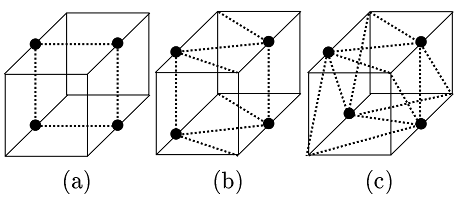


Figure 6. 1-irregular cuboids with the same split edges but different tessellations

In the event that Steiner points are located on orthogonal edges, (e.g., in a cuboid, at most three Steiner points), the tessellation is the same for both approaches: only the size of the final elements changes. Figure 7 shows two 1-irregular cuboids with the same Steiner-point but in a different position along the same edge. Both cases are tessellated in the same way.

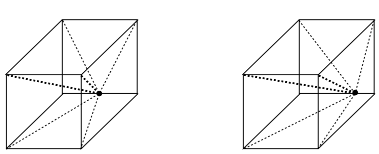


Figure 7. Tessellation of slightly different 1-irregular cuboids into four pyramids

3. 1-IRREGULAR CUBOIDS AND EQUIVALENCE CLASSES USING A BISECTION BASED APPROACH

It is already known that the total number of 1-irregular cuboids is 2^{12}. However, the number of equivalence classes or pattern types is not known. Its value is much lower than the total number of 1-irregular configurations as we will show in this section.

3.1 Theoretical Lower Bound

Theorem 1 *A d-cuboid has 2^d vertices and $d2^{d-1}$ edges.*

Proof. This is known and can be shown by induction.□

Theorem 2 *Let be a d-cuboid. Then, the number of 1-irregular configuration is $2^{d2^{d-1}}$.*

Proof. As we have said before, each edge can be bisected or not. Then, there are two possibilities for each edge (to have one or no one Steiner point) and so $2^{number\_of\_edges}$ possible 1-irregular configurations. Using theorem 2, a d-cuboid has $2^{d2^{d-1}}$ 1-irregular configurations.□

Corollary 3 *The total number of 1-irregular configurations is an upper bound of the number of equivalence classes.*

Theorem 4 *A lower bound for the number of equivalence classes in a d-cuboid of is $2^{d2^{d-1}-2d-\lg d}$*

Proof. The lower bound can be obtained considering that all the rotations and reflections are useful, i.e, each one transforms a different 1-irregular configuration into the pattern type.

(a) Each reflection divides the set of 1-irregular configurations into two parts. There is d reflections and therefore 2^d possible configurations generated using reflections.

(b) Using rotations, it is possible to bring any edge to a fixed edge. In addition, it is possible to chose two orientations. Then, the number of 1-irregular configurations that can be generated through rotations is twice the number of edges: $2d2^{d-1} = d2^d$

In the best case, the 2^d 1-irregular obtained after d reflections and the $d2^d$ configurations obtained after rotations are independents. The reduction factor is then $1/d4^d$ and the lower bound for the number of configurations is $2^{d2^{d-1}}/d4^d = 2^{d2^{d-1}-lgd-2d}$.□

In the case d = 3, the lower bound for the number of equivalence classes is 22. This means, there is at least 22 different pattern types.

3.2 Exhaustively Counting in Three Dimensions

In order to count exactly the number of equivalence classes, a program that generates the 4096 configurations , and checks which of them are equivalent, was developed.

The algorithm is very simple. For each one of the 4096 configurations, it generates all the possible combinations of rotations and reflections. The pattern type is the cube that has the lowest numerical representation. After applying this algorithm, 144 pattern types were obtained. (This number is the square of 12, the number of edges in 3-D. Then it could be expected that there is a relation between the number of pattern types and the number of edges. But in 2-D it can be easy shown that this is not true because there are 4 edges and only 6 pattern types (not 16)).

This algorithm can be used to generate automatically the corner permutations between any configuration and its pattern type, and hence to identify the right tessellation in O(1). The previous algorithm can be improved to reduce the number of superfluous rotations and reflections. But since this algorithm is used only once, when that hash table is initialized, its efficiency is not important. The tessellation of the pattern types can be fulfilled with an ad hoc algorithm as the one presented in [8].

3.3 Number of Delaunay Tessellation for Pattern Types

Theorem 5 *The number of possible Delaunay tessellation for each d-cuboid pattern type is bounded by* $F(d) = \prod_{d=1}^{d-1}(2d-1),\ F(1) = 1$.

Proof. As we have shown in Figure 1 using a 1-irregular rectangle, the Delaunay tessellation of a pattern type depends on its edge length ratio. The worst case is when there exists a different tessellation for each edge length ratio. In the case of a rectangle, edge lengths can vary in two directions: one edge length can be smaller than, equal to or greater than the other edge length. That is why it is possible to have at most three possible Delaunay tessellations for 1-irregular rectangle pattern type. In cuboids, the edge lengths can vary in three directions. The first edge length can be chosen in one way, the second edge length can be chosen smaller than, equal to or greater than the first edge length, and third, smaller than, in between, equal to or greater than the previous ones.

In general, the previous analysis can be described using the following expression. Let $F(d)$ be the maximum number of possible Delaunay tessellations for a d-cuboid pattern type. (The maximum value is obtained by considering that each edge length variation produces a new Delaunay tessellation). Then,

$$F(d+1) = (2d+1)F(d)$$

$$F(d+1) = \prod_{d=1}^{d}(2d+1),\ F(1) = 1$$

This formula can be shown by induction. It is easy to see, that if $F(d)$ is already computed, the new edge length can be the length of one of the previous edges (there are d possible lengths) or can be in between the previous ones (there are $d+1$ possible lengths). Then, the possible lengths in the new direction are $(2d+1)$. Therefore, the total number of edge length ratio in dimension $d+1$ is $(2d+1)F(d)$.

The number of possible tessellation of a 1-irregular cuboid is bounded by $F(3) = 15$. □

4. 1-IRREGULAR CUBOIDS AND EQUIVALENCE CLASSES USING AN INTERSECTION BASED APPROACH

The number of 1-irregular cuboids and the number of equivalence classes using an intersection based approach are still unknown. In this section, we will first define a new notion of equal 1-irregular configurations, and then compute the number of 1-irregular cuboids, and a theoretical upper and lower bound for the number of equivalence classes in 2D and 3D.

In order to generate the Delaunay tessellation of a 1-irregular configuration with Steiner points at any position, together with the element aspect ratio, the relative position of the Steiner points is relevant (see Figure 6 of section 2.2).

Definition 4 *A 1-irregular configuration* i_1 *is considered equal to a 1-irregular configuration* i_2 *if the relative position between the Steiner points located in the parallel edges of* i_1 *and* i_2 *is the same with respect to a normalized 1-irregular configuration.*

According to definition 4, the 1-irregular cuboids of Figure 8(a) and Figure 8(b) are equal and the 1-irregular cuboid shown in Figure 8(c) is not equal to the ones shown in Figure 8(a) and (b). The 1-irregular cuboid in Figure 8(d) is also not equal to the ones in Figure 8(a) and (b) but it belongs to their same equivalence class, because it can be considered equal to the ones in Figure 8(a) and (b) after two rotations about the y axis.

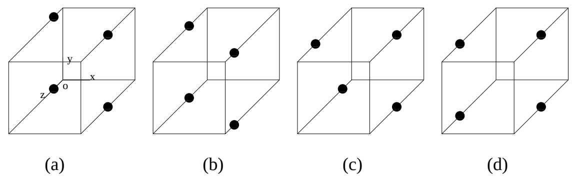


Figure 8. 1-irregular cuboids: (a) and (b) are considered equal, (c) is different from (a) and (b), and (a),(b) and (d) belong to the same equivalence class.

4.1 Number of 1-Irregular Configurations

This section introduces first several properties that simplify the computation of the number of 1-irregular cuboids, and then presents the results.

Proposition 1 *Let c be a d-cuboid and n the number of the 1-irregular cuboids with Steiner points in only one of the orthogonal axis. The total number of 1-irregular cuboids of c is n^d*

Proof. The computation of the total number of 1-irregular cuboids can be done by first counting the number of 1-irregular cuboids in each orthogonal direction independently. These numbers can then be multiplied together because the insertion of a new Steiner point has only an influence in the computation if it can be located to the left, right or on the same line of already inserted Steiner points. This occurs only in nonorthogonal edges. Since the cuboid has d orthogonal directions and has the same shape in each one, the total number of 1-irregular cuboids is n^d. □

Proposition 2 *Let n_i be, the number of 1-irregular d-cuboids with i Steiner points on parallel edges (only one of the orthogonal directions is used), then n is :*

$$n = \sum_{i=0}^{2^{d-1}} n_i$$

Proof. The parameter i is bounded by 0 and the number of parallel edges in any of the orthogonal axes of the d-cuboid. The number of parallel edges of a d-cuboid can be computed dividing the number of edges by the dimension. Using the theorem 1, the number of parallel edges is 2^{d-1}.□

Proposition 3 *Let c be a d-cuboid with i Steiner points on parallel edges. The number of locations to insert a new Steiner point along a target empty edge considering the already inserted points is $2i+1$.*

Proof. Each Steiner point can be inserted along a target empty edge to the left, to the right or aligned to one of the already inserted points. If there are i inserted points, the number of possible locations among the inserted points is $i+1$. In addition, the number of possible locations aligned to one of the inserted points is i. Then, the number of possible locations for the new point is $2i+1$.□

Theorem 6 *The number of 1-irregular rectangles are 6^2*

Proof. In 2D, the expression for n is the following:

$$n = \sum_{i=0}^{2} n_i$$

It can be easy shown that $n_0 = 1$, $n_1 = 1 \cdot 2 = 2$, $n_2 = 3 \cdot 1$ and $n = 6$ Then, the number of 1-irregular rectangles is $N = n^2 = 6^2$. □

Theorem 7 *The number of 1-irregular cuboids is $N = 187^3$*

Proof. In 3d, the expression for n is the following:

$$n = \sum_{i=0}^{4} n_i$$

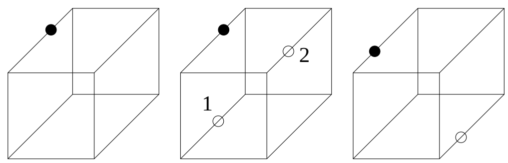
The next table shows the values of each n_i. Each n_i was computed separately using proposition 3:

i	n_i
0	1
1	4
2	18
3	60
4	105

Case $i = 0$: If there is no point inserted, $n_0 = 0$.

Case $i = 1$: If we insert one point in a cuboid with no point, once one edge is chosen, there is only one way to choose the location on it $(2 \cdot 0 + 1)$. Since there are four parallel edges, $n_1 = 4$.

Case $i = 2$: If we have already inserted one point on a cuboid edge (Figure 9(a)), once we choose the empty edge on which we will insert the next point, we have three possibilities $(2 \cdot 1 + 1 = 3)$. The empty edge can be chosen in two ways: (1) the new edge belongs to the same face of the previous edge (Figure 9(b)). Since there are four faces, this case produces $4 \cdot 3$ 1-irregular cuboids. (2) The new edge is opposite to previous one ((Figure 9(c)) . Since there are two ways to select opposite edges, this case produces $2 \cdot 3$ 1-irregular cuboids. Then, $n_2 = 18$.



(a) (b) (c)

Figure 9. (a) 1-irregular cuboids with one Steiner point, (b) and (c) possibilities of inserting a new point

Case $i = 3$: We already know that the first point is inserted in one way, the second point in three ways and the third point in 5 ways, then once we have chosen the three edges we have $1 \cdot 3 \cdot 5 = 15$ 1-irregular cuboids. Since there are 4 ways to choose the three involved edges, $n_3 = 60$.

Case $i = 4$: Each edge get a point. The number of 1-irregular cuboids is $1 \cdot 3 \cdot 5 \cdot 7 = 105$.

$$n = (4 + 18 + 60 + 105) = 187; N = 187^3 \square$$

4.2 Number of Equivalence Classes

An upper bound for the number of equivalence classes is the number of 1-irregular cuboids.

In the same way as in the bisection based approach, a lower bound for the equivalence classes can be obtained if the total number of 1-irregular cuboids is divided by the number of possible rotation and reflection transformations.

Corollary 8 *Let N be the number of 1-irregular d-cuboids. A lower bound for the number of equivalence classes of 1-irregular d-cuboids is* $\frac{N}{d 4^d}$. *In the particular case of the cuboid, the value is* $\frac{187^3}{192} > 34,058$

5. CONCLUSIONS

This paper presents the computation of the exact number of 1-irregular rectangles and cuboids for both a bisection and an intersection based approach. In case of the bisection based approach, it presents the theoretical computation of upper and lower bounds, and the empirical computation of the exact number of equivalence classes. In case of an intersection based approach, it presents the theoretical computation of the upper and lower bounds for the number of equivalence classes.

The number of equivalence classes of a cuboid in a bisection based approach is 144. This allows us to store the necessary information of all the 1-irregular cuboids (2^{12}) and the tessellation of all the pattern types in a hash table. Then, the time to get the right tessellation of any 1-irregular cuboid is O(1).

The number of equivalence classes of a cuboid using an intersection based approach is too high for storing all of them in a hash table. It is also not clear if there exists a good hash function, because the relative position between Steiner points on parallel edges should also be considered. Since the mesh generator that uses an intersection based approach to fit the device geometry, refines the coarse elements by bisecting their edges wherever required, most of the 1-irregular elements have bisected edges. Then, it is convenient to use a mixed approach, i.e, an hash table for 1-irregular elements with bisected edges and an algorithm for the rest of 1-irregular elements.

The use of pre-computed tessellations as a method to find the tessellation of any 1-irregular element (independent of the algorithm used to generate them) should be preferred over other methods, because it is a robust method (for example, it avoids the precision problems that can occur, when 1-irregular elements belong to very thin layers). In addition it computes always the right tessellation and take less computational time than a real time algorithm.

ACKNOWLEDGMENT

This work was supported by Fondecyt project No 1960735.

REFERENCES

[1] Marshall Bern and David Eppstein. " *Mesh Generation and Optimal Triangulation.* " Palo Alto Research Center. Xerox, March 1992.

[2] P. Conti. " *Grid Generation for Three-dimensional Device Simulation.* " PhD thesis, ETH Zürich, 1991. " published by Hartung-Gorre Verlag, Konstanz, Germany.

[3] B. Delaunay. " Sur la sphère vide. " *Bull. Acad. Sci. USSR(VII)*, pages 793–800, 1934.

[4] P L George and H Borouchaki. " *Delaunay Triangulation and Meshing.* " Hermes, 1998.

[5] N. Hitschfeld. " Generalization of Modified Octrees for Geometric Modeling. " In *Geometric*

Modeling: Theory and Practice. The state of the art, pages 260–272. Springer-Verlag, 1997.

[6] N. Hitschfeld. " *Grid Generation for Three-dimensional Non-Rectangular Semiconductor Devices.* " PhD thesis, ETH Zürich, Series in Microelectronics, Vol. 21, 1993. " PhD thesis published by Hartung-Gorre Verlag, Konstanz, Germany.

[7] N. Hitschfeld, P. Conti, and W. Fichtner. " Mixed Elements Trees: A Generalization of Modified Octrees for the Generation of Meshes for the Simulation of Complex 3-D Semiconductor Devices. " *IEEE Trans. on CAD/ICAS*, 12:1714–1725, November 1993.

[8] N. Hitschfeld and R. Farías. " 1-irregular element tessellation in mixed element meshes for the control volume discretization method. " In *Proceedings of the 5th International Meshing Roundtable*, pages 195–204. Pittsburgh, Pennsylvania, U.S.A., October 10-11, 1996.

[9] B. Joe. " Three-dimensional triangulations from local transformations. " *SIAM J. Sci. Stat. Comput.*, 10:718–741, 1989.

[10] Mark S. Shephard and Marcel K. Georges. " Automatic Three Dimesional Generation by the Finite Octree Technique. " In *International Journal for Numerical Methods in Engineering*, volume 32, pages 709–749, 1991.

[11] M. A. Yerry and S. Shephard. " Automatic Three-dimensional Mesh Generation by the Modified-Octree Technique. " *Int. J. Numer. Methods Eng.*, 20:1965–1990, 1984.