

Hexahedral Mesh Structure Visualization and Evaluation

Kaoji Xu and Guoning Chen *Member, IEEE*

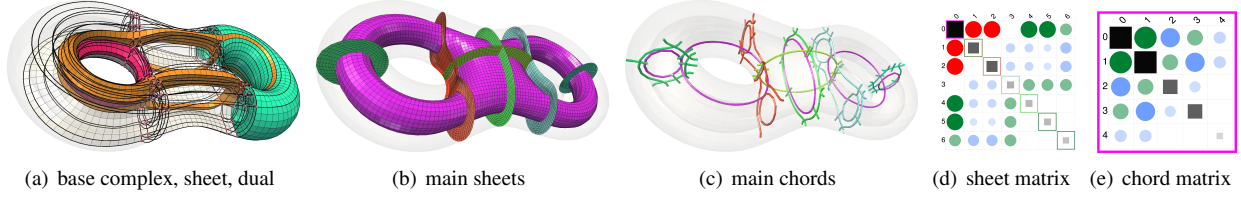


Fig. 1. We extract all base complex sheets (green and brown in (a) are two sheets) from the base complex (red dots and thick black lines in (a)) of a hexahedral (hex-) mesh, then select a set of sheets (b) that can most efficiently represent the base complex. Similarly, for each sheet, a dual surface of it could be further decomposed to a set of chords (c). We calculate the complexity of the structure using an enhanced adjacency matrix, where each diagonal entry (gray square) shows the complexity of the corresponding sheet, and the entry (i, j) shows the relations between sheet i and sheet j . From this matrix representation, the complexity of the base complex can be computed. The above procedure effectively helps us evaluate the structure configuration of a hex-mesh.

Abstract— Understanding hexahedral (hex-) mesh structures is important for a number of hex-mesh generation and optimization tasks. However, due to various configurations of the singularities in a valid pure hex-mesh, the structure (or base complex) of the mesh can be arbitrarily complex. In this work, we present a first and effective method to help meshing practitioners understand the possible configurations in a valid 3D base complex for the characterization of their complexity. In particular, we propose a strategy to decompose the complex hex-mesh structure into multi-level sub-structures so that they can be studied separately, from which we identify a small set of the sub-structures that can most efficiently represent the whole mesh structure. Furthermore, from this set of sub-structures, we attempt to define the first metric for the quantification of the complexity of hex-mesh structure. To aid the exploration of the extracted multi-level structure information, we devise a visual exploration system coupled with a matrix view to help alleviate the common challenge of 3D data exploration (e.g., clutter and occlusion). We have applied our tool and metric to a large number of hex-meshes generated with different approaches to reveal different characteristics of these methods in terms of the mesh structures they can produce. We also use our metric to assess the existing structure simplification techniques in terms of their effectiveness.

Index Terms—hexahedral mesh, base complex, sheet decomposition, complexity analysis

1 INTRODUCTION

Hexahedral meshes have received increasing attention in the past decades due to their superior numerical properties when compared to tetrahedral meshes, which is important for many critical scientific simulations. Given a valid all (or pure) hex-mesh, a global 3D parameterization is imposed within the volume. The irregular elements (e.g., the mesh edges whose valence is not 4 in the interior or not 2 on the boundary) of the mesh correspond to the discontinuity in this 3D parameterization, which form the *singularities*. From these singularities, a global volume partitioning strategy can be induced that organizes the individual hexahedral elements (i.e., hexahedra) into larger hexahedral components (or blocks). This partitioning corresponds to the structure of the given hex-mesh, which is referred to as the *base complex*. Fig. 2(a) shows such a 3D structure of a valid pure hex-mesh using transparent surfaces and wireframe.

The complexity of this 3D structure will affect a number of subsequent tasks performed on the corresponding hex-mesh, such as the Isogeometric Analysis (IGA) [6, 18] and trivariate spline fitting for a smooth volume representation [13]. Specifically, in the spline fitting, a C^2 spline basis can be fit to each hex-block for the Finite Element Analysis (FEA), while only C^0 continuity can be guaranteed across the boundaries of neighboring hex-blocks. The fewer the blocks the fewer the boundaries, and thus a higher smoothness throughout the entire volume can be achieved for the above applications. However, even

the same set of singularities are given, different mesh structures (or base complexes) can be resulted due to the mis-alignment issue [11], resulting in complicated configuration. Fig. 2(b) shows a tangling configuration of the hex-elements within a fertility hex-mesh. This configuration makes the tasks, e.g. spline fitting, challenging.

It is desirable to develop effective strategies to optimize hex-mesh structure (i.e., base complex) for various applications. To achieve so, the possible configurations in the base complex need to be identified and understood. In addition, an effective complexity measure is required to accurately quantify the complexity of a base complex so that it can be used to guide the optimization of base complexes. However, existing techniques simply rely on transparent surfaces and wireframe to visualize the base complexes. While this visualization is good at revealing the overall complexity of the structure, it does not offer an effective means to decipher the cause of the complexity. This motivates the present work.

In this paper, we offer the geometers and practitioners alike an effective visual exploration system to understand the configurations and complexity of the base complexes of their hex-meshes. Our system adopts a two-stage pipeline. First, given a valid pure hex-mesh, the base complex is extracted and decomposed to a series of base complex sheets (or sheets for simplicity) – a semi-global organization (or sub-structure) of the hexahedral blocks with planar configuration, from which a number of sheets are selected that most efficiently represent the original base complex. From these obtained sheets, their mutual connectivity relations are classified and properly represented for the subsequent assessment of the base complex complexity. The reason of using sheets instead of individual components to study the complexity of a base complex is that sheets better reveal the global configuration of the embedded 3D parameterization than local components, which is crucial for hex-mesh generation and improvement.

Second, with the above extracted information, the user performs

• Kaoji Xu and Guoning Chen are with the University of Houston. E-mails: cotrikxu@gmail.com, gchen16@uh.edu

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxxx

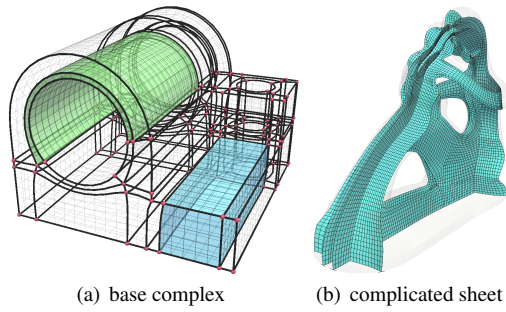


Fig. 2. (a) The red dots and thick black lines indicate the vertices and edges of a base complex, respectively. The green and blue areas highlight two components of the base complex. (b) A complicated sheet in a fertility hex-mesh.

various visual exploration operations to understand the single and/or multiple sheet configurations, and how they influence the complexity of a base complex. In addition, the users can compare the structures of the hex-meshes generated with different approaches for the same objects to identify the characteristics of those hex-meshing approaches.

In summary, this work makes the following contributions:

- (1) We propose to use the base complex sheets to study the base complex configurations and introduce an efficient algorithm to extract a near optimal subset of sheets that can most effectively represent the original base complex.
- (2) We study the connectivity configuration among neighboring sheets and propose to use an enhanced adjacency matrix to represent various configurations, from which a first complexity metric for base complexes is defined.
- (3) We develop a visualization system to aid the exploration of the base complex configurations that provides users multiple levels of structural information.

2 RELATED WORK

Hex-meshing methods Considering the importance of hex-meshes to finite element simulation [29], a large amount of efforts have been dedicated to their generation. These methods range from the early sweeping and paving [35,36], grid-based [9,32,37,40] and octree-based methods [20,26,41,42] to the polycube-based [10,15,16,25] and frame-field-based approaches [17,21,24,28]. Recent surveys [3,33] provide a detailed look at the advances in this direction. In the meantime, some work is dedicated to address a relaxed problem, i.e., hex-dominant meshing [12,34], and has achieved various levels of success.

According to our observation, every meshing method has its unique characteristic. For the polycube-based methods, the singularities of the obtained hex-meshes all reside on the surface; in contrast, for frame-field based methods, singularities could be positioned inside the volume. For the octree-based hex-meshes, there is a padding layer over the surface and most of the singularities are attached to the padding, while the inner layer is equivalent to a voxelized grid (i.e., a very complex polycube). To some extent, the structure of the polycube based hex-meshes that have a padding layer shares some similarity with the one generated by the octree-based method. The question is, given an arbitrary pure hex-mesh, can one easily identify by which method the mesh is generated? This motivates us to develop a technique to help identify the unique patterns within the hex-meshes generated with those popular meshing methods.

Hex-mesh singularity Singularities play an important role in hex-mesh generation, structural optimization, and mesh post-processing. For hexahedral mesh generation, especially for frame-field based methods, the quality of the obtained hex-meshes heavily rely on the placement of singularities. Armstrong et al. pointed out that a main task in quad/hex mesh generation is the placement of mesh singularities to give the desired element orientation and distribution [1]. Nieser et al. [28] showed that there are 24 possible singularity types and applied a coarse manually designed meta-mesh to guide the generation of a 3D global parameterization from the obtained frame field. Huang et al. introduced

a method to automatically construct a boundary-conformal 3D frame field. However, the obtained frame fields may contain unwanted types of singularities. Therefore, only hex-dominant meshes can be resulted. Later, Li et al. [24] restricted the singularities to 10 types that are needed to obtain a valid hexahedral configuration, which can be used to generate a hex-mesh with better structure conformal to the surface. Similarly, Jiang et al. [21] presented a practical framework for adjusting singularity graphs by automatically modifying the rotational transition of frames between charts to resolve the invalid (or non-hexahedral) configuration detected in the internal and boundary singularity graph. However, this repairing can only detect local invalidity and is not sufficient for global deficits of the singularity graphs. Generally, a common consensus is that a good placement of the singularities will result in a good structure of the hex-mesh. For example, by only aligning the singularities the structure becomes simpler and the quality of mesh becomes better [11]. In addition, removing unwanted singularities may help improve the mesh structure and mesh quality [14].

Base complex typically refers to the coarse configuration of mesh elements, which is of importance in mesh generation, mesh optimization, and parameterization. Eck et al. converted a triangular base complex to a quadrangular base complex which helps in extracting a B-spline surface [8]. The work shows that a mesh can be parameterized by a coarse quadrilateral layout. Khodakovsky et al. constructed a triangular base complex for a globally smooth parameterization [22], while Dong et al. obtained a quad base complex by extracting the Morse-Smale complex of a selected Laplacian eigenfunction [7]. Considering its importance, a significant amount of work on mesh base complex has been reported, especially in the field of quadrangulation [2,30,31]. It is worthy to point out that Tarini et al. proposed a method to extract simple quad domains from a quad mesh [38]. Later, the concept of base complex was extended to extract the base complex of a hex-meshes in help with the generation and simplification of hex-meshes [5,11,14].

Sheet representation Borden et al. [4] represented a hexahedral mesh using a set of spacial twist plane (STC), also referred to as *sheet*. Later, Merkley et al. applied sheet insertion to hexahedral mesh to modify the mesh connectivity [27]. Kowalski et al. proposed a theoretical classification for fundamental sheets participating in the geometry capture procedure [23]. It is important to note that the sheets discussed in the above methods are constructed using the hexahedral elements. Recently, Gao et al. combined the concept of base complex and sheets to achieve the efficient alignment of the singularities of a hex-mesh [11], and the simplification of the structure of the hex-mesh [14]. Wang et al. combined frame field and base complex sheet adjustment to improve hex-mesh topological connectivity [39]. As shown in these recent works, base complex sheets are the keys for the hex-mesh structure optimization. However, there still lacks an in-depth understanding in the hex-meshing community on the configuration of the individual sheets, and, more importantly, their mutual connectivity configuration and its impact to the overall mesh structure complexity, which hampers the development of an optimal structure simplification framework. We aim to address this lack in this work.

3 HEX-MESHES AND BASE COMPLEX

We start with the introduction of some important concepts of hex-meshes and base complex based on the work by Gao et al. [11,14].

3.1 Hex-mesh Basis

Hex-mesh In computational solutions of partial differential equations, meshing is a discrete representation of the geometry that is involved in the problem. Essentially, it partitions space into elements (or cells or zones) over which the equations can be approximated. A hex-mesh is a volumetric mesh where each cell, called hexahedron, also called a hex or a brick, is isomorphic to a cube, i.e., having 8 vertices, 12 edges, and bounded by 6 quadrilateral faces. For the same cell amount, the accuracy of solutions in hexahedral meshes is the highest. Mathematically, $G = \{V, E, F, H\}$ denotes a hex-mesh graph with a set of vertices V , edges E , faces F , and hexes H given a volume Ω with closed boundary $\partial\Omega$. Throughout

the paper, we define the *valence* of an edge with respect to the number of hex-elements (or hexes) adjacent to it.

Singularities Based on the concept of valence, an edge $e \in E$ is irregular if e is an interior edge and its valence is not 4 or e is a boundary edge and its valence is not 2. Otherwise, it is regular. Thus, we obtain a set of irregular edges S_E (e.g., black curves consisting of many small edges in Fig. 3(a)). Similarly, singular vertices, S_V , may exist (e.g., the red dots in Fig. 3(a)). A singular vertex could only be located on singular edges [28]. Specifically, for the two vertices of an irregular edge $s_e \in S_E$, if one of them is on the boundary while the other is not, then the one on the boundary is singular; otherwise (i.e., both of them are in the interior or on the boundary), then the one(s) having more than 2 irregular edges incident to it is singular. Next, we stitch the connected irregular edges in order, to form a link. The link is called a *singularity* s_e , which can be either closed or open (Fig. 3). The two end vertices of an open singularity are called singular vertices s_v . A *singularity graph* $G_S = \{S_V, S_E\}$ consists of all singular vertices and all singularities, where $s_v \in S_V, s_e \in S_E$.

Base complex After obtaining the singularities, we could trace out a patch of faces starting from a singularity, along a face incident to the starting singularity, and ending at either itself, another singularity, or the boundary (Fig. 3(c)). We refer to this surface patch as a *separation surface*, which consists of a set of connected quads (in F) in the hex-mesh. For a valence- n singularity, there are n separation surfaces originated from it. The separation surfaces extracted from all singularities form a surface network embedded in the hex-mesh, which partitions the volume into large hexahedral blocks, called hexahedral components. This partitioning is the *base complex* of the hex-mesh, denoted by $B = \{V_B, E_B, F_B, H_B\}$, where H_B are the hex-components obtained above. V_B, E_B, F_B are the corners, edges, and face patches of the individual hexahedral components. Further, we denote a hex-component in H_B as $G_b = \{V_g, E_g, F_g, H_g\}$, where V_g, E_g, F_g, H_g are the sets of vertices (located at the corners of G_b), hex-edges (forming the edges of G_b), quads (forming the face patches), and hexes (filling the volume) in the hex-mesh, respectively. Note that $|V_g| = 8$ if G_b is isomorphic to a cube, while $|V_g| = 0$ if G_b is a ring (Fig. 4).

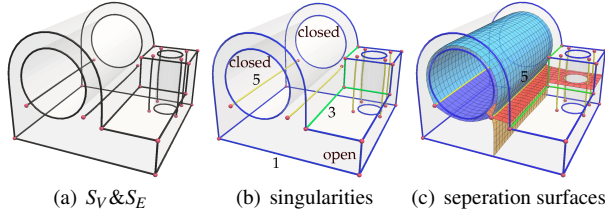


Fig. 3. (a) singular vertices S_V (red dots) and singular edges S_E (black lines) in the joint hex-mesh. (b) Valence of singularities: blue 1, green 3, yellow 5. (c) Five separation surfaces incident to a yellow singularity.

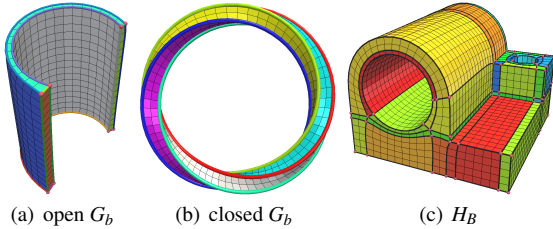


Fig. 4. (a) $|V_g| = 8$. (b) $|V_g| = 0$. (c) The base complex of the joint hex-mesh, where different colors denote different components. Red dots and thick lines connected indicate the corners (or vertices) and edges of the base complex, respectively.

Base complex sheet We denote two base complex edges e_i and e_j in a quad patch of F_B as parallel ($e_i \parallel e_j$) if they do not share a vertex. The *parallel edge set* of an edge e of a base complex is the set of edges that are parallel to e or to any other edge in the set. A base complex sheet is a collection of components that have one or more edges in the same parallel edge set (Fig. 5(a)). For simplicity, we refer to the base complex sheet as *sheet* for the rest of the paper. A sheet consists of

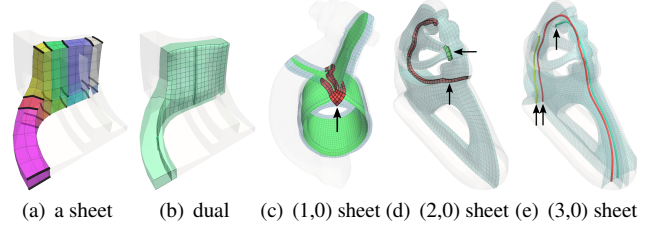


Fig. 5. (a) a sheet is a set of base complex components (in different colors) circumscribed by the parallel base complex edges (black lines). (b) The dual-surface of the sheet in (a) is a quad mesh. The lines perpendicular to the quad mesh are the parallel base complex edges defining this sheet. A number of self-intersecting sheets with one 2-intersection (c), two 2-intersections (d), and three 2-intersections (e). Intersections are highlighted by the arrows.

two side surfaces (perpendicular to the parallel edges) and an interior volume [11].

Given any open base complex component G_b , there are three sets of parallel edges, corresponding to different parameterization directions. Ideally, this means there are three different sheets going through G_b . However, due to the potential tangling configuration, the same sheet may go through the same component more than once (i.e., self-intersecting) when traversing the components of the sheet in order, creating complicated configuration. In this work, we study the configurations of individual sheets based on this returning (or self-intersecting) configuration, which has not been discussed before. In particular, if a sheet goes through all its components once and exactly once, the sheet is called a *simple* sheet. Otherwise, it is called a *non-simple* sheet. For a non-simple sheet, the complexity increases with the increasing number of self-intersections. If a component of a sheet that is visited by the same sheet twice when traversing through the components of the sheet, it creates a 2-intersection. If a component of a sheet is visited by the same sheet three times, it yields a 3-intersection. To characterize different self-intersection configurations of a sheet, we use a 2-tuple (I_2, I_3) ($I_2, I_3 \in \mathbb{Z}^+$), where I_2 and I_3 represent the numbers of the 2- or 3-intersections, respectively. Fig. 5(c-d) show a number of non-simple sheets and their corresponding 2-tuples. Note that we have not observed any sheets with 3-intersection in our testing dataset. Ideally, a simple sheet is preferred, especially for the trivariate spline fitting problem, because the components with multiple visits will cause ambiguity during parameterization. Most of the time, the complex configuration of a sheet is induced by the location, orientation, and valence of the singularities involved, which we will study later in this work (Section 6). For a closed base complex component, there is one set of parallel edges, and self-intersections cannot occur, otherwise additional components will be created at intersections.

Note that if a sheet self-intersects, at the components where self-intersections occur, the parallel edges are not actually parallel to each other since they may share vertices in the same component. Therefore, the above definition needs to be relaxed to take into account this situation. In particular, in the search of all parallel edges, if we return to the components that have been visited before, all the previously visited edges of those components will not participate with the parallel edges checking. This additional care has not been mentioned previously.

Dual Since a sheet constructed on a set of parallel edges has a planer configuration, we compute a mid-surface within the sheet that is everywhere perpendicular to these parallel edges (Fig. 5(b)). This mid-surface, a pure quad mesh, is the dual-surface of the base complex sheet (note that a base complex sheet represents a sub-volume) [19]. This dual-construction strategy enables us to decompose the 3D base complex structure into different sub-structure representations with different dimensionality. In particular, a 2D base complex can be extracted for a dual-surface quad mesh, which can be further decomposed into individual chords. A chord is a chain of quads that can be defined by a set of parallel edges in the obtained 2D base complex. The dual of a chord is the 1D skeletal representation that is perpendicular to the set of parallel edges that define the chord.

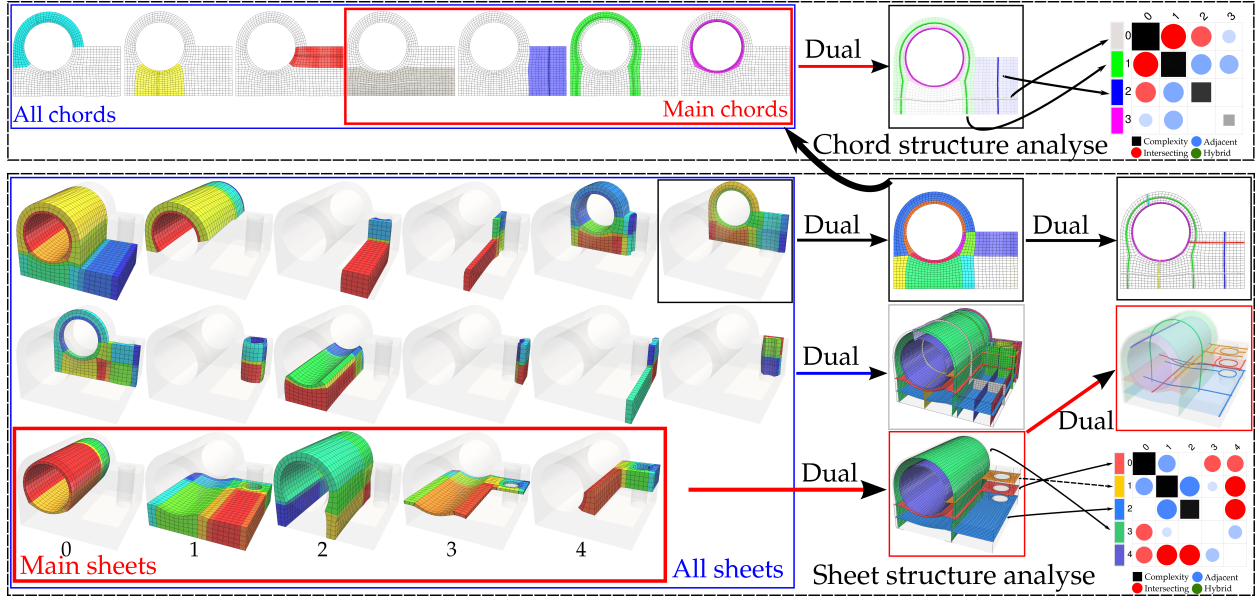


Fig. 6. A multi-level view of a hex-mesh. We extract a list of sheets from its base complex (Fig. 4(c)), from which a subset of sheets are identified that cover the space of the base complex. Each sheet could be collapsed into a quad mesh, i.e., its dual-surface. From this subset of sheets, we construct an adjacency matrix to represent their sheet-to-sheet relations (lower right inset). Next, we construct the 2D base complex for the quad mesh, from which a series of quad chords are extracted (top row). Similarly, a subset of chords are used to represent the 2D base complex. These chords can be further collapsed into some 1D skeletons, which are the dual of these chords. Based on the obtained chords, we can construct an adjacency matrix (upper right inset) to characterize their chord-to-chord relations. Using this top-down decomposition strategy, we can easily identify the most important or complex sheets/chords by sorting the rows/columns of the adjacency matrices based on the relations of the individual sheets/chords with other sheets/chords and/or their respective sheet/chord complexity. For instance, chords 0 and 2 in the upper right inset, and sheets 2 and 3 in the bottom right inset, are more important because they have more connectivity with other sheets/chords.

4 OUR METHOD

According to the concepts of hex-meshes introduced in Section 3, we see that the base complex B and the boundary surface $\partial\Omega$ can already sufficiently describe the input hex-mesh G . More importantly, a base complex is comprised of a set of non-overlapping hexahedral (or base complex) components, which can be organized as series of overlapping sheets. From each sheet, the dual-surface can be extracted, which can be further collapsed into its respective dual skeleton. This multi-level representation (Fig. 6) provides us a means to study the complex structure of 3D base complexes.

4.1 Our Pipeline

Our method consists of the following key steps. (1) We construct the base complex of the given pure hex-mesh. The extraction of singularities is performed as a pre-processing before base complex construction. (2) We extract all base complex sheets based on the above definitions. (3) From all the extracted sheets, we identify a subset of connected sheets that can most effectively represent the base complex of the hex-mesh. (4) We compute the dual-surfaces for the subset of sheets obtained in (3) by collapsing the volume of each sheet to its mid-surface. For each obtained dual-surface, we perform the similar decomposition using steps (1)–(3). (5) With the identified subset of sheets, we identify their sheet-to-sheet connectivity relations and construct an adjacency matrix. With this adjacency matrix, we will compute the complexity of the base complex and perform visual exploration for the understanding of the configurations of the obtained base complex. Fig. 6 illustrates steps (3)–(5) of our processing pipeline.

Next, we provide more details for the individual steps. The extraction of the individual base complex sheets is achieved by finding groups of parallel edges according to the definition. To compute the dual-surface for each sheet, we connect the midpoints of the four parallel edges in each component of the sheet sequentially to form a valid quad mesh. Similarly, to compute the dual 1D skeleton of a chord, we connect the midpoints of the adjacent parallel edges of the chord in order. In the following, we focus on steps (1), (3), and (5).

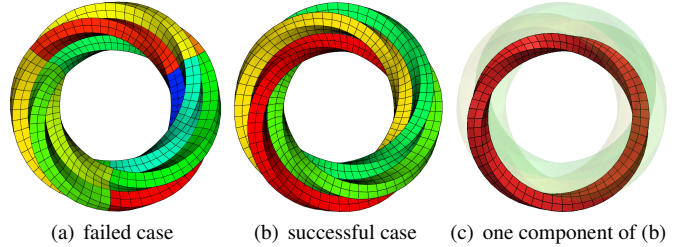


Fig. 7. (a) An incorrect base complex extracted using the previous work [11]. (b) The correct base complex using our new algorithm and one component is highlighted in (c).

4.2 Base Complex Extraction

Based on the above definition, base complex is a partitioning of the mesh based on the obtained separation surfaces originated from the individual singularities. Despite its importance in understanding and improving the hex-mesh quality, there exists little description on how to robustly construct the base complex of a pure hex-mesh in the literature. Gao et al. [11] is the only work presenting a base complex construction algorithm. However, their method requires selecting starting points on closed singularities to finish the tracing of base complex edges, which may create artificial components (Fig. 7). In this work, we revisit the base complex construction for pure hex-meshes and propose a robust algorithm. In particular, after extracting the singularity graph G_s and computing the valence of each singularity, our base complex construction proceeds as follows (Fig. 8).

We extract the separation surfaces B_S by following the individual faces attaching to each singularity (e.g., traversing in the hex-mesh graph G with a consistent direction) until they reach another singularity, return to the starting singularity, or hit the boundary surface. We then extract B_E and B_V from the intersections between surfaces in B_S . Next, we extract H_B . Specifically, for each $h \in H$, we trace out from all its six faces in G until meeting one of the patches in B_S . This will find exactly 6 patches in B_S that form a base complex component $G_b \in H_B$. The set of all G_b is H_B (Fig. 8(c)). Similarly, we can extract F_B . In particular,

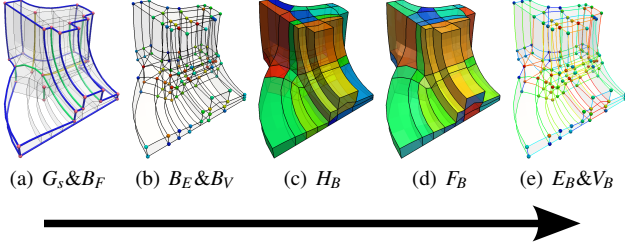


Fig. 8. Illustration of the base complex construction process. (a) Extract singularities G_s and separation surfaces B_F . (b) Extract a set of edges B_E and a set of vertices B_V from the intersection of separation surfaces. (c) Extract a set of components H_B . (d) Extract a set of face patches F_B . (e) Extract a set of links of edges E_B and a set of corners V_B . In (c), (d), and (e), different hexahedral components, face patches, links, and corners are in different colors.

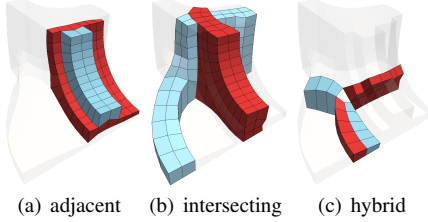


Fig. 9. Three basic types of connectivity relations between two sheets: (a) adjacent, (b) intersecting, and (c) hybrid.

for each quad $f \in B_S$ (and also $f \in F$), trace all its four edges until meeting one of the edges in B_E . This will find 4 edges in B_E that form a base complex patch F_b . The set of all F_b is F_B (Fig. 8(d)). For each $e \in B_E$, trace from both of its two vertices until meeting one of the nodes in B_V , and we will get a E_b . The set of all E_b is E_B (Fig. 8(e)). Extract V_B . In fact, $V_B = B_V$ (Fig. 8(e)).

4.3 Main Sheets Extraction

As described earlier, the extracted sheets overlap each other. In practice, only a small subset of sheets are needed to sufficiently represent the original base complex with minimum overlap. The overlap is defined as how many components the two sheets share. We refer to a subset of sheets that can cover the space of the base complex the *main sheets*. In particular, we aim to identify a subset of sheets that satisfy (1) having as few sheets as possible; (2) having as little overlap as possible; and (3) forming as complex sheet-to-sheet structure as possible. In other words, we aim to identify an optimal subset of sheets that can most effectively describe the complexity of the base complex.

Sheet-to-sheet connectivity relations Before explaining our main sheet extraction algorithm, we first define the relation between two sheets. According to our observation, there are three basic connectivity relations between two neighboring sheets: adjacent (or tangent), intersecting, or hybrid (i.e., both tangent and intersecting to each other), respectively. Fig. 9 provides a few examples of these possible relations. Ideally, the tangent relation is preferred as it easily achieves the layered configuration, while the hybrid configuration is the least preferred sheet-to-sheet configuration. In addition, sheets that are tangent to each other may have different lengths, and intersecting sheets may have more than one intersection. All this information should be considered when quantifying the complexity of the connectivity configuration between sheets. To capture this information, we use a 2-tuple, (p, I) to characterize the connectivity configuration of any two sheets, where $p \in \mathbb{R}$ represents the percentage of adjacency between two sheets, which is the ratio $\frac{|\partial S_1 \cap \partial S_2|}{|\partial S_1| + |\partial S_2|}$ ($|\partial S_1 \cap \partial S_2|$ is the portion of side surface that sheets S_1 and S_2 share). $I \in \mathbb{Z}^+$ counts the number of intersections between two sheets. Although one can choose to use the number of components the two sheets share as I , in practice we find that using the number of intersections as I is sufficient.

Main sheets identification algorithm Finding the optimal set of main sheets satisfying the above goals while covering the space of base

complex is a set cover problem (SCP), which is NP-complete. We propose a practical approximate algorithm based on some greedy depth first search (DFS) and breath first search (BFS) on a sheet connectivity graph $\mathbf{CG}$, which enables us to obtain a near optimal solution. The nodes of $\mathbf{CG}$ are the individual sheets, and the edges are inserted between two sheets if they have one of the above three connectivity relations. Our algorithm consists of the following steps: (a) Use a DFS and a BFS with backtracking starting from **every node** of $\mathbf{CG}$ to find a number of candidate sets of sheets that can cover the base complex; (b) Remove the redundant sheets in each candidate set and remove duplicated candidate sets. (c) Find the set that can best satisfy the three goals. We now describe each step in detail below.

(a) In the DFS, for the current sheet S_i in the current set U_{S_i} , we insert one of its neighboring sheets (based on $\mathbf{CG}$) that is not in U_{S_i} and with minimum number of intersecting base complex components (i.e., minimum overlap – goal (2)) with S_i to U_{S_i} . If there are multiple candidate sheets with minimum overlap, we select the one having the largest number of base complex components because it allows us to fill the base complex quicker – goal (1). If at this point we still have multiple choices, we randomly select one. We then set the new sheet as the current sheet and repeat the above process until the space of base complex is covered (i.e., all its components are visited at least once by one of the sheets in U_{S_i}). In order to backtrack the previous visited sheet and its neighbors if the current search path cannot lead to a full coverage of the base complex, we use a stack to store this path.

The above DFS is a local greedy strategy, which may not find the optimal subset. To ensure we search more paths and enhance the chance of capturing the optimal set, we also perform a BFS search from every node of $\mathbf{CG}$. The difference of this BFS from the above DFS search is that DFS selects one next sheet, while BFS selects multiple next sheets which have the same minimum overlap with the current sheet. In either search, backtracking is needed to obtain a subset covering the entire base complex.

(b) In step (a), redundancy may be introduced. In particular, a sheet whose components have all been visited at least once by other sheets in the same set is removable. For instance, assume we have a sequence of main sheets $\{S_a, S_b, S_c\}$, where $S_a = \{1, 3\}$, $S_b = \{2, 3\}$, and $S_c = \{2, 4\}$ (1–4 are the component IDs). S_b is removable given S_a and S_c . To eliminate this redundancy, we sort all the sheets in a subset based on the numbers of their components, and remove those removable ones with small numbers of components. This will enable us to keep both the number of sheets and overlap among sheets small (i.e., goals (1) and (2)). Also, the above DFS and BFS searches may produce identical sets of candidate main sheets, which should be removed.

(c) We compute the complexity of each candidate subset using a measurement, $\|M\|_F$, defined in Section 4.4, Eq. 2. We then compute the average per sheet complexity of this set. We rank all the candidate subsets based on their average complexity, and select the top-ranked set as the main sheets of our algorithm.

The pseudo-code of our implementation to the above solution is provided in the supplemental document. The complexity of our algorithm is $O(n^3)$, as the complexity of both DFS and BFS is $O(n^2)$ and we perform the search starting from every node in $\mathbf{CG}$.

Correctness of the main sheets extraction It is easy to see that step (a) of the above algorithm will guarantee the complete coverage of the space of base complex. When we search the next sheet we always use the minimum overlap, which satisfies the second goal. Step (b) ensures that goals (1) and (2) are satisfied. We use the maximum average per sheet complexity to select a subset in step (c), which indirectly enables us to satisfy both the first and third goals. This is because a higher average complexity will favor a subset with few number of sheets but relatively high overall complexity. Fig. 10 shows a number of possible subsets for a fandisk hex-mesh sorted according to the numbers of sheets in the respective subsets. Our algorithm selects Fig. 10(b) as the main sheets, which is indeed the optimal set satisfying all three goals. We also provide some detailed verification for a number of meshes in the supplemental document to justify the correctness of our algorithm.

The above algorithm can be similarly applied to the identification of the subset of main chords for a dual-surface quad mesh corresponding

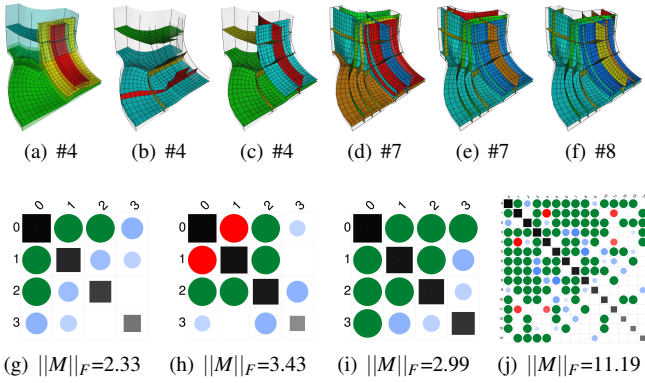


Fig. 10. Given the fan disk model generated by the frame-field method in [24] (a–f) We obtained different sets of representative sheets, ‘#’ denotes the number of sheets. (a), (b) and (c) has the same number of sheets, (g), (h), and (i) are their relations and complexity $\|M\|_F$, respectively. However we use (b) as the main sheets, see more details about complexity calculation in the Section 4.4. (j) shows the relations for all sheets (Fig. 16(b)).

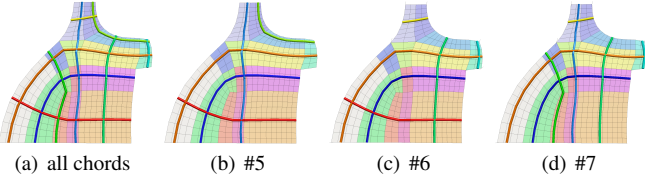


Fig. 11. Given a dual-surface of a sheet in the fan disk model (green quad mesh in Fig. 10(a)), (a) we can extract the base complex (different colors in quads) and a set of all chords (tubes). (b–d) We use the main sheet/chord extraction algorithm to extract different sets of chords, ‘#’ denotes the number of chords.

to a base complex sheet. Fig. 11 shows a number of possible subsets of chords for a dual-surface quad mesh.

4.4 Structure analysis

Based on the above obtained connectivity graph **CG** and the identified subsets, we introduce an adjacency matrix M to represent and visualize the configurations of the individual sheets and sheet-to-sheet relations. In particular, each row i and column j represents sheet S_i . An entry (i, j) is non-zero if sheets S_i and S_j have one of the above three connectivity relations. This non-zero value is computed based on Eq. (1), which quantifies the complexity of the sheet-to-sheet relation between S_i and S_j . The value of each diagonal entry (i, i) is $c_i = \bar{a}_i + I_2 + (I_3 + 1)^2$, which characterizes the complexity of sheet S_i . I_2 and I_3 are the numbers of 2- and 3-intersections of S_i , respectively. We take the square of I_3 to emphasize the complexity of this configuration. $\bar{a}_i = \sum_{m_{ij} \in N} \frac{m_{ij}}{|N|}$ where N is the subset of non-zero entries of row i .

$$m_{ij} = \begin{cases} p & \text{if } I = 0 \\ I & \text{if } p = 0 \\ (p+1)I & \text{Others} \end{cases} \quad (1)$$

Given the above M , we use its Frobenius norm to define a complexity metric for a subset of sheets as follows.

$$\|M\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |m_{ij}|^2} = \sqrt{\text{trace}(M^T M)} = \sqrt{\sum_{i=1}^n \sigma_i^2(M)} \quad (2)$$

where M^T denotes the transpose of M , and $\sigma_i(M)$ are the singular values of M . To our best knowledge, this is the first quantitative complexity metric for 3D base complex. Note that to identify a near optimal set of main sheets from the candidate subsets (Section 4.3), we use $\|M - \text{diag}(M)\|_F$ for their ranking where $\text{diag}(M)$ is a diagonal matrix whose diagonal entries are the diagonal elements in M . The reason we do so is we found that with this alternative complexity the identified optimal subset has fewer number of main sheets in most cases.

Effectiveness of the extracted main sheets We can utilize the proposed complexity metric to assess the effectiveness of the selected main sheets in representing the complexity of the base complex. Take the fan disk results shown in Fig. 10 as an example, the selected main sheets have a complexity value of 3.425656, while the original base complex with all sheets has a complexity of 11.193765. That said, the four main sheets (out of 15 total sheets) contribute 30.6% of the complexity of the base complex. Visually, it is also apparent that the main sheets better convey the layout configuration of hexes than using all sheets.

Similarly, we construct an adjacency matrix for the obtained subset of chords derived from each dual-surface quad mesh of a sheet, from which we compute a complexity value for the dual-surface (Fig. 11).

5 STRUCTURE COMPLEXITY VISUALIZATION

Previous sections describe how we construct and decompose the base complex of a given pure hex-mesh into multi-level sub-structures to aid its analysis. In this section, we describe how we visualize the obtained base complex and its complexity in an intuitive way. In particular, we visualize both the structure in its original 3D space and its abstract configuration relation in the 2D space.

3D visualization With the extracted multi-level structure information, we provide a number of visualizations to aid our exploration. First, to study the configuration of a single sheet, we visualize either the set of parallel edges that define this sheet or its dual-surface. The user can set various transparency and line widths to emphasize different aspects of the sheet. Second, to study the complexity of the entire base complex, we visualize its main sheets obtained from the above extraction. To alleviate the occlusion, transparent dual-surfaces for the main sheets are shown. To further reduce the occlusion, each dual-surface can be represented by the dual 1D skeletal curves of the extracted main chords. Fig. 12 shows an example of how we visualize this multi-level structure information in 3D. More examples are in Fig. 13.

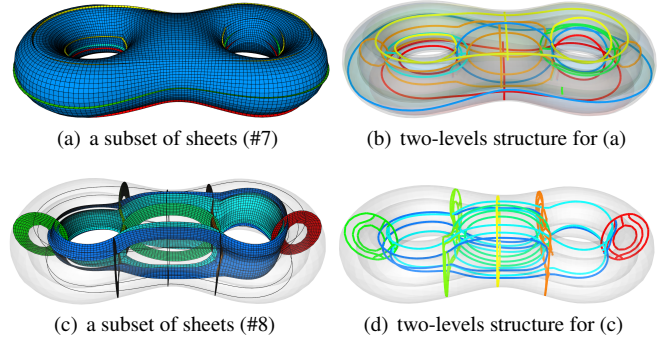


Fig. 12. (a) A padding layer occludes the interior. (b)(d) Use two-levels structure to represent a hex-sheet. (c) another set of main sheets with size of 8. (d) two-levels structure of (c).

2D matrix view To ease the user interaction, we offer a 2D visualization of the obtained enhanced adjacency matrix based on either all the sheets or the extracted main sheets. In particular, we use colored disks to represent sheet-to-sheet connectivity relations with blue disks for adjacent, green for intersecting, and red for hybrid. Also, we scale the disks and determine the saturation of their colors based on the values of the individual entries. To visualize the diagonal entries of the matrix, we use some dark gray squares. The size of each square and its gray color are determined based on the value of the corresponding diagonal entry. Fig. 10 provides a few examples of our matrix visualization.

6 APPLICATIONS

We have applied the aforementioned base complex structure analysis and visualization framework to a number of well-known hex-mesh data sets produced by a few popular hex-meshing techniques, including the volumetric polycube approach [15], ℓ_1 -polycube approach [16], polycut [25], close-form polycube approach [10], and a frame-field-based method [24]. Fig.13 shows the multi-level structure visualization for a number of representative hex-meshes using our approach.

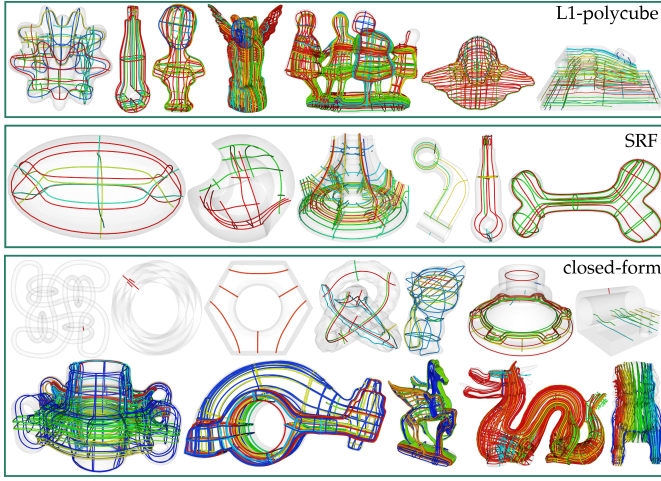


Fig. 13. A gallery of multi-level structures for different meshes.

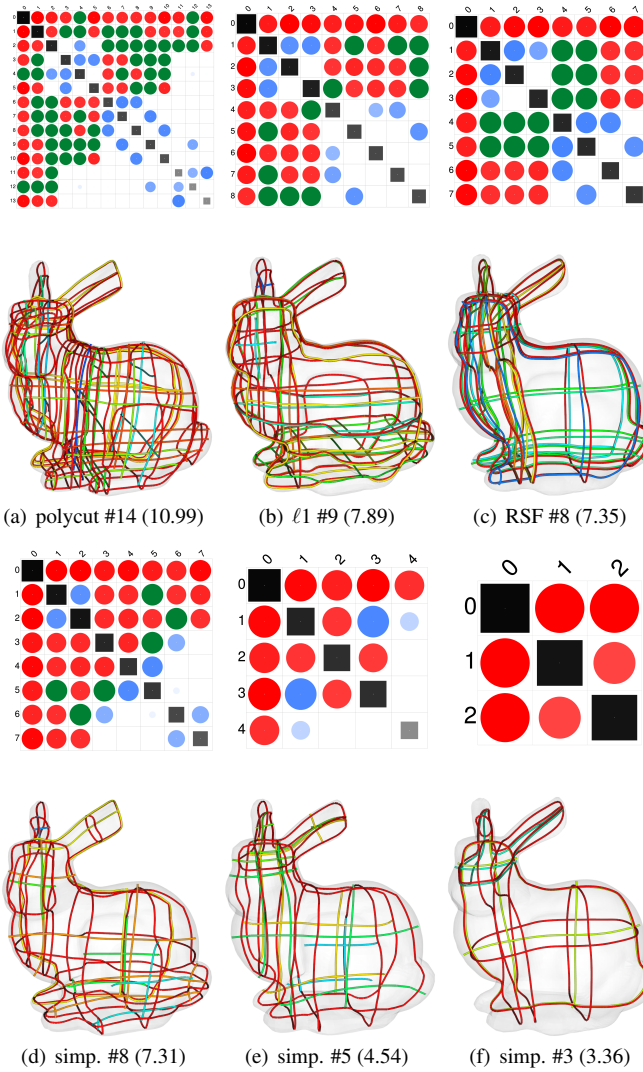


Fig. 14. Comparison of different structures. (a), (b), and (c) are generated by polycut [25], ℓ_1 -polycube [16], and Singularity Restricted Field (SRF) approach [24], respectively. (d), (e), and (f) are their respective simplifications [14]. Different main sheets are shown in different colors. # denotes the number of sheets. The number in each pair of parentheses is the complexity of the corresponding main sheets. The adjacency matrix of each set of main sheets is visualized above each model.

Comparing hex-meshes generated with different approaches

Fig. 14(a-c), Fig. 15, and Fig. 16 provide the visualization of the main sheets and the corresponding adjacency matrices for a number of hex-meshes generated with different approaches. From these results and their comparison, we see that in general, the hex-meshes generated by the frame field based approach have a less complex structure than those generated by polycube approaches (see the bunny and fandisk examples shown in Fig. 14 and Fig. 16). This may be due to the padding layer that is usually applied in a post-processing step for polycube approaches to push the boundary singularities into the interior to improve the boundary conformality and the quality of boundary elements. This padding layer may increase the structure complexity due to the additional misalignment between singularities in the padding layer. However, compared to the polycube-based approach, the base complex of the hex-meshes generated with the frame field based method may have a very complicated configuration in the interior of the volume. This is because the 3D parameterization induced by the frame field may encounter the integer mismatch issue around the holes of the models, easily resulting in helical configuration in the iso-contours of one of the three harmonic fields. As can be seen in the fertility example (Fig. 15), the main sheets of the base complex of the hex-mesh produced by the frame field method (Fig. 15(c)) contain a tangling sheet (also shown in Fig. 2(b)) that has 3 2-intersections. Similarly, there is also one complicated sheet (making 27 2-intersections) in the main sheets of the mesh generated by the closed-form method. This is because the closed-form method also relies on a frame field to compute the 3D parameterization. To summarize, frame field based method may generate hex-meshes with simpler structure for low genus models (like the bunny, fandisk, rod and bone), while polycube-based method works well for high-genus models (like fertility, rocker arm, and joint).

Table 1 provides some statistics of our analysis for the base complexes of a number of representative hex-meshes. Specifically, the effectiveness of the main sheets extracted using our approach for each mesh is computed as the ratio between the complexity of the main sheets (3rd column) and the complexity of all the sheets (5th column). From these examples, we see that the main sheets extracted using our algorithm indeed contribute the most to the complexity of their corresponding base complexes. More statistics can be found in the supplemental document.

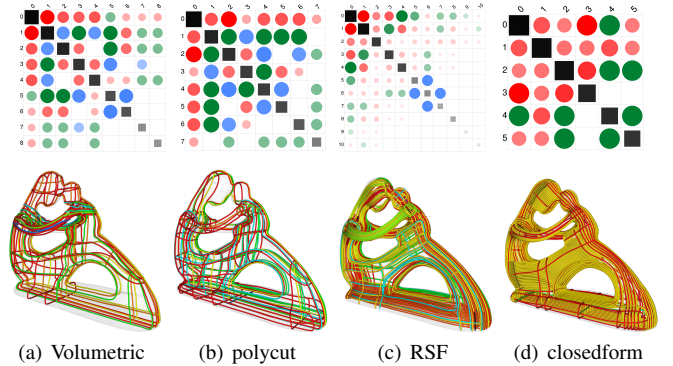


Fig. 15. Comparison of different structures. (a), (b), (c), and (d) are generated by volumetric polycube [15], polycut [25], RSF [24], and closed-form polycube [10], respectively. Different main sheets are shown in different colors. The adjacency matrix of each set of main sheets is visualized above each model.

We also apply our framework to study the structure complexity of the hex-meshes generated by other approaches, including the octree-based method [40] and splitting from an existing tetrahedral (tet-) mesh. See Figs. 16(c) and 16(d) for the fandisk example. Their corresponding main sheets are shown in Figs. 16(g) and 16(h), respectively. Comparing to the preceding parameterization-based approaches, the base complexes of the two hex-meshes generated by the octree-based method and by splitting from a tet-mesh are much more complex. This is expected as both methods produce unstructured hex-meshes. Comparing these two methods, though, despite having more regular hexahedra in

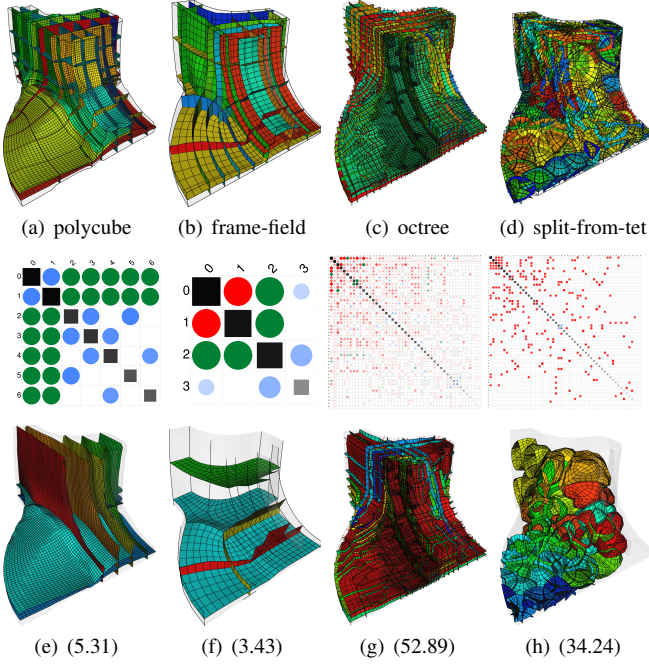


Fig. 16. Base complexes of hex-meshes of the fandisk model generated by different approaches. Top row shows all sheets of each base complex, while the bottom row shows the extracted main sheets and their complexity values. The adjacency matrices of the sets of main sheets are shown in the middle row.

Table 1. Base complex and main sheets complexity

Model	main sheets		all sheets		effectiveness
	#sheets	$\ M\ _F$	#sheets	$\ M\ _F$	
bunny [16]	9	7.63	23	16.55	46.11%
children [16]	37	34.58	105	59.67	57.96%
knot [10]	5	5.28	11	10.06	52.50%
carter [10]	9	25.76	44	43.71	58.92%
bone [24]	8	6.40	14	11.24	56.89%
impeller [24]	14	40.87	51	52.21	78.29%

Table 2. Structure complexity for different simplifications. # denotes the number of main sheets.

Model	Origin	Aligned	Simplified
BU [15]	11.79 (#16)	10.39 (#13)	6.34 (#7)
bumpy_torus [15]	17.11 (#24)	13.25 (#13)	7.19 (#8)
bunny [15]	15.39 (#20)	7.50 (#9)	7.19 (#8)
kiss [15]	24.40 (#23)	17.59 (#16)	10.28 (#9)
BU [25]	12.11 (#15)	6.72 (#8)	5.66 (#6)
fertility [25]	12.24 (#8)	10.07 (#7)	10.11 (#6)
rockerarm [25]	9.09 (#8)	7.93 (#6)	6.62 (#4)

the octree-mesh, the structure of the hex-mesh split from a tet-mesh has a relatively simpler configuration. In particular, only adjacent and intersecting connectivity between sheets is observed in the latter, while all three connectivity relations exist in the sheets of the octree-mesh. This is captured by our complexity metric. Nonetheless, the identified main sheets in the hex-mesh split from the tet-mesh do not follow the shape of the object well, which may not be desired for the subsequent computation tasks. That is probably why octree-mesh is more preferred than the hex-mesh split from a tet-mesh in critical simulation. Another reason is that the Jacobian quality of the individual hex-elements is typically low for a hex-mesh split from a tet-mesh.

Comparing different structure optimization approaches As most existing hex-meshing techniques do not take into account the structural information, the base complexes of the resulting hex-meshes can be arbitrarily complex. To improve their structures, structure simplification and optimization is performed. So far, there are only two existing

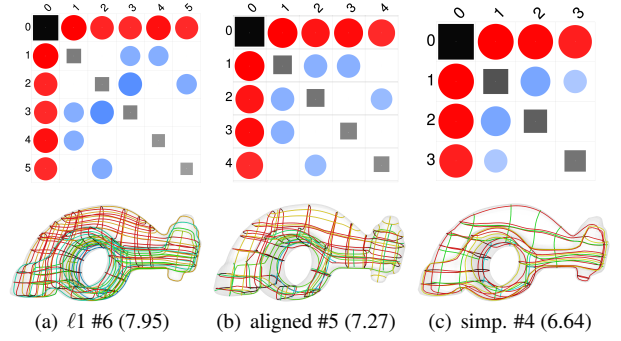


Fig. 17. Comparison of different structure-optimization. (a), (b), and (c) are generated by ℓ_1 -polycube [16], alignment [11] of (a), and simplification [14] of (a), respectively. Different main sheets are shown in different colors. # denotes the number of sheets. The number in each pair of parentheses is the complexity of the corresponding main sheets. The adjacency matrix of each set of main sheets is provided above each model.

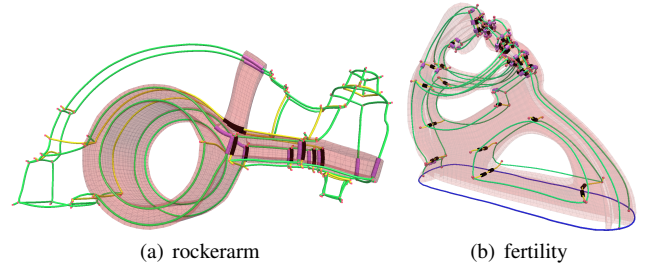


Fig. 18. The self-intersections are typically accompanying with singularities that have orthogonal configurations, especially when these orthogonal singularities belong to the same set of parallel edges that define the sheet. We highlight the parallel base complex edges with valence 3 and 5 in purple and black tubes, respectively.

techniques [11, 14] that can directly simplify the base complex of a pure hex-mesh. We apply our structure analysis and visualization framework to evaluate the effectiveness of these two approaches (Fig. 17 and Fig. 14(d-f)). Table 2 shows the simplification results of the two approaches for a number of pure hex-meshes. In general, the latter simplification that may reduce the number of singularities in the mesh can produce simpler structure, compared to the alignment method. This is expected considering the close relation between the singularities of the mesh and the base complex structure. However, there is at least one case in our experiment that the simplification approach cannot produce a simpler structure than the alignment method, i.e., for the fertility hex-mesh generated by the polycut approach. This is likely caused by the simplification framework that does not explicitly take care of the misalignment during structure reduction.

Relation between singularities and sheets Based on definition, the base complex of a pure hex-mesh is determined by the locations and orientations of singularities of the mesh. As shown in the work by Gao et al. [11], the misalignment between singularities may result in additional sheets or complicated sheet configuration. In addition, neighboring singularities may have different orientations (i.e., different parameterization directions), increasing the chance of misalignment. To have an in-depth understanding of the relations between singularities and the sheet complexity, we concentrate on the sheets that have complicated configuration (e.g., self-intersecting). Our hypothesis is that the self-intersections of sheets are often induced by the non-simple (or non-parallel) configuration of the singularities in the vicinity of the self-intersecting area. To verify our conjecture, we inspect a number of self-intersecting sheets in Fig. 18. As shown in this figure, in both cases the singularities at the locations of intersections are in directions orthogonal to each other. And more importantly, parts of these orthogonal singularities are the parallel edges that define the respective sheets, which explains the change of the parameterization direction at self-intersections.

Fig. 19(a) shows a sheet with an irregular configuration in a kitten hex-mesh. Specifically, the sheet has a Y-shape like configuration while the kitten model does not. By carefully examining the configuration of the singularities (the green and yellow tubes), we can quickly identify some non-symmetric (or more specifically, non-hexahedral) configuration among some neighboring singularities (highlighted by the red circles) that are not aligned with the orientation of the model. This is likely the reason that causes the Y-shape configuration of the sheet. Fig. 19(b) provides a case where the two main sheets define the entire base complex (see the main sheet adjacency matrix in Fig. 19(d)). The two sheets have a hybrid connectivity relation. To understand the cause of their complex relation, we inspect its singularity configuration and locate a singularity (highlighted in purple) that is not aligned with the feature orientation of the model and has some twisting behavior. This leads to the twisting parameterization, thus the tangling configuration of the sheets.

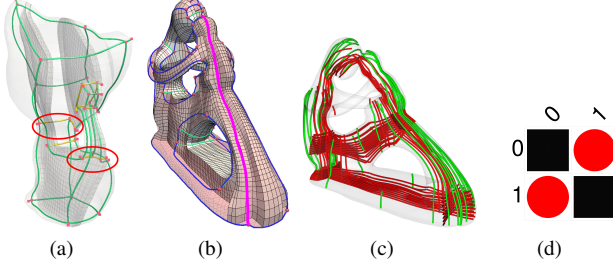


Fig. 19. (a) Non-symmetric configuration of singularities (highlighted in red circles) might produce a misaligned sheet. (b) Non-symmetry singularity (highlighted in purple) might generate self-intersecting sheets (transparent red and non-transparent gray). (c) A multi-level view of (b). (d) Enhanced adjacency matrix of (b).

In the other end of the spectrum, singularities with simple configuration do lead to simple sheet configuration. Fig. 20 shows a number of examples of base complexes, each of which requires only a single main sheet to represent the entire base complex. In the meantime, their singularities have rather simple configuration. For example, the four singularities of the fancy ring hex-mesh (Fig. 20(a)) are closed and parallel to each other (i.e., along the same parameterization direction). Similar singularity configuration can be seen in the KPDloekr hex-mesh (Fig. 20(b)). For the nut hex-mesh (Fig. 20(c)), although there are two groups of singularities that are orthogonal to each other (i.e., following two orthogonal parameterization directions), one of them forms closed singularities and aligned with the boundary, thus, a single main sheet (represented by its dual-surface) can be found to represent the entire base complex.

From the above examples, we can sufficiently determine that an ideal and regular configuration of singularities conformal to the feature orientation of the model is crucial to producing an ideal hex-mesh with the desired structure. This observation will provide guidance for the development of an effective singularity structure creation and optimization for a robust hex-meshing framework.

Another interesting discovery we found during our exploration is that if the valence of a singularity is 1, then the relation between any two sheets attaching to this singularity are intersecting; if the valence is

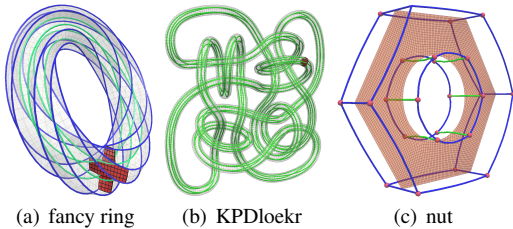


Fig. 20. Meshes that have only one main sheet. (a) and (b) are two base complexes that have only one sheet. Red dots are singular vertices. Blue and green tubes are valence 1 and 3 singularities, respectively. Red surface denotes the dual-surface of the main sheet in (c).

Table 3. Performance for close-form models [10]. G_B denotes the time for base complex extraction and CG is for main sheets extraction.

Model	#V	#H	#H _B	#S	time (s)	
					G_B	CG
fancy_ring	3k	1k	5	1	0.1	0.01
KPDloekr	31k	23k	2	1	25.1	0.01
nut	15k	13k	12	13	0.9	0.02
joint	5k	4k	59	15	0.3	0.02
hollow-eight	11k	9k	249	18	0.7	0.04
kitten	14k	12k	208	16	0.9	0.1
knot	29k	24k	50	11	2.0	0.1
rockerarm	20k	18k	1,202	41	1.5	1.6
carter	75k	65k	1,101	44	5.3	1.7
fertility	22k	16k	2,002	38	2.4	2.1
grayloc	29k	24k	3,183	73	1.9	10.3
chinese-lion	17k	15k	6,235	75	1.4	23.4
pegasus	14k	12k	9,745	120	34.7	69.3
dragon	13k	114k	12,488	134	15.4	112.4

3, it is hybrid. If it is valence 5 or 6, then it would be either adjacent or intersecting. However, we have not thoroughly verified this.

Performance We tested the performance of our method with a list of meshes generated by the closed-form method [10]. Table 3 provides the timing information of this test. Base complex extraction time (column G_B) includes mesh reading, mesh connectivity building, base complex extraction, and base complex connectivity building. Column CG reports the time of main sheet extraction. The timing information was obtained in a workstation with Intel(R) Xeon(R) CPU E5-1620 v2 3.70GHz and 48GB Memory 1866MHz.

7 CONCLUSION

In this work, we propose a multi-level solution to the understanding of the base complex structure of any given pure hex-mesh. Our solution decomposes the base complex extracted from an input valid pure hex-mesh into multi-level sub-structures. In particular, we utilize the concept of base complex sheet as the means to help us define the configurations within a 3D base complex. We propose an effective algorithm to identify a small subset of sheets, called main sheets, to achieve a simplified representation of the base complex. From this subset of sheets, we define a first quantitative metric to measure the complexity of 3D base complexes. To aid the visual exploration of the base complexes, we introduce an adjacency matrix representation to encode the sheet-to-sheet connectivity configuration effectively. We have applied our framework to the understanding of the characteristics of the base complex structures of the hex-meshes generated by different methods, and to the assessment of the efficiency of the existing structure simplification approaches. Finally, we apply our visual exploration system to aid our understanding of the relations between singularities and base complex configuration.

Limitations There are a number of limitations of our work. First, for some models, although our algorithm can return a near optimal set of main sheets, they may not be optimal (see the supplemental document for some examples). Second, it is still hard to show the structure of an octree or tet-split hex-mesh due to the overly complex structure and a large number of extracted main sheets. Furthermore, our current complexity metric does not explicitly take into account the geometric complexity of a sheet. Third, the identified set of main sheets for some meshes could be too abstract to sufficiently represent the configuration of the corresponding base complex (e.g., the one-main-sheet examples shown in Fig. 20, adding additional sheets perpendicular to the main sheet direction may help). Finally, our current framework is for pure hex-meshes, but can be potentially extended for hex-dominant meshes should a proper structure and an equivalent concept to sheets be defined there. We hope to address the above issues in the future.

ACKNOWLEDGMENT

We thank all anonymous reviewers for their constructive comments. We thank Brian Kline for proofreading the paper. This research was supported by NSF IIS 1553329.

REFERENCES

- [1] C. G. Armstrong, H. J. Fogg, C. M. Tierney, and T. T. Robinson. Common themes in multi-block structured quad/hex mesh generation. *Procedia Engineering*, 124:70–82, 2015.
- [2] D. Bommers, T. Lempfer, and L. Kobbelt. Global Structure Optimization of Quadrilateral Meshes. *Comput. Graph. Forum*, 30(2):375–384, 2011.
- [3] D. Bommers, B. Lévy, N. Pietroni, E. Puppo, C. T. Silva, M. Tarini, and D. Zorin. Quad-Mesh Generation and Processing: A Survey. *Comput. Graph. Forum*, 32(6):51–76, 2013.
- [4] M. J. Borden, S. E. Benzley, and J. F. Shepherd. Hexahedral Sheet Extraction. In N. Chrisochoides, ed., *IMR*, pp. 147–152, 2002.
- [5] G. Cherchi, M. Livesu, and R. Scateni. Polycube Simplification for Coarse Layouts of Surfaces and Volumes. *Comput. Graph. Forum*, 35(5):11–20, 2016.
- [6] J. Cottrell, T. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer methods in applied mechanics and engineering*, 196(41–44):4160–4183, 2007.
- [7] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart. Spectral surface quadrangulation. *ACM Trans. Graph.*, 25(3):1057–1066, 2006.
- [8] M. Eck and H. Hoppe. Automatic Reconstruction of B-Spline Surfaces of Arbitrary Topological Type. In J. Fujii, ed., *SIGGRAPH*, pp. 325–334. ACM, 1996.
- [9] J. D. Edgel. An adaptive grid-based all hexahedral meshing algorithm based on 2-refinement. 2010.
- [10] X. Fang, W. Xu, H. Bao, and J. Huang. All-hex meshing using closed-form induced polycube. *ACM Trans. Graph.*, 35(4):124, 2016.
- [11] X. Gao, Z. Deng, and G. Chen. Hexahedral mesh re-parameterization from aligned base-complex. *ACM Trans. Graph.*, 34(4):142, 2015.
- [12] X. Gao, W. Jakob, M. Tarini, and D. Panozzo. Robust hex-dominant mesh generation using field-guided polyhedral agglomeration. *ACM Trans. Graph.*, 36(4):114:1–114:13, 2017.
- [13] X. Gao, T. Martin, S. Deng, E. Cohen, Z. Deng, and G. Chen. Structured Volume Decomposition via Generalized Sweeping. *IEEE Trans. Vis. Comput. Graph.*, 22(7):1899–1911, 2016.
- [14] X. Gao, D. Panozzo, W. Wang, Z. Deng, and G. Chen. Robust structure simplification for hex re-meshing. *ACM Trans. Graph.*, 36(6):185:1–185:13, 2017.
- [15] J. Gregson, A. Sheffer, and E. Zhang. All-Hex Mesh Generation via Volumetric PolyCube Deformation. *Comput. Graph. Forum*, 30(5):1407–1416, 2011.
- [16] J. Huang, T. Jiang, Z. Shi, Y. Tong, H. Bao, and M. Desbrun. 1-Based Construction of Polycube Maps from Complex Shapes. *ACM Trans. Graph.*, 33(3):25:1–25:11, 2014.
- [17] J. Huang, Y. Tong, H. Wei, and H. Bao. Boundary aligned smooth 3D cross-frame field. *ACM Trans. Graph.*, 30(6):143, 2011.
- [18] C. J. B. Y. Hughes T.J. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.
- [19] J. D. II, C. T. Silva, J. Shepherd, and E. Cohen. Quadrilateral mesh simplification. *ACM Trans. Graph.*, 27(5):148, 2008.
- [20] Y. Ito, A. M. Shih, and B. K. Soni. Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates. *International Journal for Numerical Methods in Engineering*, 77(13):1809–1833, 2009.
- [21] T. Jiang, J. Huang, Y. Wang, Y. Tong, and H. Bao. Frame Field Singularity Correction for Automatic Hexahedralization. *IEEE Trans. Vis. Comput. Graph.*, 20(8):1189–1199, 2014.
- [22] A. Khodakovskiy, N. Litke, and P. Schröder. Globally smooth parameterizations with low distortion. *ACM Trans. Graph.*, 22(3):350–357, 2003.
- [23] N. Kowalski, F. Ledoux, M. L. Staten, and S. J. Owen. Fun sheet matching: towards automatic block decomposition for hexahedral meshes. *Eng. Comput. (Lond.)*, 28(3):241–253, 2012.
- [24] Y. Li, Y. Liu, W. Xu, W. Wang, and B. Guo. All-hex meshing using singularity-restricted field. *ACM Trans. Graph.*, 31(6):177, 2012.
- [25] M. Livesu, N. Vining, A. Sheffer, J. Gregson, and R. Scateni. PolyCut: monotone graph-cuts for PolyCube base-complex construction. *ACM Trans. Graph.*, 32(6):171:1–171:12, 2013.
- [26] L. Maréchal. Advances in Octree-Based All-Hexahedral Mesh Generation: Handling Sharp Features. In B. W. Clark, ed., *IMR*, pp. 65–84. Springer, 2009.
- [27] K. G. Merkle, C. Ernst, J. Shepherd, and M. J. Borden. Methods and Applications of Generalized Sheet Insertion for Hexahedral Meshing. In M. L. Brewer and D. L. Marcum, eds., *IMR*, pp. 233–250. Springer, 2007.
- [28] M. Nieser, U. Reitebuch, and K. Polthier. CubeCover- Parameterization of 3D Volumes. *Comput. Graph. Forum*, 30(5):1397–1406, 2011.
- [29] S. J. Owen. A Survey of Unstructured Mesh Generation Technology. In L. A. Freitag, ed., *IMR*, pp. 239–267, 1998.
- [30] F. H. Razafindrazaka and K. Polthier. Optimal base complexes for quadrilateral meshes. *Computer Aided Geometric Design*, 52:63–74, 2017.
- [31] F. H. Razafindrazaka, U. Reitebuch, and K. Polthier. Perfect Matching Quad Layouts for Manifold Meshes. *Comput. Graph. Forum*, 34(5):219–228, 2015.
- [32] R. Schneiders. A grid-based algorithm for the generation of hexahedral element meshes. *Engineering with computers*, 12(3–4):168–177, 1996.
- [33] J. F. Shepherd and C. R. Johnson. Hexahedral mesh generation constraints. *Eng. Comput. (Lond.)*, 24(3):195–213, 2008.
- [34] D. Sokolov, N. Ray, L. Untereiner, and B. Lévy. Hexahedral-Dominant Meshing. *ACM Trans. Graph.*, 35(5):157, 2016.
- [35] M. L. Staten, R. A. Kerr, S. J. Owen, T. D. Blacker, M. Stupazzini, and K. Shimada. Unconstrained plastering hexahedral mesh generation via advancing-front geometry decomposition. *International journal for numerical methods in engineering*, 81(2):135–171, 2010.
- [36] M. L. Staten, S. J. Owen, and T. D. Blacker. Unconstrained Paving & Plastering: A New Idea for All Hexahedral Mesh Generation. In B. W. Hanks, ed., *IMR*, pp. 399–416. Springer, 2005.
- [37] L. Sun and G. Zhao. Adaptive hexahedral mesh generation and quality optimization for solid models with thin features using a grid-based method. *Eng. Comput. (Lond.)*, 32(1):61–84, 2016.
- [38] M. Tarini, E. Puppo, D. Panozzo, N. Pietroni, and P. Cignoni. Simple quad domains for field aligned mesh parametrization. *ACM Trans. Graph.*, 30(6):142, 2011.
- [39] R. Wang, S. Gao, Z. Zheng, and J. Chen. Hex mesh topological improvement based on frame field and sheet adjustment. *Computer-Aided Design*, 2017.
- [40] H. Zhang and G. Zhao. Adaptive hexahedral mesh generation based on local domain curvature and thickness using a modified grid-based method. *Finite Elements in Analysis and Design*, 43(9):691–704, 2007.
- [41] Y. Zhang and C. Bajaj. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. *Computer Methods in Applied Mechanics and Engineering*, 195(9–12):942–960, Feb. 2006.
- [42] Y. Zhang, X. Liang, and G. Xu. A Robust 2-Refinement Algorithm in Octree and Rhombic Dodecahedral Tree Based All-Hexahedral Mesh Generation. In X. Jiao and J.-C. Weill, eds., *IMR*, pp. 155–172. Springer, 2012.