Advanced Automatic Hexahedral Mesh Generation from Surface Quad Meshes

Michael Kremer[1], David Bommes[2], Isaak Lim[1], and Leif Kobbelt[1]

[1] Computer Graphics Group, RWTH Aachen University, Germany
{mkremer,isaak.lim,kobbelt}@cs.rwth-aachen.de
[2] INRIA Sophia Antipolis - Méditerranée, France
david.bommes@inria.fr

Summary. A purely topological approach for the generation of hexahedral meshes from quadrilateral surface meshes of genus zero has been proposed by M. Müller-Hannemann: in a first stage, the input surface mesh is reduced to a single hexahedron by successively eliminating loops from the dual graph of the quad mesh; in the second stage, the hexahedral mesh is constructed by extruding a layer of hexahedra for each dual loop from the first stage in reverse elimination order. In this paper, we introduce several techniques to extend the scope of target shapes of the approach and significantly improve the quality of the generated hexahedral meshes. While the original method can only handle "almost convex" objects and requires mesh surgery and remeshing in case of concave geometry, we propose a method to overcome this issue by introducing the notion of concave dual loops. Furthermore, we analyze and improve the heuristic to determine the elimination order for the dual loops such that the inordinate introduction of interior singular edges, i.e. edges of degree other than four in the hexahedral mesh, can be avoided in many cases.

1 Introduction

Many methods of computer driven simulation like the widely used finite element analysis are based on discretizations of a given volumetric domain. Usually, CAD programs are used to design surface meshes representing the boundary of these domains. In most cases a decomposition of the enclosed volume into discrete elements is neither obvious nor computationally simple to determine. Some volumetric meshing approaches rely on triangulating the CAD surface mesh first and computing a volumetric mesh through Delaunay tetrahedralization of the inner volume afterwards [12, 30]. Nevertheless, in practice, using hexahedra as the discretization element of choice offers some important advantages, also see e.g. [29]. The most important benefits are:

- Numerical computations on hexahedral meshes are up to 75% less memory and time consuming in comparison to tetrahedral meshes [34].

- The loss of numerical accuracy when non-uniformly scaling hexahedra is significantly smaller compared to tetrahedra [3].
- Hexahedral meshes are between four to ten times less complex with respect to the number of elements than tetrahedral meshes (with the complexity of the input mesh being constant) [3].

Obviously, the boundary of a 3D hexahedral complex is a closed all-quadrilateral surface mesh. So, a first approach would suggest to quadrangulate the boundary first and generate a surface-conforming hexahedral mesh afterwards. Using all-quadrilateral surface meshes as input data comes with the advantage of being more intuitive during the process of design. Moreover, it offers the opportunity to gain higher approximation quality of a polynomial surface by aligning the edges with the principal curvature directions which are orthogonal [32].

However, generating hexahedral meshes of valid topology and practical geometric properties from quadrilateral surface meshes of arbitrary genus and possibly self-intersecting loops of quadrilaterals is unresolved in general. In [23] Müller-Hannemann proposes a combinatorial method for the generation of hexahedral complexes from quad meshes that are of genus zero and do not contain self-intersecting quad loops. At a first glance, this appears to be a serious restriction of the class of hex-meshable surface meshes. Nevertheless, recent approaches to quad meshing significantly increased the quality of quadrangulations of surfaces while still offering a decent amount of control over the parameters [5, 7, 10, 15, 36, 6].

In the method from Müller-Hannemann the dual graph of a surface quad mesh is reduced to a topological cube by successively eliminating cycles from it. Each cycle can be interpreted as to enclose a *sheet*, i.e. a layer, of hexahedra [24], depicted in Figure 1. The hexahedral complex is then built by starting with a single hexahedron and topologically "extruding" the eliminated sheets in reverse elimination order. In a subsequent step, the geometric embedding of the mesh is computed using the *Mesquite* library [9] which offers methods to optimize the geometry of hex meshes. While the method works appropriately on *almost* convex objects, the author suggests to decompose non-convex objects into convex ones and remesh the interfaces accordingly [23, 22]. However, cutting and remeshing the surface meshes is generally not desirable since it either requires the user to know the interior topology of the decomposition and/or might compromise global topological properties of the initial surface quadrangulation.

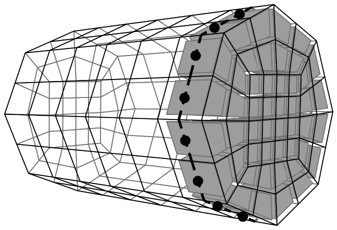

Fig. 1. The boundary complex of a hexahedral mesh with a dual cycle and its corresponding sheet (gray hexahedra).

1.1 Contribution

In this paper, we present an extension to the method described in [23] that allows us to automatically generate surface conforming hexahedral meshes for non-convex objects of genus zero. For this, we first analyze how the order in which the dual cycles are eliminated from the dual graph influences the resulting mesh's topology and propose some practical guidelines for many common mesh configurations in Section 4. Furthermore, for the handling of concave objects, we introduce the notion of *concave dual cycles* and the significance of their elimination from the dual graph in Section 5. From this analysis we derive a set of rules that, applied to the heuristic that determines the elimination order of the dual cycles, improves the topological quality of the resulting mesh. Since the basic concepts described in [23] are provided with a rather theoretical context, we briefly outline the basic ideas in a more practical and implementable fashion in Section 2 and 3. However, we focus on the concepts needed in order to understand the presented method and deliberately omit certain details that go beyond the scope of this work. Although we admit that our approach is still of rather academic value, the class of objects that are meshable with our method is considerably extended as compared to the original method.

1.2 Related Work

Apart from the mentioned (dual) topological hexahedral meshing approaches there exists a large body of research on various other meshing techniques.

For a more complete overview on hexahedral meshing techniques we refer to [3]. In the following, we only provide a brief overview of the most related existing hex-meshing approaches. In [16, 31] hexahedral mesh primitives are built by sweeping a surface mesh along an arbitrary path generating hexahedral meshes with constant cross-section topology. The generated hexahedral meshes are boundary sensitive. A major drawback of this method, however, is the high manual effort required to mesh objects of complex shape.

A further class of meshing approaches generate hexahedralizations by decomposing the original surface mesh into several simpler units. This can either be performed using geometric decompositions as in [1, 20, 28], or integrally during the meshing processing by using an interior mesh as the cutting mechanism as in [21, 13, 2, 35]. These methods have the advantage that they are mainly boundary sensitive. However, a common drawback of all these methods is the need for a (manual) multilevel shape detection for the decomposition as well as the disparity of the meshed cut areas between the components.

In recent years, a considerable amount of effort has been put into the development of parameterization-based hex-meshing approaches [19, 14]. In these methods, the 3D domain enclosed by a (closed) two-manifold is tetrahedralized first. Then a sufficiently smooth field of rotation symmetric 3D crosses is generated over the domain. These fields are then used to extract the hexahedral mesh [26]. However, these methods are still subject to certain limitations: the automatic generation of 3D cross fields with valid singularity structures is still unsolved [19]. Thus, a certain amount of manual interaction is potentially needed in order to meet the requirements of the mesh extraction routine. Poorly designed cross fields often lead to inverted elements and the existence of a topologically valid hexahedral mesh can generally not be guaranteed. Therefore, parameterization-based methods oftentimes only generate *hex-dominant* meshes.

Another class of hexahedral meshing techniques are the so-called *advancing front techniques*. They are designed to generate a hexahedral mesh from the boundary of the surface mesh inward [4, 18]. In these methods, most singularities are propagated to the medial axes of of the objects which might also lead to poorly shaped elements. As a further development to this approach, the *whisker weaving* method [33, 11] converts the surface mesh to its dual and then generates a connectivity of “unknown” hexahedral elements in the enclosed 3D domain. An extension to this described in [17] also incorporates geometric information to overcome some geometric issues. The method proposed in this paper is also an advancing front technique. The advantage of this class of methods is the boundary sensitivity, i.e. the topology and geometry of the boundary surface remains fixed. In these cases, many numerical applications benefit from higher accuracies at the boundary regions. In the proposed method we consider entire layers of hexahedra instead of single elements. These global operations effectively avoid poor quality near the medial axes as often can be observed in local schemes.

2 Prerequisites and Terminology

In this section we will give a brief outline of the prerequisites necessary in order to understand the proposed method.

2.1 Combinatorial Quad Meshes

Let $G = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ denote a closed and two-manifold all-quadrilateral mesh with vertices $\mathcal{V}$, edges $\mathcal{E}$, and faces $\mathcal{F}$. The mesh's geometry is defined via a function $p : \mathcal{V} \mapsto \mathbb{R}^3$, the so-called *embedding*. The topology of G is completely defined by $\mathcal{E}$ and $\mathcal{F}$. Let G^* denote the *dual* mesh of G. We henceforth refer to G as the *primal* mesh. G^* usually does not have a geometric embedding and can be considered a *combinatorial graph*. A dual edge connects two dual vertices if and only if the corresponding primal faces are adjacent, i.e. they share a common edge. See Figure 2 for an example of a simple geometric quad mesh and its combinatorial dual mesh. Since G is a pure quad mesh, all

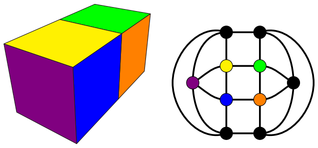

Fig. 2. Left: A simple geometric quad mesh. Right: Its combinatorial dual mesh. Corresponding entities are marked with the same color.

vertices of G^* have valence four. Two adjacent dual edges are said to be part of the same dual *cycle* if their respective primal edges oppose each other in the shared primal face. All other dual edges in that same cycle are determined through transitivity of this adjacency relation. A dual mesh can be uniquely decomposed into a set of pairwise disjoint cycles in linear time. Observe that dual cycles can be *self-intersecting*, i.e. all four dual edges incident to one dual vertex may belong to the same cycle. All non-self-intersecting cycles are called *simple*. Two distinct cycles cross each other if they share a common vertex. All cycles be arbitrarily but consistently oriented.

3 Dual Graph Reduction

Consider the input quad mesh to be the boundary complex of an "unknown interior hexahedral complex". Following the ideas of Murdoch et al. [24], each dual cycle encloses a so-called *sheet*, i.e. a layer of hexahedra (Fig. 1).

3.1 Dual Cycle Elimination

Let G^* be the dual graph of an all-quadrilateral surface mesh. Consider C to be some dual cycle in G^*. C can be eliminated by deleting each edge in C from G^*. Afterwards, all dual vertices incident to the deleted edges are also removed and each pair of the remaining edges formerly incident to these vertices is merged. See Figure 3 (a) for a simple example.

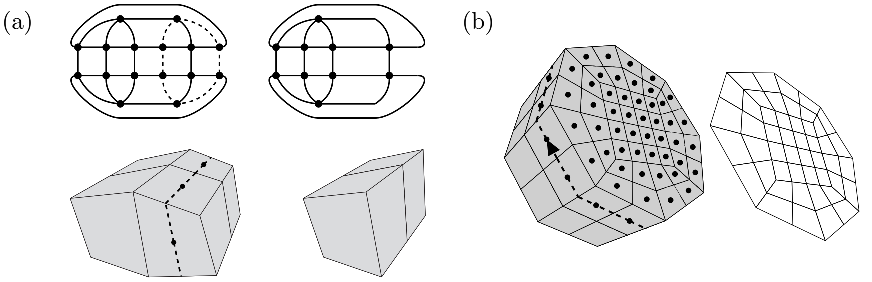

Fig. 3. (a) The elimination of a cycle from a dual graph (upper row) and its corresponding elimination in the primal mesh (lower row). The cycle in question is marked with a dashed line. The resulting graphs after the removal of the cycle are shown in the second column, respectively. (b) A dual cycle (dashed lines) on a surface mesh. The dots on the planar area enclosed by the cycle on its right side, denoted by $V^*_{C,r}$, induce a connected subgraph of the dual surface graph. The dualization of this subgraph corresponds to the primal surface sub-mesh depicted on the right.

Let C be a dual cycle in G^* with a consistent orientation and no self-intersection. Then C splits G^* into two connected components on either side of C (since the mesh has ball topology) which we denote by $V^*_{C,\mathrm{l}}$ and $V^*_{C,\mathrm{r}}$. In this notation, l and r refer to the left or right hand side of C with respect to its orientation. The dualization of each subgraph induced by the $V^*_{C,l}$ and $V^*_{C,r}$ is a connected part of the primal surface. See Figure 3 (b) for an illustration.

The elimination of a dual cycle can be interpreted as the elimination of an entire layer of hexahedra by applying a series of local graph transformations—illustrated in [23]—that successively "take away" hexahedra on the *boundary* of the unknown hexahedral complex. In this approach, we define that the elements to be eliminated from the unknown hexahedral complex are *always* situated on its surface. This accounts for the fact that the surface is the only region of the unknown complex the topological structure of which is known to us. Therefore, the eliminated sheet can either be situated to the left or to the right side of C (with respect to C's orientation). In this context, let $V^*_{C,\mathrm{l}}$ and $V^*_{C,\mathrm{r}}$ denote the set of dual vertices situated on the left and right side of C, respectively, and let V^*_C denote the vertices incident to C. Then, when eliminating the sheet to e.g. the right side of C, each hexahedron in this sheet is incident to *exactly* one primal quadrilateral in $V^*_{C,\mathrm{r}}$ and the entire sheet is bounded by the primal quadrilaterals that correspond to the V^*_C. See Fig. 3 (b) for an illustration of a dual cycle and the set of dual vertices to its right side and Fig. 1 for the corresponding sheet of hexahedra. Consequently, for each dual vertex in $V^*_{C,\mathrm{l}}$, $V^*_{C,\mathrm{r}}$ respectively, exactly *one* hexahedron is taken away from the unknown hexahedral complex. Whether a dual cycle is eliminated with respect to its left or right side actually does not make any difference during the elimination process. However, we will see that it has a significant influence on the resulting hexahedral mesh's topology when it comes to the reconstruction step explained in Section 3.3, also refer to Figure 5 to get a picture of the described elimination scheme.

3.2 Perfect Cycle Elimination Schemes

According to Müller-Hannemann, an elimination of a dual cycle C is called *feasible* if

1. C is a simple cycle, i.e. it is free of self-intersections,
2. the subgraph of G^* induced by the dual vertices incident to the virtually eliminated sheet on the respective side of C, thus $V^*_{C,\mathrm{l}}$ or $V^*_{C,\mathrm{r}}$, is connected[1], and
3. The dual graph after the elimination of C, denoted as G'^*, is 3-*vertex-connected*, i.e. it has at least four edges and cannot be disconnected by removing one or two vertices from the graph. This can be checked in linear time complexity, see [27] for details.

The general idea behind the meshing algorithm is to successively eliminate outer layers of hexahedra, i.e. "peel off" sheets on the boundary, from the unknown hexahedral complex. For this, a series of feasible dual cycle eliminations

[1] An undirected graph is connected if there is a path between any pair of nodes.

that transforms the dual surface graph of the input mesh into the dual graph of a single hexahedron has to be found. This requires the elimination of all dual cycles except three from G^* in a specific order. Such a series of feasible dual cycle eliminations is called *Perfect Cycle Elimination Scheme*, PCES in short. A PCES is generally not unique.

Obviously, a single hexahedron is not only the simplest hexahedral mesh but also a convex polyhedron the dual graph of which has a planar embedding and consists of three simple cycles. Therefore, the given criteria that have to be satisfied in order for a dual cycle elimination to be feasible must not be violated at any stage when determining a PCES for a given input mesh. In order to determine the order of dual cycle eliminations, we have to examine the reconstruction of cycle eliminations and the geometric classification of the cycles more closely.

3.3 Reconstruction of Dual Cycle Eliminations

The reconstruction starts with a hexahedral complex which only consists of one hexahedron. We now process a PCES in *reverse* order. Let C_i be the dual cycle to be reconstructed and let furthermore C_i be eliminated with respect to side $s \in \{l, r\}$ (which, at this point, is still unclear how to choose). Then $V^*_{C_i,s}$ denotes the set of dual vertices situated on side s of C_i w.r.t. C_i's orientation. The elimination of C_i is then reconstructed by *extruding* all (connected) faces in the intermediate surface mesh G' that correspond to the $V^*_{C_i,s}$ (cf. Figure 3 (b), right). In this context, an extrusion is the topological operation of transforming each boundary face in question to a hexahedron being adjacent to the hexahedron formerly adjacent to the face. Also see Figure 4 for an illustration of this operation.

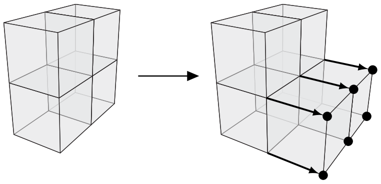

Fig. 4. The extrusion of the lower row of two faces.

Although the decision of whether a cycle should be reconstructed to its left or right side is irrelevant for the cycle elimination procedure, it obviously plays a crucial role during the reconstruction steps. Since a dual cycle elimination is reconstructed by extruding all primal faces corresponding to the dual vertices on one side of the cycle, it is desirable the cycle be situated at geometric borders, i.e. near geometric features, and the elimination side be chosen the feature side. If the elimination side is not chosen accordingly, the reconstructed hexahedral mesh might contain unnecessary interior singular edges, i.e. inner edges of degree other than four in the hexahedral mesh. In many cases, singular edges are not desired since they might cause the hexahedral elements to suffer from compromised numerical properties. See Figure 5 for an illustration of the two different reconstruction schemes and the resulting hexahedral meshes after the reconstruction. In order to avoid the generation of these undesired

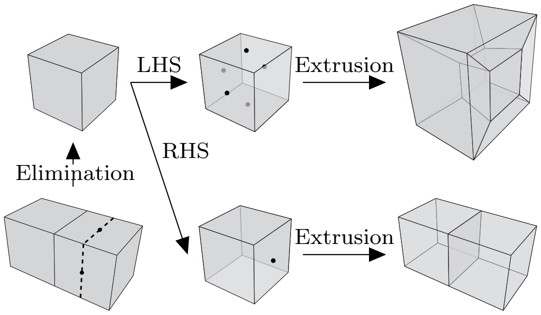

Fig. 5. Left: A dual cycle (dashed line) is eliminated from a surface mesh. Top row: The elimination is reconstructed with respect to the cycle's left-hand side by extruding the faces marked with a dot. Bottom row: The elimination is reconstructed with respect to the cycle's right-hand side. The resulting hexahedral meshes are depicted in the right-most column. Although both meshes have the same boundary topology, they essentially differ in their interior topology. While the mesh on top has interior singular edges, the bottom mesh does not.

configurations, a geometric classification of the dual cycles used to introduce a basic elimination prioritization is described in the next section.

3.4 Geometric Rating of Dual Cycles

In order to use the geometry of a surface mesh G to guide the dual cycle elimination process, Müller-Hannemann proposes to assign each dual edge a binary status: *sharp* or *flat*. For this, we measure the (unsigned) dihedral angle of the corresponding primal edge outside the solid for each dual edge and determine whether it exceeds a user-defined threshold—experiments proved values between 50° and 70° deviation from flat as being reasonable. If it exceeds this threshold it is marked as sharp, otherwise flat. In the following, let V_C^* denote the set of dual vertices in G^* incident to the dual edges of a cycle C. Then, in a subsequent step, each dual cycle is assigned three values:

Left-Hand Side Weight Iterate over the oriented edges of C and accumulate the number of sharp dual edges incident to a dual vertex in V_C^* and situated on the *left-hand side* of the cycle. The final weight is the quotient of this number and the total number of dual edges in C.

Right-Hand Side Weight The same as before but this time for the right-hand side of C.

Flatness Weight Accumulate the number of sharp dual edges in C.

We can now use the first two weights to incorporate the underlying geometry in the heuristic to determine the PCES: the elimination priority increases proportionally with the value of the highest of either weights. The elimination side is then the one whichever has the higher weight. The third weight is used

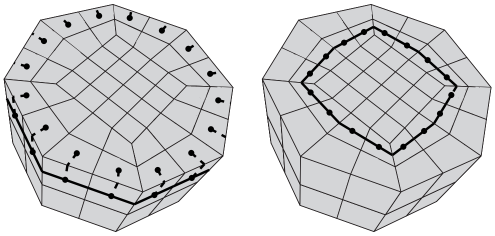

Fig. 6. Left: A dual cycle on a surface mesh (marked with thick solid lines). The dashed lines are the neighboring dual edges to one of the cycles sides (with respect to its orientation). In this example, the corresponding primal edges all have a sharp dihedral angle and thus the presented dual cycle's weight is high. Right: An illustration of a flat cycle on a planar surface region. The elimination of these cycles should be avoided.

to classify a cycle as either flat, in case this weight is zero, or non-flat, in case this weight is positive. The elimination of dual cycles with flatness weight zero should be avoided since they are situated within a flat region and thus enclose hexahedral sheets that have zero volume and are therefore geometrically infeasible. An illustration of a sharp and a flat dual cycle arrangement is given in Figure 6. Notice that, when eliminating a dual cycle, as described in Section 3.1, the weights of each merged edge is just the sum of the weights of both former edges. Since we solely operate on the dual graph we do not care about updating the geometric embedding of G during the elimination steps.

4 Dual Cycle Elimination Optimizations

In this section we describe different classes of problems commonly encountered with the original approach and discuss some tools to overcome them. These techniques are presented in terms of simple rules that can easily be integrated into the heuristic to determine the dual cycle elimination order.

4.1 Zero Volume Hexahedra

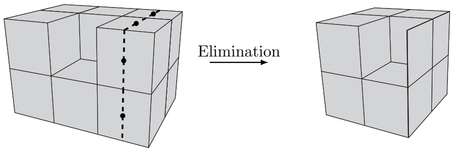

Fig. 7. A simple example of a surface mesh with a concave area. The elimination of the cycle marked as dashed lines will result in a surface mesh that induces at least one hexahedron with zero volume.

The main problem that occurs when reconstructing objects with concavities is that, at some point during the reconstruction, poorly shaped hexahedra, i.e. hexahedra with small or negative Scaled Jacobian value, will be added to the hexahedral complex. This is always the case if the sum of the dihedral angles of at least two successive crossing dual edges adjacent to C is close to 180°. The problem is depicted in Figure 7 at the example of a simple surface mesh configuration.

These configurations are easy to detect. During the elimination process, we merely have to make sure that no such dual cycle is eliminated as long as there are still other candidates that do not contain such configurations—even if their elimination weight is smaller. In the example of Figure 7, this would lead to the elimination of the cycle horizontally enclosing the top layer of hexahedra prior to the depicted one leading to the resulting hexahedral mesh this surface intuitively suggests.

4.2 Enclosed Dual Cycles

Let C be a dual cycle to be reconstructed with respect to side $s \in \{l, r\}$. Then, the primal faces corresponding to $V^*_{C,s}$ define a patch on the primal quadrilateral surface mesh representing the boundary of the intermediate hexahedral mesh as depicted in Figure 3 (b). Let this mesh be denoted by $Q := Q(C, s)$. Since we extrude one hexahedron for each dual vertex in $V^*_{C,s}$, a connected sheet of $|V^*_{C,s}|$ hexahedra is created when extruding the patch Q. Therefore, the degree (or oftentimes referred to as *valence*) of each entity in Q is increased by one. Note that an edge on the boundary of a hexahedral complex is singular if it is incident to other than three faces. Analogously, an interior edge is singular if it is incident to other than four faces. When extruding Q, we can now observe the following (the described configurations refer to the meshes *before* the extrusion):

(1) For each vertex on the boundary of Q that is incident to other than *three edges* in Q a new singular boundary edge is added to the hex complex.
(2) For each vertex in the interior of Q that is incident to other than *four edges* in Q a new singular edge is created in the interior of the hex complex.
(3) Each edge on the boundary of Q that is incident to other than *two faces* in the intermediate hexahedral complex becomes a singular boundary edge.
(4) Each edge in the interior of Q that is incident to other than *three faces* in the intermediate hexahedral complex becomes a singular interior edge.

Obviously, during the elimination of the dual cycles, we do not have any knowledge about the topology of the intermediate hexahedral mesh obtained in the reconstruction steps. For this reason, it is not possible to predict configurations of case (3) or (4) and thus they cannot be respected when eliminating the cycles. However, it is possible to detect configurations of type (1) and (2).

Let C be a dual cycle to be eliminated at elimination step i with respect to side $s \in \{l, r\}$. Furthermore, let G^*_{i-1} denote the intermediate dual surface graph after $i - 1$ cycle eliminations. The set $V^*_{C,s}$ induces a connected subgraph of G^*_{i-1} denoted by $G^*_{C,s,i-1}$. Let n be the total number of dual cycle eliminations. Since the described method is surface conforming the dual surface graph after i cycle eliminations is always topologically equivalent to the surface of the intermediate hexahedral mesh after $n - i$ reconstruction steps. Accordingly, if $G^*_{C,s,i-1}$ contains at least one closed dual cycle C_{enc} during the elimination, i.e. C *encloses* C_{enc} on side s, we can conclude that the corresponding surface of the intermediate hexahedral mesh contains at least four primal vertices of degree three, thus of type (2). If, in addition, the lengths of C_{enc} and C differ, the intermediate hexahedral mesh's surface contains at least one more vertex of type (1) and/or type (2). It turns out that, in practice, these configurations, illustrated in Figure 8, are encountered considerably often.

Therefore, we extend the cycle elimination selection routine such that if a cycle C encloses another non-flat cycle C_{enc} to its elimination side, then C_{enc} must be eliminated first. This minimizes the number of singular vertices within the extrude region and thus the number of singular edges added to the hexahedral complex. In case C_{enc} is a flat cycle, we must not eliminate it prior to C because C_{enc} might be part of a so-called *double cycle arrangement*, see [23] for details. In these cases, it is recommended to proceed with the next elimination candidate if there is any.

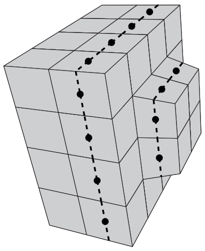

Fig. 8. A simple surface mesh with two cycle elimination candidates (dashed lines). Eliminating the cycle on the left side prior to the right one would introduce unnecessary interior singular edges into the resulting hexahedral mesh.

5 Concave Cycle Elimination

The method to avoid degenerate hexahedra to be created, as discussed in Section 4.1, works sufficiently well for meshes with "notches" such as depicted in Figure 7 but is insufficient when it comes to meshes that contain closed concave regions such as cavities. Consider the two meshes depicted in Figure 9 (a). In this example, any cycle elimination order of the mesh on the left side would obviously cause degenerate hexahedra to be generated during the reconstruction. Consequently, in order to cover these surface mesh configurations, a more sophisticated technique is needed. In order to find a practical means to overcome this issue, we divide the cycles into two subclasses referring to their geometric properties: convex and concave dual cycles. All concave dual cycles are then eliminated prior to the elimination of the convex ones. The informal interpretation of the elimination of concave cycles is to topologically *add* a virtual sheet of hexahedra to the "unknown hexahedral complex" that is situated *outside* the volume bounded by the surface for each eliminated cycle. The introduction of virtual layers of hexahedra then successively "stuffs" the concavity so that, after the elimination of all concave cycles, we end up with a modified surface mesh that is (almost) convex. See Figure 9 (b) for an illustration of the topological interpretation.

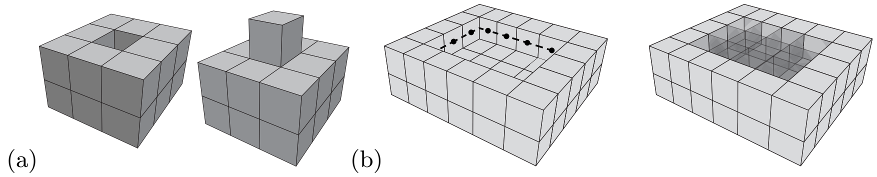

Fig. 9. (a) Although both meshes have the exact same topology, the left one does not induce an elimination scheme that avoids the generation of degenerate hexahedra, whereas the mesh on the right can be meshed using only well-shaped hexahedra. (b) Left: A surface mesh with a concavity in its center. The dual cycle marked with dashed lines is identified as a concave cycle. Right: This concave cycle can be interpreted as enclosing a sheet of virtual hexahedra outside the volume bounded by the surface mesh filling the cavity (marked as transparent cells). The elimination of this cycle *adds* the marked hexahedra to the unknown hexahedral complex and results in a surface graph in which the concavity is removed.

5.1 Classification of Concave Dual Cycles

In the following, we require the input surface mesh to be consistently oriented. In our implementation, we use the OpenMesh half-edge data structure [8]. At the basis of this orientation we can now determine whether the dihedral angle of two adjacent primal faces is convex, i.e. the angle is positive, or concave, i.e. the angle is negative. We now extend the set of weights computed for each dual cycle. For each dual cycle a *convexity weight* is computed for each side similar to the rules described in Section 3.4. This time, the weight is set to be the quotient of *convex* sharp edges and the total number of dual edges in the cycle. Similarly, we now introduce the *concavity weight* for each side which is the quotient of *concave* sharp edges and the number of edges in the cycle.

5.2 Elimination Scheme

We now extend the heuristic to determine the dual cycle elimination order in a straightforward way. All dual cycles with a positive concavity weight on either side must be eliminated prior to the ones with positive convexity weights. Once again, the elimination priority should be proportional to the value of the concavity weight. As a further rule, the elimination of a dual cycle with high LHS *and* RHS convexity weight must be avoided as long as there is any other elimination candidate in the graph. This is due to the fact that it is highly likely that this cycle contains zero volume configurations such as described in Section 4.1. Once all concave dual cycles are eliminated, we proceed with the elimination of the convex dual cycles.

5.3 Reconstruction Scheme

After the elimination of all concave dual cycles the unknown hexahedral complex contains additional hexahedra making the complex convex. After applying the reconstruction step of the convex dual cycle eliminations in reverse order, the reconstructed hexahedral complex contains these additional hexahedra used to fill the concavities. Consequently, as a last reconstruction step, we reconstruct the elimination of the concave dual cycles in reverse elimination order. Let C be a concave dual cycle to be reconstructed to side $s \in \{l, r\}$. This time, instead of extruding the connected region induced by $V^*_{C,s}$ on the hexahedral mesh's surface, we apply an *intrusion* of the layer of hexahedra incident to the primal faces corresponding to the $V^*_{C,s}$ in the intermediate hexahedral mesh. An intrusion is the opposite operation of an extrusion: all hexahedra incident to the primal faces corresponding to the $V^*_{C,s}$ are deleted from the hexahedral complex. More formally, let $\mathcal{H}_{C,s}$ be the set of hexahedra being incident to the primal faces corresponding to $V^*_{C,s}$. Then, when intruding $\mathcal{H}_{C,s}$, all entities of the intermediate hexahedral complex of any dimension that *only* have incidence relations to elements in $\mathcal{H}_{C,s}$ are deleted.

5.4 Problems

Although the elimination and reconstruction of concave cycles is a straightforward extension to the schemes described by Müller-Hannemann that extends the class of meshable objects, there are still scenarios that cause problems.

Insufficient Cycle Alignment

The idea of classifying dual cycles as either concave or convex requires that the cycles be aligned with the feature (curvature) lines of the surface mesh. Therefore, it is understood that finding a topologically simple as well as geometrically meaningful quad mesh layout is crucial for generating high quality hexahedral meshes. In case the cycle layout's alignment diverges significantly from the mesh's main features, the resulting hexahedral mesh will presumably contain undesired degenerate elements. This also applies for the alignment of the convex dual cycles.

False Positives

In some cases it occurs that a concave cycle is not part of a concave area on the surface mesh. In Figure 10, the dashed dual cycle is identified as being concave but, in fact, does not enclose a concavity. In order to correctly detect these configurations, we implemented the following workaround: for a cycle C with elimination side $s \in \{l, r\}$ we assume that the primal faces corresponding to the $V^*_{C,s}$ form an almost flat region, i.e. we check if all edges in the subgraph induced by $V^*_{C,s}$ are flat. If so, the cycle in question is marked concave.

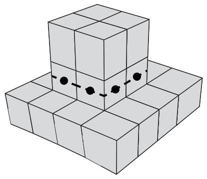

Fig. 10. A dual cycle that is falsely classified as concave because its parallel neighboring primal edges (below the cycle) are all characterized as concave.

Invalid Intrusion Operations

The elimination of a concave dual cycle C with respect to side $s \in \{l, r\}$ is equivalent to adding $|V^*_{C,s}|$ hexahedra to the unknown complex. Accordingly, it is a necessary requirement that each intrusion operation removes exactly $|V^*_{C,s}|$ hexahedra from the hexahedral complex. However, since we still do not have general control over the topology of the reconstructed hexahedral mesh, it might occur in some cases that the number of intruded hexahedra differs from $|V^*_{C,s}|$. That is the case if multiple primal faces corresponding to the $V^*_{C,s}$ are incident to the same hexahedron in the complex as illustrated in Figure 11 (a). As soon as the reconstruction of another dual cycle involves a subset of the newly revealed faces, the algorithm will get stuck since the requirement of surface conformity is violated in these cases. This is still an open problem and can only, up to this state, be partially addressed via the technique described in Section 5.4. Unfortunately, these configurations are not rare in complex geometries and thus this aspect needs further investigation.

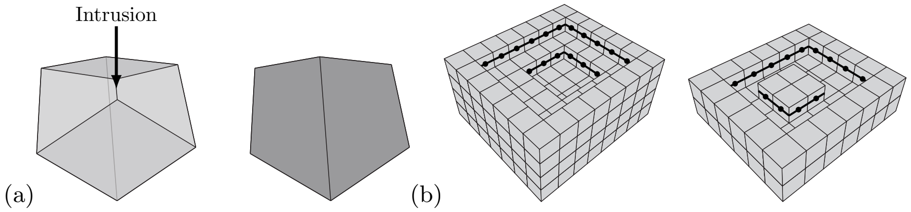

Fig. 11. (a) The surface of a hexahedral complex before (left) and after an intrusion (right). This is a scenario where the number of revealed primal faces after an intrusion is not equal to the number of faces in the original intrusion region. These configurations cause the algorithm to get stuck in most cases. (b) Two examples of input surface meshes in which a cycle that is classified as concave encloses another cycle to its elimination side. Left: the enclosed cycle is classified as concave. Right: the enclosed cycle is classified as convex.

Enclosed Concave Cycles

Since the lack of information about the intermediate hexahedral complex during the elimination steps we still do not have control over the problem described in Section 5.4.

However, it is possible to avoid the occurrence of these configurations in many cases. It is easy to understand that, whenever a concave cycle encloses another cycle to its elimination side, it is highly likely that two faces within the intrusion area will be incident to the same hexahedron. Therefore, it is straightforward to overcome this issue: for a concave dual cycle C, all enclosed dual cycles except flat cycles must be eliminated prior to eliminating C. This has to be done regardless of whether the enclosed cycle is classified as concave or convex. An illustration of two respective cycle arrangements is shown in Figure 11 (b). It is easy to see that, in order to generate a feasible hexahedral mesh in a concave region, the topology of the surface mesh in the concave area and its “backside”, i.e. the surface on the opposite side, have to match.

6 Results and Evaluation

We have run the proposed algorithm on a set of quadrilateral surface meshes which predominantly have been generated using an implementation of the mixed-integer quadrangulation method [7] with manual singularity placement and connectivity constraints [25]. Figure 13 shows some hexahedral meshes generated using our method. The geometric embedding is computed using the Mesquite library [9]. We compare our results to the results produced with the original method (without cutting and remeshing of concave objects) in terms of number of elements and the Scaled Jacobian footprint. The statistics for the results are shown in the table below. The columns from left to right: mesh, number of hexahedra, minimum, average, and maximum Scaled Jacobian (SJ). The numbers on the left (green) are the results from the original method and those on the right (blue) are our results.

Mesh	**# Hex.**	**Min. SJ**	**⊘ SJ**	**Max. SJ**
Drilling H.	12196/10260	-.99/.38	.86/.98	.99/.99
Fandisk	16088/16088	.14/.14	.98/.98	.99/.99
Knotted H.	4713/1529	-.99/.92	.53/.99	.99/.99
RockerArm	13302/3572	-0.99/.15	.51/.76	.99/.99
Sphere	900/900	.04/.04	.81/.81	.99/.99
Fanpart	188/188	.85/.85	.97/.97	.99/.99
Bullet	188/56	.39/.28	.65/.72	.97/.98

It can be seen that, in most cases, our algorithm generates fewer elements as compared to the original approach. Furthermore, in most cases, our minimum and average Scaled Jacobians are not only exclusively positive but are significantly greater than those generated with the original approach. This is due to the fact that, in the original method, many degenerate hexahedra are created near concave regions whereas the proposed approach is able to generate well-shaped elements in these regions. Compared to the results in [19] (using CubeCover), one can see that the minimum and average Scaled Jacobian for the Fandisk model is greater using our method. However, since the resolution of our mesh is considerably higher, this comparison is rather insignificant. Figure 12 shows a comparison of the topology of a mesh generated with the original approach and our method. Due to the cycle elimination dependency rules, our method produces the hex-mesh topology as suggested by the input surface. The original method creates many asymmetric interior singularities.

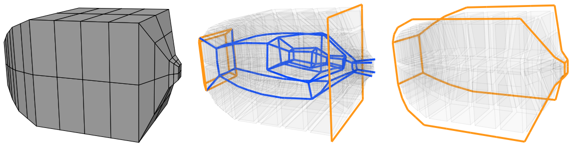

Fig. 12. Left: the original bullet quad mesh. Middle: the result from the original method. Right: the result from our algorithm.

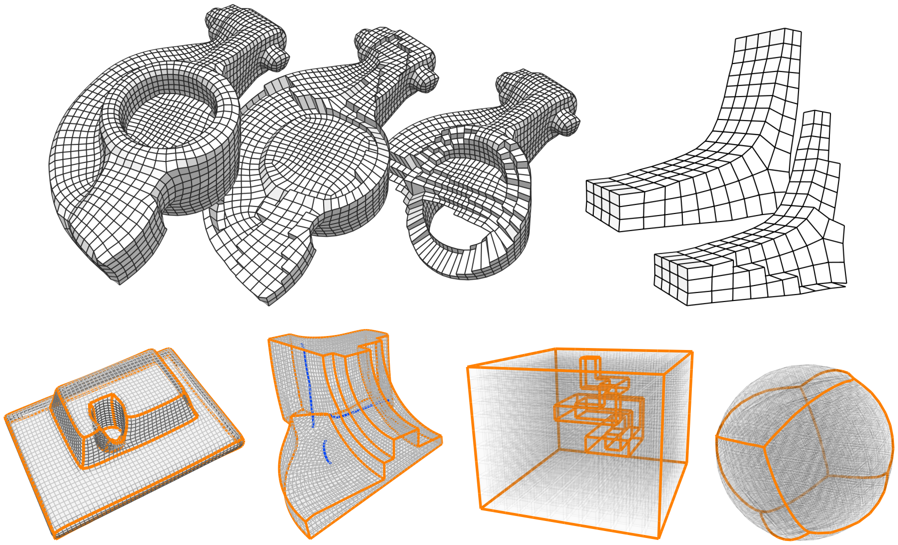

Fig. 13. Top row, left: The RockerArm model and dissections that reveal its interior structure. The mesh was modified such that the handle in the middle is closed with two planar patches. Right: The Fanpart mesh with dissection. Bottom row, from left to right: Drilling Hole, Fandisk, Furch's Knotted Hole, Sphere. Interior singularities are shown as blue lines and orange lines indicate exterior singularities.

7 Conclusions

We describe a set of improvements to the conceptual hex-meshing method initially proposed by M. Müller-Hannemann that not only considerably increase the quality of the generated meshes but also extends the class of meshable surface meshes. The proposed method can handle many concave surface configurations without the need of mesh surgery prior to the actual meshing. Also, we describe means to minimize the number of generated singularities by incorporating a set of simple rules in the dual cycle elimination routine. Our techniques advance the rather theoretical original approach further towards being a practical solution for the problem of finding feasible hex-meshes for surface meshes of genus zero. However, the restriction to input meshes of genus zero that do not have self-intersecting dual cycles still is a serious limitation the realization of which may need a considerable amount of manual intervention. Furthermore, as both the original method and the proposed one are surface conforming, the quality of the resulting meshes crucially depends on the quad mesh layout of the input mesh. The more a quad mesh's layout is aligned with its inherent geometric features the better results will be achieved using the proposed method. Nevertheless, as recent research tends to produce increasingly powerful approaches to quad meshing, the described method is a valuable alternative to parameterization-based hex-meshing methods that, in the current state, also require user interaction and are more error-prone.

Acknowledgments

This project was funded by the European Research Council (ERC Starting Grant "Robust Geometry Processing", Grant Agreement 257474) and the DFG Cluster of Excellence on Ultra High-Speed Mobile Information and Communication (UMIC), German Research Foundation grant DFG EXC 89.

References

1. C. Armstrong, D. Robinson, R. McKeag, T. Li, S. Bridgett, R. Donaghy, C. McGleenan, R. Donaghy, and C. Mcgleenan. Medials for meshing and more, 1995.
2. T. Blacker. The cooper tool. In *5th IMR, Sandia*, 1996.
3. T. Blacker. Meeting the challenge for automated conformal hexahedral meshing. In *9th IMR*, 2000.
4. T. D. Blacker and R. J. Meyers. Seams and wedges in plastering: A 3-d hexahedral mesh generation algorithm. *Engineering w. Comp.*, 9, 1993.
5. D. Bommes, M. Campen, H.-C. Ebke, P. Alliez, and L. Kobbelt. Integer-grid maps for reliable quad meshing. *ACM Trans. Graph.*, 32(4), July 2013.
6. D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin. State of the art in quad meshing. In *Eurographics*, 2012.
7. D. Bommes, H. Zimmer, and L. Kobbelt. Mixed-integer quadrangulation. In *ACM SIGGRAPH 2009*, New York, USA.
8. M. Botsch, S. Steinberg, S. Bischoff, and L. Kobbelt. Openmesh - a generic and efficient polygon mesh data structure, 2002.
9. M. Brewer, L. Diachin, P. Knupp, T. Leurent, and D. Melander. The mesquite mesh quality improvement toolkit. In *12th IMR*, 2003.
10. M. Campen, D. Bommes, and L. Kobbelt. Dual loops meshing: quality quad layouts on manifolds. *ACM Trans. Graph.*, 31(4), 2012.
11. N. T. Folwell and S. A. Mitchell. Reliable whisker weaving via curve contraction. In *7th IMR, Sandia*, 1999.
12. P. J. Frey, H. Borouchaki, and P. L. George. Delaunay tetrahedralization using an advancing-front approach. In *5th IMR*, 1996.
13. M. E. Hohmeyer and W. Christopher. Fully-automatic object-based generation of hexahedral meshes. In *4th IMR*, 1995.

14. J. Huang, Y. Tong, H. Wei, and H. Bao. Boundary aligned smooth 3d cross-frame field. In *SIGGRAPH Asia 2011*. ACM, 2011.
15. F. Kälberer, M. Nieser, and K. Polthier. Quadcover - surface parameterization using branched coverings. In *CG Forum*, volume 26, 2007.
16. P. Knupp. Next-generation sweep tool: A method for generating all-hex meshes on two-and-one-half dimensional geomtries. In *7th IMR*, 1998.
17. F. Ledoux and J.-C. Weill. An extension of the reliable whisker weaving algorithm. In *16th IMR*, pages 215–232. Springer, 2008.
18. M. Li and R. Tong. All-hexahedral mesh generation via inside-out advancing front based on harmonic fields. *The Visual Computer*, 2012.
19. Y. Li, Y. Liu, W. Xu, W. Wang, and B. Guo. All-hex meshing using singularity-restricted field. *ACM Trans. Graph.*, 31(6), 2012.
20. S. Liu and R. Gadh. Automatic hexahedral mesh generation by recursive convex and swept volume decomposition. In *6th IMR, Sandia*, 1997.
21. K. Miyoshi and T. Blacker. Hexahedral mesh generation using multi-axis cooper algorithm, 2000.
22. M. Müller-Hannemann. High quality quadrilateral surface meshing without template restrictions: A new approach based on network flow techniques, 1998.
23. M. Müller-Hannemann. Shelling hexahedral complexes for mesh generation. *Journal of Graph Algorithms and Applications*, 5, 2001.
24. P. Murdoch, S. Benzley, T. Blacker, and S. A. Mitchell. The spatial twist continuum: A connectivity based method for representing all-hexahedral finite element meshes. *Finite Elements in Analysis and Design*, 28, 1997.
25. A. Myles, N. Pietroni, D. Kovacs, and D. Zorin. Feature-aligned t-meshes. *ACM Trans. Graph.*, 29, 2010.
26. M. Nieser, U. Reitebuch, and K. Polthier. Cubecover–parameterization of 3d volumes. In *CG Forum*, volume 30, 2011.
27. A. M. Saifullah and A. Üngör. A simple algorithm for triconnectivity of a multigraph. In *Proc. of the 15th Australasian Symp. on Computing*, CATS '09. Australian Computer Society, Inc., 2009.
28. A. Sheffer, M. Etzion, and M. Bercovier. Hexahedral mesh generation using the embedded voronoi graph. In *7th IMR*, 1999.
29. J. F. Shepherd and C. R. Johnson. Hexahedral mesh generation constraints. *Engineering with Computers*, 24, 2008.
30. J. R. Shewchuk. Tetrahedral mesh generation by delaunay refinement. In *Symp. on Computational Geometry*, SCG '98. ACM.
31. M. Staten, S. A. Canann, and S. Owen. Bmsweep: Locating interior nodes during sweeping, 1998.
32. D. J. Struik. *Lectures on Classical Differential Geometry*. Addison-Wesley Publishing Co., Massachusetts, 1961.
33. T. J. Tautges, T. Blacker, and S. A. Mitchell. The whisker weaving algorithm: A connectivity-based method for constructing all-hexahedral finite element meshes, 1995.
34. P. M. Tuchinsky and B. W. Clark. The hextet hex-dominant automesher: An interim progress report. In *6th IMR*, 1997.
35. D. White, L. Mingwu, S. E. Benzley, and G. D. Sjaardema. Automated hexahedral mesh generation by virtual decomposition. In *4th IMR, Sandia*, 1995.
36. M. Zhang, J. Huang, X. Liu, and H. Bao. A wave-based anisotropic quadrangulation method. In *ACM SIGGRAPH 2010*.