Hexahedral Mesh Re-parameterization from Aligned Base-Complex

Xifeng Gao* Zhigang Deng† Guoning Chen‡
University of Houston

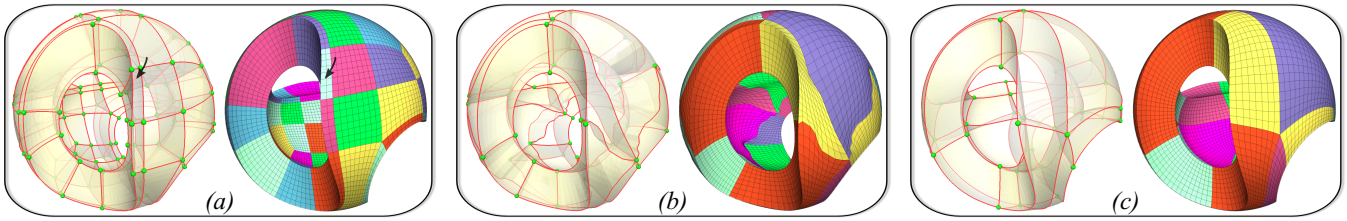


Figure 1: *(a) An input hex-mesh [Li et al. 2012]: The image on the left shows its base-complex that partitions the hexahedral mesh into different large components, illustrated with different colors on the right. Due to the misalignments between singularities, many (typically small) components arise. For instance, a strip of small components near the sharp feature is highlighted. (b) Our alignment algorithm reduces the complexity of the base-complex but leads to a hex-mesh with a large distortion. (c) Both the singularity placement and the element quality of the resulting hex-mesh are improved by our structure-aware optimization algorithm.*

Abstract

Recently, generating a high quality all-hex mesh of a given volume has gained much attention. However, little, if any, effort has been put into the optimization of the hex-mesh structure, which is equally important to the local element quality of a hex-mesh that may influence the performance and accuracy of subsequent computations. In this paper, we present a first and complete pipeline to optimize the global structure of a hex-mesh. Specifically, we first extract the *base-complex* of a hex-mesh and study the misalignments among its singularities by adapting the previously introduced hexahedral sheets to the base-complex. Second, we identify the valid removal *base-complex sheets* from the base-complex that contain misaligned singularities. We then propose an effective algorithm to remove these valid removal sheets in order. Finally, we present a structure-aware optimization strategy to improve the geometric quality of the resulting hex-mesh after fixing the misalignments. Our experimental results demonstrate that our pipeline can significantly reduce the number of components of a variety of hex-meshes generated by state-of-the-art methods, while maintaining high geometric quality.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

Keywords: Hex-mesh, Singularity alignment, Optimization

1 Introduction

Given a two-dimensional closed surface $\mathcal{M}$, producing a high quality volume parameterization f: $\mathcal{M} \rightarrow R^3$, that is aligned with boundary features, is a prerequisite for a number of important applications including finite element analysis (FEA), volume sampling, 3D texture synthesis, and data compression. A hex-mesh $\mathcal{H}$ resulting from a non-degenerate volume parameterization is suitable for partial differencial equation (PDE) solving, tensor-product/trivariate spline fitting [Wang et al. 2012; Li and Qin 2012; Martin et al. 2008], Isogeometric Analysis (IGA) [Hughes et al. 2005; Bazilevs et al. 2006], and multi-grid and adaptive computations [Leonard et al. 2000; Wada et al. 2006].

Although the quality criteria of hex-meshes are application dependent, the hex-meshes that conform to boundary surfaces, preserve boundary features, have regular distribution of parameterization lines, and have low element distortion, are generally preferred [Motooka et al. 2011; Gao et al. 2014]. It is noteworthy that a hex-mesh with the above desired characteristics usually require a certain amount of well-placed singularities that are distributed either on the boundary or in the interior of the volume [Nieser et al. 2011; Huang et al. 2011; Li et al. 2012; Jiang et al. 2014]. Unlike the singularities in frame fields, the singularities of a semi-structured mesh consist of irregular mesh vertices and edges, i.e., those vertices whose valences are not 4 on a 2D quad-mesh [Bommes et al. 2011; Tarini et al. 2011], or edges whose valences are not 4 in the interior of a hex-mesh. These irregular vertices and edges correspond to those places where the parameterization is discontinuous. Intuitively, the singularities as well as the separation structures starting from them partition the domain into small regions. Figure 1(a) shows such a partition example for a Sculpture hex-mesh. Its wireframe and associated transparent surfaces visualize the partitioning structure, which is referred to as the *base-complex* in this paper. Each sub-volume of this partition, referred to as a *component* (i.e., the individual colored regions shown in the right of Figure 1(a)), can be mapped to a cube.

With the above partition, a C^2 spline basis can be fit to each component for FEA, while only C^0 continuity can be guaranteed across the boundaries of different components. For applications, such as IGA, that seek a high level of smoothness throughout the entire volumetric domain, fewer components are desired, as it would lead to more accurate and faster simulations [Hughes et al. 2005]. However, given the same set of singularities, without careful control, it can often result in a partition with a large number of components (Figure 1(a)). This issue has been discussed in quadrangulation applications [Myles et al. 2010; Bommes et al. 2011; Tarini et al. 2011], and is attributed to the *misalignment* of singularities. Li

*e-mail:gxf.xisha@gmail.com
†e-mail:zdeng4@uh.edu
‡e-mail:gchen16@uh.edu

et al. [2012] also pointed out a similar misalignment issue in hex-meshes. To date, an effective and automatic pipeline has not yet been proposed to reduce the number of components by fixing the hex-mesh misalignment issue.

In this paper, we propose a first solution to simplify the base-complex of a hex-mesh by procedurally removing its misalignments. Our technique is based on the key insight that the base-complex (Section 3, Figure 1(a)) is essentially the coarsest hex-mesh that has the same structure as the input hex-mesh. Therefore, its simplification can be achieved by adapting the previously introduced hexahedral sheet removal technique [Borden et al. 2002] to the base-complex. However, removing hexahedral sheets from the base-complex may alter its singularity structure, which is undesired in this work. In addition, removing hexahedral sheets by simply collapsing them into their dual planes as the previous method does may lead to severe geometric artifacts due to the coarse resolution of the base-complex (Figure 8(a)). To address these new challenges, we introduce an effective pipeline to enable the identification of valid removal sheets (Section 4.2) and their ranking (Section 4.3), and the extraction of a quad surface for the removal of a candidate sheet (Section 4.4). After simplifying the base-complex, both the resulting hex-mesh (Section 5.1) and its base-complex could be highly distorted (Figure 1(b)). We then extend the parameterization-based optimization from quad-meshes [Tarini et al. 2011] to hex-meshes to further optimize the placement of singularities and reduce the distortion of the hex-meshes (Section 5.3, Figure 1(c)).

2 Related Work

Volume parameterization and hex-meshing: A variety of methods exist to generate unstructured hexahedral meshes. For a thorough survey, please refer to [Shepherd and Johnson 2008]. Several recent approaches, including polycube based methods [Gregson et al. 2011; Livesu et al. 2013; Huang et al. 2014] and frame field based methods [Nieser et al. 2011; Huang et al. 2011; Li et al. 2012; Jiang et al. 2014], have been proposed to generate hex-meshes with relatively large components while having a high local element quality. By mapping models to axis-aligned polycubes, researchers introduced Polycuts [Livesu et al. 2013] and L1-Polycubes [Huang et al. 2014] to remove unnecessary small cubes produced by the conventional polycube method [Gregson et al. 2011]. Li et al. [2013] extended the conventional polycubes [Gregson et al. 2011] to generalized polycubes (GPC) to handle curved cuboid representations. However, the control of the interior structure of the hex-mesh is still missing. Huang et al. [2011] proposed a first automatic solution to create a boundary conformal 3D cross field via an expensive optimization. Nieser et al. [2011] pointed out that only 10 types of singularities can lead to a valid all-hex mesh. Li et al. [2012] and Jiang et al. [2014] introduced techniques to convert a general 3D cross field to a restricted field with only these 10 types of singularities. After regularizing the cross field and fixing degeneracies, high quality hex-meshes can then be generated using the CubeCover technique [Nieser et al. 2011] or solving a mixed-integer problem. However, as mentioned in [Li et al. 2012], the singularities in the obtained hex-meshes may not be aligned with each other, leading to many small components. This misalignment issue also arises in the hex-meshes generated by the polycube approaches.

Structure optimization: Many methods have been proposed to produce quad-meshes with high quality global structures. Please refer to [Bommes et al. 2013] for a survey of these techniques. In contrast, our work focuses on optimizing the global structures of hex-meshes by correcting the misalignment of singularities. In 2D cases, the misalignment of singularities in the global structure can be greedily tackled by the methods introduced in [Myles et al. 2010; Bommes et al. 2011; Tarini et al. 2011]. Specifically, Myles et al. [2010] introduced T-mesh to simplify the patch domain by allowing the existence of T-junctions. While Bommes et al. [2011] aligned the singularities by removing helix configurations, Tarini et al. [2011] simplified the patch domain by optimizing the connectivities of separatrices originating from the singularities. For hex-meshes, existing methods either simplify (or coarsen) the hex-meshes via local modifications of some hexahedral sheets [Borden et al. 2002] or alter the local areas of the hex-meshes via hexahedral duals [Tautgesa and Knoopb 2003]. Neither method can guarantee to reduce the number of components in the base-complexes.

3 Hex-mesh Structures and Misalignments

In this section, we first review the singularity structure of hex-meshes. Then, we extend the base-complex concept from 2D quad-meshes to 3D hex-meshes, and introduce a robust algorithm to extract it. Finally, we describe the misalignments in the base-complex.

3.1 Singularity Structure

Consider a hex-mesh $\mathcal{H} = (V, E, F, H)$, where V is a set of vertices, E is a set of edges, F is a set of faces, and H is a set of hexahedral elements. Throughout the paper, we define the valence of a vertex or an edge with respect to the number of its neighboring hexahedral elements. An edge is said to be *irregular* if its valence is not 2 (on the boundary) or not 4 (in the interior). A vertex is called regular if its valence is 4 (on the boundary) or 8 (in the interior); otherwise, it is an irregular vertex.

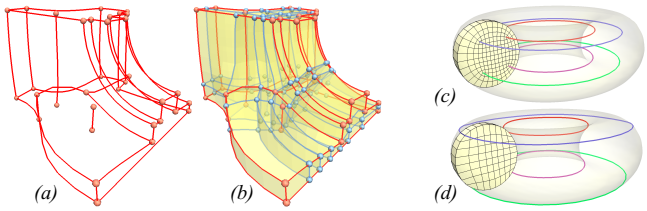


Figure 2: *The singularity structure (a) and base-complex (b) of a Fandisk hex-mesh. The singular nodes and singular edges of both the singularity structure and the base-complex are in red, while the non-singular nodes and regular edges of the base-complex are in blue. (c) shows four closed singular edges in the interior of a Torus hex-mesh, while (d) shows four closed singular edges on the boundary.*

Assume $\mathcal{H}$ is a valid manifold all-hex mesh, where each edge has a neighborhood that is homeomorphic to a cylinder or a half-cylinder, and each vertex has a neighborhood that is homeomorphic to a sphere or a half-sphere. Edges with half-cylinder neighborhoods and vertices with half-sphere neighborhoods are on the boundary. The boundary of a manifold hex-mesh is a closed two-manifold mesh.

With the above assumptions, the singularity structure of $\mathcal{H}$, denoted by $\mathcal{S}$, consists of a set of singular nodes and singular edges. A singular edge is a 1D curve composed of a sequence of connected irregular edges with the same valence, either in the interior (Figure 2(c)) or on the boundary (Figure 2(d)) of the hex-mesh. The singular edges can be classified into two types: open or closed (Figure 2(c-d)). Specifically, the end points of an open singular edge are singular nodes (i.e., the intersections of some singular edges or their intersections with the boundary). It cannot simply start or end in the interior of the volume; instead, it either hits the boundary or connects with other open singular edges via a singular node. By contrast, no singular node exists on a closed singular edge, and it

is completely either on the boundary or contained in the volume. Note that a singular node can be either regular or irregular.

The above definition of the singularity structure also self-describes an automatic algorithm to identify it. Figure 2(a) shows an example of the singularity structure of a Fandisk hex-mesh.

3.2 Base-Complex

After extracting the singularity structure, we now describe the definition and computation of the base-complex of a hex-mesh. Analogous to the base-complex of a quad-mesh [Bommes et al. 2011], the base-complex of a hex-mesh is an all-hexahedral structure.

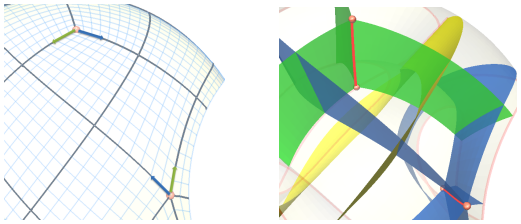
Similar to the curve separatrices in 2D [Tarini et al. 2011], which consist of edges in a quad-mesh, the surface separation structures starting from any singular edges are needed to define the base-complex of a hex-mesh. We refer to these surface separation structures as the *separation surfaces*. Each separation surface consists of a set of connected quads in the hex-mesh. For a singular edge with valence n, there are n separation surfaces originating from it. All the separation surfaces from singular edges form a surface graph network embedded in the hex-mesh, describing its topological structure.

We denote the base-complex of a hex-mesh $\mathcal{H}$ as $\mathcal{B} = (\mathcal{B}_V, \mathcal{B}_E, \mathcal{B}_F, \mathcal{B}_C)$ (Figure 2(b)). $\mathcal{B}_E$ is the set of the intersections of the separation surfaces. Each base-complex edge in $\mathcal{B}_E$, either singular or regular, consists of a sequence of connected hex edges of $\mathcal{H}$. The singular base-complex edges in $\mathcal{B}$ are illustrated as red curves in Figure 2(b), while the regular edges are illustrated as blue curves. $\mathcal{B}_V$ is the set of the end points of the base-complex edges in $\mathcal{B}_E$, which are either singular (red nodes in Figure 2(b)) or non-singular (blue nodes in Figure 2(b)). The reason to include non-singular nodes and regular base-complex edges is to remove T-junction configurations. $\mathcal{B}_F$ is a set of base-complex faces, each of which has four edges in $\mathcal{B}_E$. $\mathcal{B}_F$ partitions the domain of $\mathcal{H}$ into a set of components in $\mathcal{B}_C$. Each component is a cuboid-like sub-volume.

Algorithm 1 describes the pseudo code for extracting the base-complex $\mathcal{B}$ from a given hex-mesh $\mathcal{H}$. It is important to note that our defined base-complex is distinct from the one mentioned in [Livesu et al. 2013] which is actually a polycube structure.

Algorithm 1: Pseudo code of extracting $\mathcal{B}$ from $\mathcal{H}$

Input : $\mathcal{H}$
Output: $\mathcal{B}$
Extract $\mathcal{S}$ from $\mathcal{H}$;
foreach *singular edge* se_i *of* $\mathcal{S}$ **do**
  if se_i *is open* **then**
    Trace n separatrices for each of the two singular nodes of se_i, where n is the valence of se_i;
  if se_i *is closed* **then**
    Trace n separatrices for every irregular vertex on se_i;
  Trace n separation surfaces for se_i, which is guided by the separatrices;
Extract $\mathcal{B}_V$ and $\mathcal{B}_E$ from the intersections of all separation surfaces;
Extract $\mathcal{B}_F$ by decomposing separation surfaces into patches based on their intersections, i.e., $\mathcal{B}_E$;
Extract $\mathcal{B}_C$ by labeling hex elements that fall in different cuboid regions separated by $\mathcal{B}_F$ using a flooding algorithm.



(a) *(b)*

Figure 3: *A misaligned singularity pair on surface (a) and in volume domain (b), respectively. The singular nodes and edges are shown in red.*

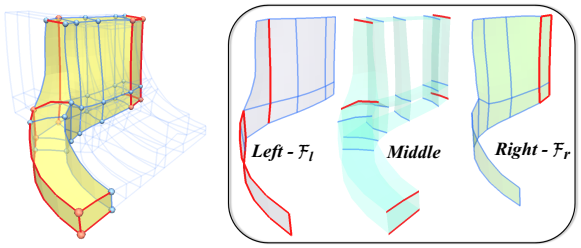
3.3 Misalignments in Base-Complex

Two singular edges are considered as aligned when there is a separation surface directly connecting them. Similar to regular nodes in the base-complexes of quad-meshes, which are caused by the mismatch of the separatrices of singular nodes in 2D (Figure 3(a)), the mismatch of the separation surfaces of singular edges in hex-meshes (Figure 3(b)) causes the arising of regular base-complex edges in 3D, and thus creates additional patches and components in the base-complex.

4 Alignment Algorithm

Given a hex-mesh $\mathcal{H}$ and its extracted base-complex $\mathcal{B}$, we propose an algorithm to simplify $\mathcal{B}$ by aligning mismatched singular edges in it. During the simplification, we maintain the singularity structure of $\mathcal{H}$, which means the numbers and valences of singular nodes and edges, and their connectivities remain unchanged.

4.1 Pipeline



(a) *(b)*

Figure 4: *A base-complex sheet (a) extracted from the base-complex of a Fandisk hex-mesh consists of three parts (b): a left surface ($\mathcal{F}_l$), a middle volume, and a right surface ($\mathcal{F}_r$). Regular and singular edges of the base-complex are shown in blue and red, respectively.*

When applying the hexahedral sheets from hex-meshes [Borden et al. 2002] to their base-complexes, we refer to them as the *base-complex sheets*. Figure 4(a) shows a base-complex sheet extracted from the base-complex of a Fandisk hex-mesh. All the components in a base-complex sheet can be removed by performing the hexahedral sheet collapse process introduced in [Borden et al. 2002]. However, since we want to maintain the singularity structure, we only remove those base-complex sheets that do not merge, remove, or create singularities in the singularity structure. For the sake of simplicity, in the remainder of this paper, we refer to those removable base-complex sheets as the *valid removal base-complex sheets*, or further abbreviated as the *candidates*.

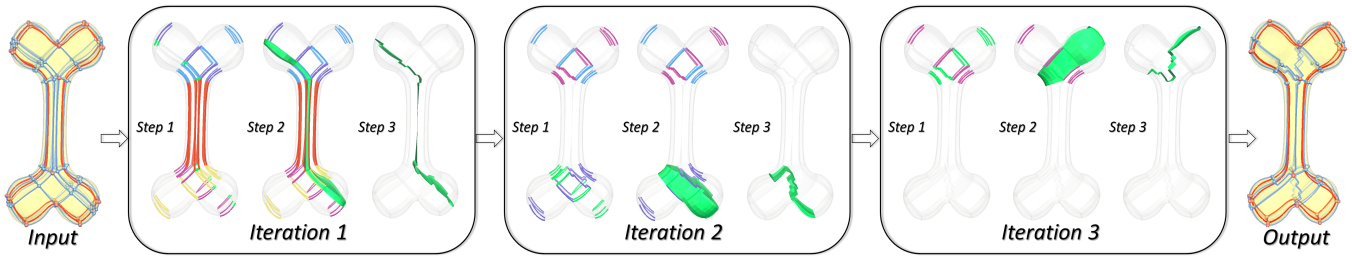


Figure 5: *Our alignment algorithm is applied to a Bone hex-mesh. Input: the singularity structure (red) and the constructed base-complex. Within each iteration, the results are shown in Step 1, Step 2, and Step 3, respectively. Because the resulting base-complex is valid for each iteration, the results of Step 4 are not shown. Output: the constructed aligned base-complex.*

Our alignment algorithm can be summarized as follows:

1. Detect all the valid removal base-complex sheets in the base-complex $\mathcal{B}$ (Section 4.2) and put them in a candidate list for removal;

2. Rank all the candidates in the list based on their shapes and numbers of components (Section 4.3);

3. Correct the top-ranked candidate to obtain a simplified base-complex, denoted by $\mathcal{B}'$ (Section 4.4);

4. Check the validity of $\mathcal{B}'$ (Section 3.2). If it is valid, assign $\mathcal{B}'$ to $\mathcal{B}$ and go to Step 1. Otherwise, discard $\mathcal{B}'$, re-parameterize $\mathcal{H}$ guided by $\mathcal{B}$ (Section 5.1) and go to Step 1.

Repeat the above process until no more candidates can be found or the number of corrected candidates meets p, which is the number of valid removal base-complex sheets that users want to remove.

At the above Step 4, the resulting simplified base-complex $\mathcal{B}'$ may not be valid due to an insufficient number of hexahedral element layers within the valid removal base-complex sheet (Figure 13). This can be well addressed by a re-parameterization step (Section 5.1). Note that this re-parameterization only inserts additional hexahedral element layers to satisfy the topology preservation described in Section 4.4.3. Figure 5 shows a pipeline overview of our alignment algorithm.

4.2 Candidate Detection

To identify these candidates, we first decompose each base-complex sheet into three parts: a left surface $\mathcal{F}_l$, a right surface $\mathcal{F}_r$, and a middle (volume) part, as shown in Figure 4(b). The volume part encloses a sequence of parallel base-complex edges (i.e., the blue and red edges in the middle image of Figure 4(b)). The left and right surfaces consist of those base-complex faces that do not share any parallel edges with the volume part.

We then consider a base-complex sheet as a candidate if none of the singular edges in its left surface is aligned with any singular edge in its right surface. Specifically, we check whether each base-complex edge in the volume part satisfies one of the following criteria: 1) if the base-complex edge is regular, its two end nodes cannot be on singular edges simultaneously; 2) if it is singular, it must be a part of a singular edge of the singularity structure (not the entire singular edge). Based on these criteria, the base-complex sheet shown in Figure 4 is a candidate. We can easily find all the candidates in the base-complex by performing this check. The leftmost image of iteration 1 in Figure 5 shows all the detected candidates for a Bone hex-mesh. The parallel edges with the same color represent one candidate.

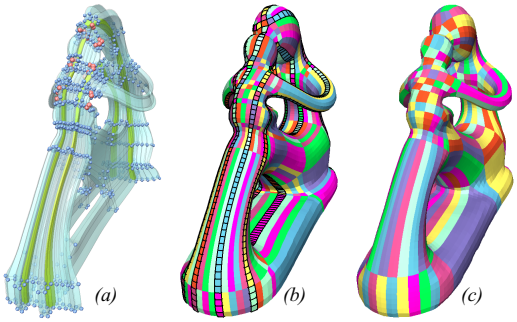


Figure 6: *(a) A single entangled candidate (crossing components in green); the singular nodes and non-singular nodes in $\mathcal{B}$ are in red and blue, respectively, (b) the original Fertility hex-mesh embedded with the entangled candidate (black edges), and (c) the optimized Fertility hex-mesh with a simplified base-complex.*

It is noteworthy that the candidates may be entangled with themselves to form complicated scenarios. Figure 6(a) shows such an example. For simplicity, we explain our algorithm using a scenario where the candidates are not entangled. Our alignment algorithm can also remove entangled candidates in a similar manner. Figure 6(c) shows the optimized Fertility that removes the entangled candidate existing in Figure 6(b).

4.3 Candidate Ranking

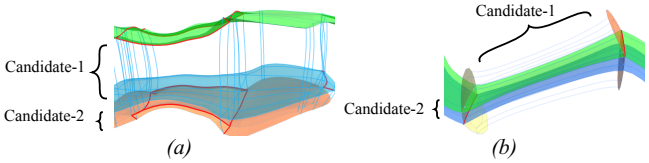


Figure 7: *Removing one candidate eliminates the other one as well: (a) two parallel candidates, and (b) two orthogonal candidates. The left and right surfaces of the candidates are highlighted with different colors.*

Figure 7(a) shows two neighboring candidates that are parallel to each other, while Figure 7(b) shows an orthogonal candidate pair. In either case, removing one candidate leads to the elimination of the other. However, removing these candidates in different orders may lead to distinct numbers of components in the resulting base-complex. Ideally, we want to first correct those candidates that can lead to a smaller number of components in the resulting base-complex. They should also have small and thin shapes, which can

lead to less mesh distortion after removal. To achieve these goals, we introduce a priority metric for a candidate, m,

$$W_m = A_m + \alpha {T_m}^2 + N_{\mathcal{B}'} \quad (1)$$

where A_m counts the number of quad elements in either the left or the right surface of m (Figure 4(b)), T_m measures the width of the middle part of m, which counts the number of the hex edges of any base-complex edge in the volume part, and $N_{\mathcal{B}'}$ is the number of components in the original base-complex minus the number of components in m. Here, α balances the importances of A_m and T_m. We use $\alpha = 6$ for all our experiments. The smaller this priority metric is, the higher a candidate is ranked.

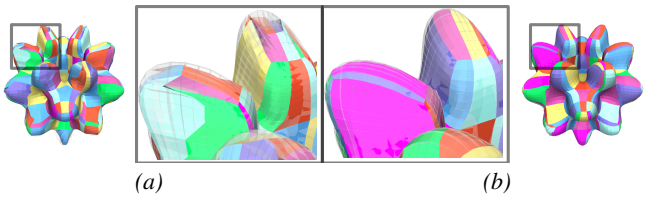


Figure 8: *Aligned hex-meshes by (a) simply collapsing the candidates without preserving the features on the surface boundary of the hex-mesh, and (b) our approach.*

4.4 Candidate Correction

The direct collapse of a candidate onto its dual plane [Borden et al. 2002] could remove the misalignments between the singular based-complex edges located at the left and right surfaces. While this procedure is ideal for some circumstances, it could result in hex-meshes with missing geometric features near the boundary (Figure 8(a)), which will be difficult to recover by optimization. In this section, we introduce a novel strategy to remove the misalignments in a candidate by directly connecting its contained singularities using a quad surface in a zig-zag manner. This quad surface consists of a sequence of patches, each of which is extracted from a component in the candidate. Each of these patches will become a base-complex face in the simplified base-complex.

For each component in the candidate, we extract a patch for the quad surface as follows. First, we determine the four corners of the patch, which is referred to as its *configuration* (Section 4.4.1). Figure 9(b) shows the computed configurations for all the components in the candidate shown in Figure 9(a). Second, based on this configuration, we extract the patch consisting of quads from the hex-mesh (Section 4.4.2). Figure 9(c) shows the extracted patches that form the quad surface. After extracting the quad surface, we reconstruct the simplified base-complex, as shown in Figure 9(d) (Section 4.4.3).

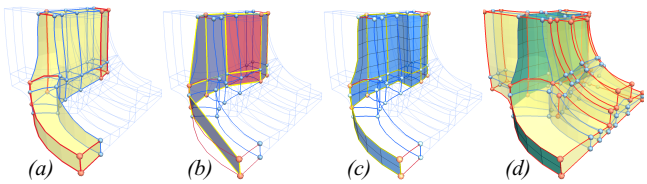


Figure 9: *(a) A candidate in a Fandisk hex-mesh; (b) the computed patch configurations, which are indicated by colored planes (blue for original configurations and red for diagonal configurations); (c) the extracted quad surface; and (d) the reconstructed simplified base-complex.*

Given a base-complex component that can be mapped to a regular cube with three principal directions (i.e., i, j, k), we denote its nodes, edges, and faces as $v_{i,j,k}$, $\varepsilon_{i,j}$, and τ_i, respectively. $\varepsilon_{i,j}$ connects nodes $v_{i,j,k}$ and $v_{i,j,k+1}$. τ_i and τ_{i+1} denote the two faces parallel to the jk plane (Figure 10).

4.4.1 Candidate Configuration

To extract a quad surface from a candidate, the following constraints are enforced: 1) if one or more edges of a patch of the quad surface are singular, then these edges should remain unchanged; otherwise, the aligned base-complex would not sit in the original hex-mesh any more, and 2) any two neighboring patches of the quad surface are connected via exactly one base-complex edge. Condition 1 is for singularity structure preservation, while Condition 2 is for the correct connectivities of both the quad-surface and the subsequent simplified base-complex.

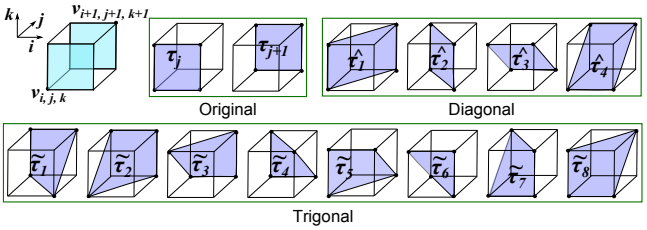


Figure 10: *A valid patch can be extracted from a component shown at the upper-left corner in three distinct cases: original (2 configurations), diagonal (4 configurations), and trigonal (8 configurations).*

For a component in the candidate (the upper-left of Figure 10), we assume the base-complex faces τ_j and τ_{j+1} belong to the left and right surfaces of the candidate, respectively. A patch can be extracted from the component in three topologically distinct cases: *original* (6 configurations in total), i.e., four nodes are on the same face of b_c, *diagonal* (6 configurations in total), i.e., two nodes on one face and the other two on another face, and *trigonal* (24 configurations in total), i.e., three nodes on one face and the fourth one on another face. For each of the three cases, we need to filter out those configurations that contain the base-complex edges belonging to the volume part of the candidate (e.g., the parallel edges in Figure 4(b), middle). Otherwise, the subsequent simplified base-complex will not be valid after the collapse process described in Section 4.4.3. The remaining valid configurations (14 in total) are: 2 *original*, 4 *diagonal*, and 8 *trigonal*. Figure 10 illustrates the connectivity of the four corners for each valid configuration.

To extract a patch from a component, only one configuration is needed. The determination of the configurations for all the components in the candidate is performed as follows. We first initialize all the components with all the above 14 valid configurations. Then, for those components that contain singular base-complex edges, we discard their configurations that do not satisfy both Condition 1 and Condition 2. For the other components, we discard their configurations that do not satisfy Condition 2. Once those unqualified configurations of a component are discarded, the configurations of its neighboring components will be updated accordingly based on Condition 2. By performing one iteration of this filtering process through all the components in the candidate, we remove all their unqualified configurations. However, after this process, some components may still have more than one qualified configuration remaining. In this case, we randomly select one of these components, and randomly choose one among its remaining qualified configurations, and propagate this update to the other components. By repeating this process, only one qualified configuration remains for

each component of the candidate (Figure 9(b)).

4.4.2 Quad Surface Extraction

After determining the configurations for the components in a candidate m, we now describe how to extract a quad surface $\mathcal{F}'_m$ by extracting its patches from the components in m. Each patch can be constructed independently. As shown in Figure 10, we denote the to-be-extracted patches for the three topologically distinct configuration cases as τ. (*original*), $\hat{\tau}$. (*diagonal*), and $\tilde{\tau}$. (*trigonal*), respectively. For each component b_c in m, the to-be-extracted patch will be either set as one of the two original faces of b_c (i.e., the original configurations), or newly extracted (i.e., the diagonal and trigonal configurations). In the former case, the patch will be either τ_j or τ_{j+1} of b_c (Figure 10). In the following we concentrate on the extraction of the patch from b_c whose configuration is diagonal. Trigonal configurations can be handled similarly.

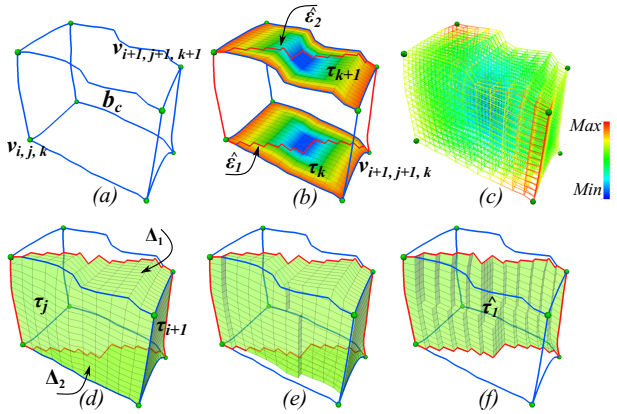


Figure 11: *A new base-complex face with a diagonal configuration can be constructed from a component (a) by first extracting two new base-complex edges (b); second, peeling hex elements gradually from the surface (d) to the interior (e) by measuring the weights calculated for them (c). The constructed patch is shown in (f).*

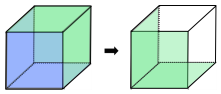
Assume the to-be-extracted patch is $\hat{\tau}_1$ corresponding to the first diagonal configuration in Figure 10, we now describe how to extract this patch. One example is illustrated in Figure 11.

First, as shown in Figure 11(b), we find the four edges of $\hat{\tau}_1$: two of them are the original base-complex edges, and the other two need to be newly extracted, i.e., $\hat{\epsilon}_1$ from τ_k and $\hat{\epsilon}_2$ from τ_{k+1}, respectively. Let us take the extraction of $\hat{\epsilon}_1$ as an example. Consider τ_k consisting of points and edges as an undirected graph, then $\hat{\epsilon}_1$ is the shortest valid path from $v_{i,j,k}$ to $v_{i+1,j+1,k}$. A path could be invalid if three edges on the path share a hexahedral element in the hex-mesh. The $\mathbf{k}_{th}$Dijkstra algorithm [Yen 1971] is employed to efficiently extract the shortest valid path.

Second, with the four edges (red curves in Figure 11(b)) extracted for $\hat{\tau}_1$, we find its interior quads that are located inside the component b_c in two steps.

As shown in Figure 11(c), we first calculate a weight function over the whole volume of b_c. The weight of a vertex v in b_c is calculated using $s(v) = d_{2e}(v) - d_c(v) - d_e(v) - d_f(v)$, where $d_{2e}(v)$, $d_c(v)$, $d_e(v)$, and $d_f(v)$ are the minimal distances from v to the two newly extracted edges $\hat{\epsilon}_1$ and $\hat{\epsilon}_2$, to the eight corners, to the twelve edges, and to the six base-complex faces of b_c, respectively. The distances are measured as the number of traveled hex edges, calculated through a bread-first search. The weight of each quad within b_c is then computed by averaging the weights of its four vertices. As shown in Figure 11(c), the weights become smaller from the corners corresponding to $\epsilon_{i,j+1}$ and $\epsilon_{i+1,j}$ to the center, as the color is gradually changed from red to blue.

Then, the interior of $\hat{\tau}_1$ can be obtained by a hex element peeling process. The surface of the component b_c is separated into two parts based on the four boundary edges of $\hat{\tau}_1$. Either one of the two parts can be used for initialization. Here, we initialize $\hat{\tau}_1$ as the partial surface consisting of faces τ_j, τ_{i+1} and patches Δ_1, Δ_2 (Figure 11(d)). If a quad within b_c is on $\hat{\tau}_1$ then we mark it as a boundary quad; otherwise, we mark it as an interior quad. All the hex elements in b_c are marked as valid.



Next, we check each valid hex element, h, that has at least a boundary quad. As shown in the inset figure, if the sum of the weights of all of its boundary quads (green, left) is larger than the sum of the weights of all of its interior quads (blue, left), then we mark all of its original boundary quads as invalid (white, right), set its original interior quads as new boundary quads (green, right), and label it as invalid. By repeating the above process, $\hat{\tau}_1$ continuously moves towards the center of b_c. When all the valid hex elements that contain boundary quads do not change their states any more, the final $\hat{\tau}_1$ is reached. Figure 11(e) shows an intermediate state of $\hat{\tau}_1$ during the above peeling. The final extracted $\hat{\tau}_1$ is shown in Figure 11(f). Figure 9(c) shows the constructed quad surface of a candidate of a Fandisk hex-mesh.

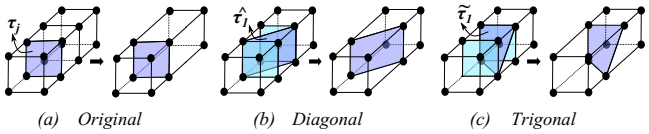


Figure 12: *Illustration of the collapse for the original (a), diagonal (b), and trigonal (c) configurations.*

4.4.3 Simplified Base-Complex

After extracting the quad surface, a new base-complex $\mathcal{B}'$ can be constructed by collapsing the candidate onto the quad surface (Figure 9(d)).

As shown in Figure 12(a), for a patch corresponding to an original configuration, if τ_j is selected, we discard τ_{j+1} and merge the component b_c with its neighbor that is adjacent to τ_{j+1}. If τ_{j+1} is on the surface boundary of the hex-mesh, we simply discard b_c. For a diagonal patch (e.g., $\hat{\tau}_1$ in Figure 12(b)), we reconstruct its local base-complex by removing the two original base-complex faces (in cyan) and inserting the diagonal patch (in blue). The trigonal patch (e.g., $\tilde{\tau}_1$ in Figure 12(c)) can be handled in a similar fashion to the diagonal patch.

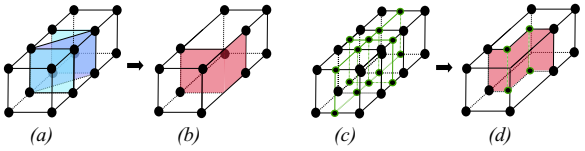


Figure 13: *(a) shows a diagonal configuration of a component that is comprised of only a single hex element. The two cyan quads belong to the left and right surfaces of the candidate, respectively. (b) shows the actual extracted patch that would lead to an invalid $\mathcal{B}'$. (c) illustrates the same component with additional refined hex elements, and (d) is the resulting patch with the correct connectivity.*

Topology preservation: During the quad surface extraction step described in Section 4.4.2, while the *original* configurations work

perfectly, there is a constraint for both the *diagonal* and *trigonal* configurations. Figure 13(a) shows a diagonal configuration where the component in the middle contains only one hexahedral element. The two quads in cyan are parts of the left and right surfaces of the base-complex sheet. Figure 13(b) shows its actual extracted patch (red), which would lead to an invalid base-complex after collapse. Figure 13(c-d) illustrate that this issue can be easily resolved by inserting an additional hex element layer.

To avoid an invalid simplified base-complex, if the configuration of a component is *diagonal*, then two of its three principal directions must have a resolution of at least two hex elements; if the configuration of a component is *trigonal*, then all of its three principal directions must have a resolution of at least two hex elements. As described in Step 4 of our alignment algorithm (Section 4.1), this can be satisfied through the re-parameterization (see Section 5.1) of the hex-mesh to increase the number of hex element layers in the candidates.

5 Parameterization and Optimization

After removing misalignments in the base-complex, we re-parameterize the volume of the model based on the simplified base-complex (Section 5.1) and generate a new hex-mesh (Section 5.2). The simplified base-complex and its resulting hex-mesh may exhibit certain geometric artifacts (Figure 1(b)). To improve them, we propose a structure-aware re-parameterization algorithm (Section 5.3).

5.1 Component-Wise Volume Parameterization

As the simplified base-complex is always valid based on the above processing (Section 4), each hex element of the hex-mesh belongs to only one component in the simplified base-complex. The parameterization of the volume of the model is performed within each component of the base-complex. Each base-complex face can be mapped to a planar rectangular parametric domain, and each component can be mapped to an axis-aligned cube domain. A component can be re-meshed into a tetrahedral mesh by subdividing each hex element into five tets [Hacon and Tomei 1989]. Its parameterization proceeds by assigning parametric values to its eight corners, to its twelve base-complex edges, to its six base-complex faces, and finally to its volume, respectively. Mean value coordinate techniques for 2D [Floater 2003] and 3D [Ju et al. 2005] are used to calculate the parametric coordinates of the vertices inside the base-complex faces and inside the components, respectively.

5.2 Discretization of Parameterization

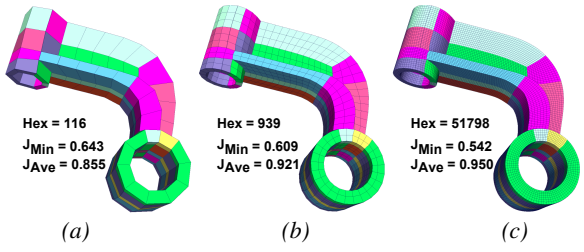


Figure 14: *Multi-resolution hex-meshes can be generated by only providing different user-specified element numbers ($N_{\mathcal{H}}$). They are: (a) 100, (b) 1000, and (c) 50000, respectively.*

Typically, the discretization resolution of the parameterization is determined by a scalar value (e.g., the average edge length). However, it is impossible for users to know beforehand how many hex elements will be produced. Given the desired number of elements, $N_{\mathcal{H}}$, we present a strategy to re-parameterize the mesh such that its element number is reasonably close to $N_{\mathcal{H}}$. Recall that the resolution of a component is determined by the resolutions along the three parameterization directions. Therefore, as long as we can determine the resolution of each base-complex edge, the total element number of the final hex-mesh can be soundly estimated. Assume N_m is the number of base-complex sheets in the base-complex. Since all the parallel base-complex edges in the volume part of a base-complex sheet are expected to have the same resolution, we can categorize them as one group; as a result, we have a total of N_m groups of base-complex edges. For a group of parallel edges, we calculate its representative length, δ, as follows.

$$\delta = \frac{\sum_t \frac{l_{E_t}}{l_{\bar{e}_t}}}{n_c} \tag{2}$$

Here, n_c is the number of base-complex edges in the group, and δ is the ratio between the sum of the lengths of all the n_c base-complex edges (l_{E_t}) in the group and the average hex-edge length ($l_{\bar{e}_t}$) of the original hex-mesh $\mathcal{H}$. Assume the resolutions along the three directions of a component are $w\delta_i$, $w\delta_j$, and $w\delta_k$, respectively, where w is a scalar weight. The number of hexahedral elements contained in the component will be $w^3\delta_i\delta_j\delta_k$. Let $N_{\mathcal{H}}$ equal to the total number of elements in the resulting hex-mesh, which is $\sum w^3\delta_i\delta_j\delta_k$, by summing up all the elements in all the components in the base-complex, w can be calculated as follows.

$$w = \sqrt[3]{\frac{N_{\mathcal{H}}}{\sum \delta_i\delta_j\delta_k}} \tag{3}$$

Thus, to produce a hex-mesh with approximate $N_{\mathcal{H}}$ elements, each base-complex edge in the same base-complex edge group should contain $\lceil w\delta \rceil$ hex-edges. Figure 14 shows three hex-meshes of Hanger model with different user-specified numbers of hexahedal elements.

5.3 Global Optimization

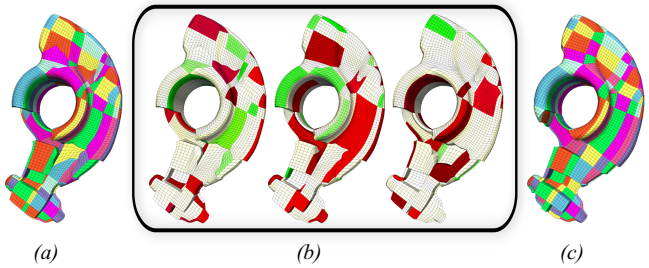
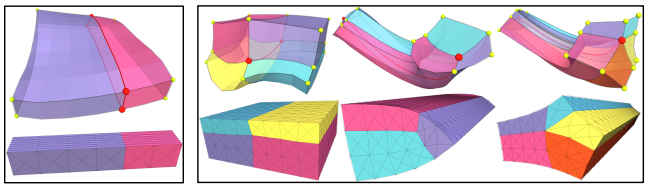


Figure 15: *(a) The re-parameterized hex-mesh after alignment for a Rockerarm model before optimization. Distinct components are colored differently. (b) Different sets of parameterization domains for each iteration. Isolated edge and face domains are colored in red and green, respectively; the domains colored in white are fixed for the current iteration. (c) The optimized hex-mesh.*

The initially re-parameterized hex-mesh after misalignment correction may contain a large distortion (see Figure 1(b) and Figure 15(a)), which is less suitable for downstream applications. Therefore, to obtain a high quality hex-mesh, a post-processing step such as optimization may be required. For certain cases, commonly used optimization techniques, such as various Laplacian smoothing and geometric flow [Zhang et al. 2005], may push some interior

vertices onto the boundary, which would pose challenges for subsequent untangling processing [Brewer et al. 2003]. The reason is that, after alignment, singularities may no longer be at their optimal locations. To improve the quality, we perform a structure-aware global parameterization over the base-complex domain, as detailed below. Our method efficiently optimizes the base-complex structure while ensuring that those geometrically and topologically interior nodes in the input volume remain inside after parameterization. The quality of the generated hex-mesh is consequently improved.



(a) (b)

Figure 16: *Base-complex patch (a) and edge (b) based parameterization domains. Domains shown in (b), from left to right, have valence 4, 3, and 5, respectively.*

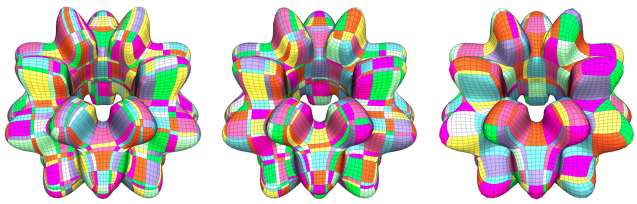
For each interior base-complex edge or face, we build a parameterization domain, as shown in Figure 16. This strategy is inspired by [Dong et al. 2006; Pietroni et al. 2010; Tarini et al. 2011], which introduced interpolation-space parameterizations for 2D surfaces. Specifically, for a base-complex face, its parameterization domain stitches its two neighboring components together to form a larger component, which can be mapped to a regular cuboid. For a base-complex edge $b_{e_i} = (b_{v_i}, b_{v_j})$ with valence n, its adjacent n components form a unified parameterization domain. In this parameterization domain, for the axes i, j, and k, let the k axis points in the direction of b_{e_i}, and the i, j axes are perpendicular to b_{e_i}. Assume b_{v_i} is at the origin $O(0, 0, 0)$, then b_{v_j} will be $(0, 0, k)$. For each ij layer, its parametric coordinates are expressed in the form of polar coordinates (ρ, θ), and further transformed to $(\rho^t, t\theta)$ via the exponential map, where $t = 4/n$.

Through face and edge based domain re-parameterization, distorted base-complex faces and edges can be optimized. With the above parameterization domains, we perform our global optimization iteratively, i.e., the edge and patch parameterization domains are executed alternately. As shown in Figure 15(b), at each iteration, the parameterization domains are set to be isolated from each other, thus parallel techniques can be employed to significantly speed up the optimization performance.

Surface feature preserving: Typically no feature exists inside the volume unless users define one. In this work, we only tackle the isotropic volume space; thus, only boundary features are considered. Surface features can be preserved by projecting specific parameterizaiton lines onto the detected features of the boundary surface. After global optimization, the hex-mesh is further improved using the Mesquite software [Brewer et al. 2003].

6 Results

We have applied the proposed approach to several datasets, including the hex-meshes provided by the authors of [Gregson et al. 2011; Li et al. 2012; Livesu et al. 2013; Huang et al. 2014]. The datasets cover a spectrum of man-made and natural objects, with various complexities. Table 1 provides the statistics of the tested hex-meshes before and after alignment, including: the number of hex elements($|\mathcal{H}|$), the detected and corrected valid removal candidates (|m|), the average and minimal Scaled Jacobians (S. J.), and the number of components ($|\mathcal{B}_C|$) in the base-complex. We removed all the detected candidates shown in Table 1. The improvement of singularity alignment is measured as the ratio between $|\mathcal{B}_C| - |\mathcal{B}'_C|$ and $|\mathcal{B}_C|$, which is also provided for each mesh (i.e., the AR column in Table 1). As shown in the $|\mathcal{B}_C|$ column in Table 1, we can see that our alignment algorithm significantly simplifies the base-complexes of the tested hex-meshes, while reliably preserving local element quality (the S. J. column in Table 1). Figure 17 provides visual results before and after optimization on some of the hex-meshes listed in Table 1. All the other results can be found in supplemental material. The hexahedral elements that belong to the same components of the base-complex are shown in the same colors. In our experiments, the computational time varied from one minute (e.g., Fandisk model) to half an hour (e.g., Dragon model). All the timing information was recorded on a PC with an Intel Xeon (E5-1620) processor and 16GB memory.



(a) Input, $|\mathcal{B}_C| = 2518$ (b) p = 7, $|\mathcal{B}_C| = 2172$ (c) p = 26, $|\mathcal{B}_C| = 590$

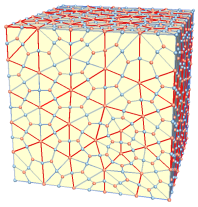
Figure 18: *p controls the simplicity of the optimized global structure. The larger the p value, the smaller the number of components in the resulting simplified base-complex.*

User control: Our alignment algorithm is intuitive for users to control. Although we removed all the detected misalignments for the hex-meshes shown in Table 1, we also allow users to specify a parameter p to decide how many candidates they want to remove. The Bumpy torus model (Figure 18) is used to demonstrate different simplification results controlled by p.

7 Conclusion and Discussion

In this paper, we introduce a complete framework to reduce the number of components in the base-complex of a hex-mesh by preserving its singularity structure. We have applied our framework to numerous hex-meshes to demonstrate its effectiveness.

Limitations: Similar to the alignment problem on surface domain, the major limitation of our work lies in that for an input hex-mesh, if the singularities are at a high order, the number of singularities is large, or the singularities are badly placed, its simplified base-complex will still be too complicated to be used for some applications. The inset shows such an example, where the hex-mesh is directly subdivided from a tet-mesh. However, the complexities of the base-complexes of most tested hex-meshes (Table 1) are reduced, unless the input hex-mesh already has a well aligned singularity-structure, such as the Sculpture-B model shown in Table 1. In this case, our framework directly performs the optimization step, while the hex-mesh after optimization cannot guarantee a higher geometric quality than the original input. Another limitation is that, although all the valid removal base-complex sheets in the tested hex-meshes can be detected and corrected by our approach, other unreported valid removal configurations may exist. Furthermore, in theory, our approach cannot guarantee the optimized base-complex is free of any



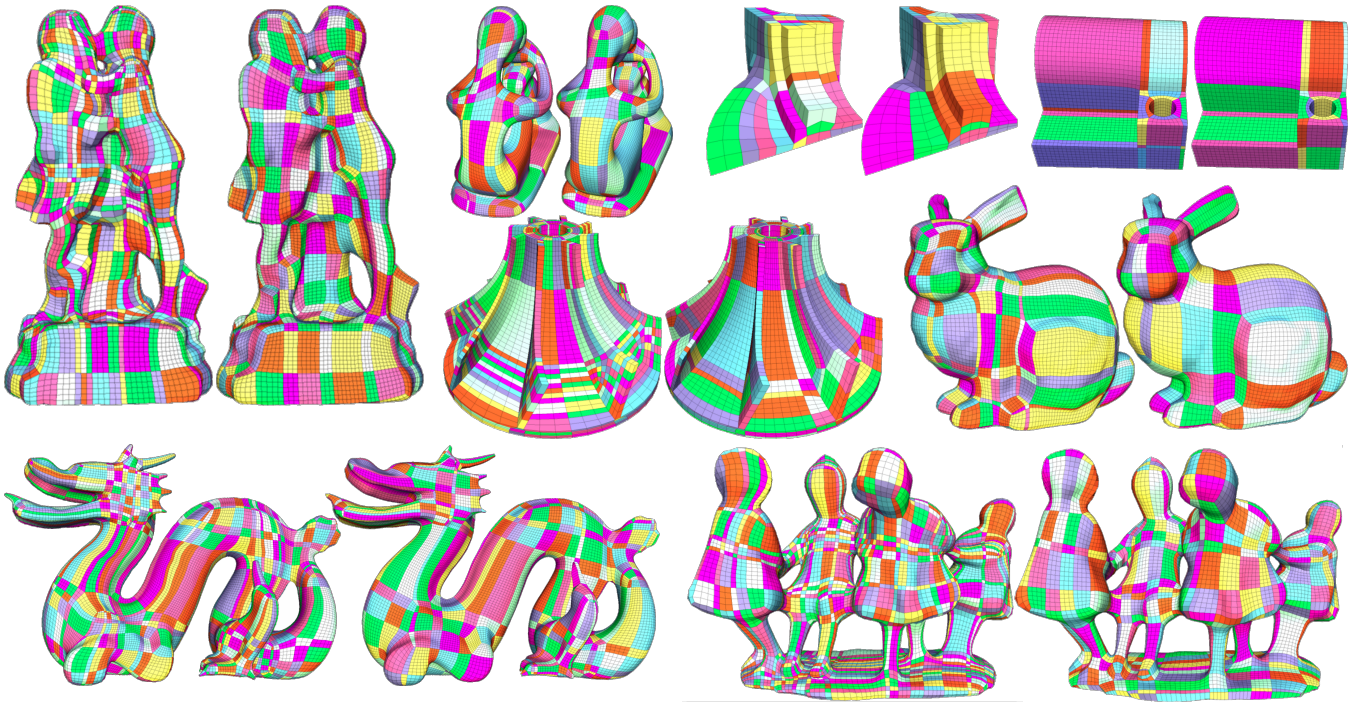


Figure 17: *The hex-meshes (from left to right and top to bottom) of the Kiss [Gregson et al. 2011], Fertility [Livesu et al. 2013], Fandisk, Joint, Impeller [Li et al. 2012], Bunny [Livesu et al. 2013], Dragon, Dancing children [Huang et al. 2014] before (the left of each model) and after misalignment correction and optimization (the right of each model).*

Models	$\|\mathcal{H}\|$	|m|	S. J.	$\|\mathcal{B}_C\|$	**AR**
girl/BU*	193k **44k**	16	.925/.235 **.894/.121**	1098 **401**	**63%**
Bumpy-torus* (Fig. 8(b), 18(c))	35k **35k**	33	.891/.271 **.866/.189**	2518 **590**	**77%**
Bunny*	82k **33k**	7	.930/.138 **.906/.200**	1324 **240**	**82%**
Fertility-refine*	20k **22k**	15	.911/.196 **.884/.364**	598 **390**	**35%**
Kiss-coarse* (Fig. 17)	27k **59k**	24	.901/.163 **.910/.190**	3690 **1365**	**63%**
Bone†	3k **9k**	6	.930/.620 **.924/.577**	87 **48**	**45%**
Bunny†	134k **96k**	2	.940/.293 **.930/.276**	259 **184**	**29%**
Rod†	6k **7k**	1	.947/.658 **.937/.527**	66 **33**	**50%**
Sculpture-A†(Fig. 1)	24k **24k**	6	.961/.689 **.890/.426**	51 **16**	**69%**
Fandisk†(Fig. 17)	.4k **.8k**	3	.936/.609 **.940/.413**	49 **30**	**39%**
Fertility†(Fig. 6)	14k **28k**	7	.911/.351 **.88/.300**	1352 **934**	**31%**
Hanger†	5k **9k**	1	.964/.599 **.950/.448**	60 **41**	**32%**
Impeller†(Fig. 17)	11k **19k**	8	.924/.185 **.925/.238**	944 **399**	**58%**
Joint†(Fig. 17)	18k **18k**	4	.984/.729 **.983/.724**	83 **59**	**29%**
Rockerarm†	11k **18k**	9	.866/.209 **.862/.145**	578 **291**	**50%**
Sculpture-B†	6k **6k**	0	.892/.055 **.889/.049**	51 **51**	**0%**

Models	$\|\mathcal{H}\|$	|m|	S. J.	$\|\mathcal{B}_C\|$	**AR**
Gargoyle♯	26k **23k**	52	.906/.196 **.911/.214**	7563 **1920**	**75%**
Angel-1♯	14k **15k**	11	.923/.470 **.915/.296**	1284 **698**	**46%**
Angel-2♯	16k **16k**	5	.919/.212 **.914/.260**	302 **196**	**35%**
Angel-3♯	14k **17k**	2	.898/.222 **.867/.157**	78 **56**	**28%**
Bumpy-torus♯	39k **39k**	23	.929/.335 **.879/.270**	2254 **910**	**60%**
Bunny♯	38k **48k**	2	.926/.382 **.937/.373**	273 **221**	**19%**
Bustle♯	12k **22k**	3	.934/.302 **.929/.393**	348 **292**	**16%**
Dancing-children♯(Fig. 17)	35k **38k**	48	.870/.143 **.876/.190**	5482 **1458**	**73%**
Dragon♯(Fig. 17)	118k **113k**	22	.857/.150 **.899/.120**	3977 **1958**	**51%**
Elephant♯	172k **55k**	26	.878/.221 **.890/.170**	2842 **1008**	**65%**
Rockerarm-1♯(Fig. 15)	24k **26k**	12	.920/.439 **.890/.329**	686 **395**	**42%**
Rockerarm-2♯	25k **26k**	13	.905/.378 **.899/.366**	835 **263**	**69%**
Bunny‡(Fig. 17)	74k **46k**	13	.938/.274 **.928/.263**	580 **194**	**67%**
Fertility‡(Fig. 17)	54k **74k**	11	.872/.259 **.885/.272**	693 **396**	**43%**
Girl/BU‡	56k **77k**	9	.926/.401 **.899/.279**	580 **252**	**57%**
Rockerarm‡	57k **57k**	10	.890/.370 **.893/.297**	664 **335**	**50%**

Table 1: *The number of components of base-complex and quality comparisons of hex-meshes before and after alignment. For each model, the original hex-mesh and optimized hex-mesh by our method are shown in the upper row and bottom row, respectively. Original hex-meshes with *, †, ♯, and ‡ are obtained from [Gregson et al. 2011; Li et al. 2012; Huang et al. 2014; Livesu et al. 2013], respectively.*

misalignments. We plan to address these limitations in our future work.

Acknowledgments

This work is supported in part by US NSF IIS-1352722, NIH 1R21HD075048-01A1, and National Natural Science Foundation of China (NSFC) Grant 61328204. We wish to thank Chang H. Yun for his help of proofreading. We would also like to thank the anonymous reviewers for their constructive comments.

References

BAZILEVS, Y., BEIRAO DA VEIGA, L., COTTRELL, J., HUGHES, T., AND SANGALLI, G. 2006. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences 16*, 07, 1031–1090.

BOMMES, D., LEMPFER, T., AND KOBBELT, L. 2011. Global structure optimization of quadrilateral meshes. *CGF 30*, 2, 375–384.

BOMMES, D., LÉVY, B., PIETRONI, N., PUPPO, E., SILVA, C., TARINI, M., AND ZORIN, D. 2013. Quad-mesh generation and processing: A survey. *CGF 32*, 6, 51–76.

BORDEN, M. J., BENZLEY, S. E., AND SHEPHERD, J. F. 2002. Hexahedral sheet extraction. In *Proc. of the 11th International Meshing Roundtable*, 147–152.

BREWER, M., DIACHIN, L., KNUPP, P., LEURENT, T., AND MELANDER, D. 2003. The mesquite mesh quality improvement toolkit. In *Proc. of the 12th International Meshing Roundtable*, 239–250.

DONG, S., BREMER, P.-T., GARLAND, M., PASCUCCI, V., AND HART, J. C. 2006. Spectral surface quadrangulation. *ACM Trans. Graph. 25*, 3 (July), 1057–1066.

FLOATER, M. S. 2003. Mean value coordinates. *Computer Aided Geometric Design 20*, 1, 19–27.

GAO, X., HUANG, J., LI, S., DENG, Z., AND CHEN, G. 2014. An evaluation of the quality of hexahedral meshes via modal analysis. In *1st Workshop on Structured Meshing: Theory, Applications, and Evaluation*.

GREGSON, J., SHEFFER, A., AND ZHANG, E. 2011. All-hex mesh generation via volumetric polycube deformation. *CGF 30*, 5, 1407–1416.

HACON, D., AND TOMEI, C. 1989. Tetrahedral decompositions of hexahedral meshes. *Eur. J. Comb. 10*, 5 (Aug.), 435–443.

HUANG, J., TONG, Y., WEI, H., AND BAO, H. 2011. Boundary aligned smooth 3d cross-frame field. *ACM Trans. Graph. 30*, 6 (Dec.), 143:1–143:8.

HUANG, J., JIANG, T., SHI, Z., TONG, Y., BAO, H., AND DESBRUN, M. 2014. L1-based construction of polycube maps from complex shapes. *ACM Trans. Graph. 33*, 3, 25:1–25:11.

HUGHES, T. J., COTTRELL, J. A., AND BAZILEVS, Y. 2005. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering 194*, 4135–4195.

JIANG, T., HUANG, J., YUANZHEN WANG, Y. T., AND BAO, H. 2014. Frame field singularity correction for automatic hexahedralization. *IEEE TVCG 20*, 8 (Aug.), 1189–1199.

JU, T., SCHAEFER, S., AND WARREN, J. 2005. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph. 24*, 3 (July), 561–566.

LEONARD, B., PATEL, A., AND HIRSCH, C. 2000. Multigrid acceleration in a 3d navier-stokes solver using unstructured hexahedral meshes with adaptation. In *Multigrid Methods VI*. Springer, 150–156.

LI, B., AND QIN, H. 2012. Full component-aware tensor-product trivariate splines of arbitrary topology. *Comput. Graph. (SMI 2012) 36*, 5 (Aug.), 329–340.

LI, Y., LIU, Y., XU, W., WANG, W., AND GUO, B. 2012. All-hex meshing using singularity-restricted field. *ACM Trans. Graph. 31*, 6 (Nov.), 177:1–177:11.

LI, B., LI, X., WANG, K., AND QIN, H. 2013. Surface mesh to volumetric spline conversion with generalized poly-cubes. *IEEE TVCG 19*, 9, 1539–1551.

LIVESU, M., VINING, N., SHEFFER, A., GREGSON, J., AND SCATENI, R. 2013. Polycut: monotone graph-cuts for polycube base-complex construction. *ACM Trans. Graph. 32*, 6, 171.

MARTIN, T., COHEN, E., AND KIRBY, M. 2008. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. In *ACM SPM '08*, 269–280.

MOTOOKA, Y., NOGUCHI, S., AND IGARASHI, H. 2011. Evaluation of hexahedral mesh quality for finite element method in electromagnetics. *Materials Science Forum 670*, 318–324.

MYLES, A., PIETRONI, N., KOVACS, D., AND ZORIN, D. 2010. Feature-aligned t-meshes. *ACM Trans. Graph. 29* (July), 117:1–117:11.

NIESER, M., REITEBUCH, U., AND POLTHIER, K. 2011. Cubecover- parameterization of 3d volumes. *CGF 30*, 5, 1397–1406.

PIETRONI, N., TARINI, M., AND CIGNONI, P. 2010. Almost isometric mesh parameterization through abstract domains. *IEEE TVCG 16*, 4 (July), 621–635.

SHEPHERD, J. F., AND JOHNSON, C. R. 2008. Hexahedral mesh generation constraints. *Eng. with Comput. 24*, 3 (June), 195–213.

TARINI, M., PUPPO, E., PANOZZO, D., PIETRONI, N., AND CIGNONI, P. 2011. Simple quad domains for field aligned mesh parametrization. *ACM Trans. Graph. 30*, 6 (Dec.), 142:1–142:12.

TAUTGESA, T. J., AND KNOOPB, S. E. 2003. Topology modification of hexahedral meshes using atomic dual-based operations. *Algorithms 11*, 12.

WADA, Y., SHINBORI, J., AND KIKUCHI, M. 2006. Adaptive fem analysis technique using multigrid method for unstructured hexahedral meshes. *Key Engineering Materials 306*, 565–570.

WANG, K., LI, X., LI, B., XU, H., AND QIN, H. 2012. Restricted trivariate polycube splines for volumetric data modeling. *IEEE TVCG 18*, 5, 703–716.

YEN, J. Y. 1971. Finding the k shortest loopless paths in a network. *Management Science 17*, 11, 712–716.

ZHANG, Y., BAJAJ, C., AND XU, G. 2005. Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. *Communications in Numerical Methods in Engineering 25*, 1, 1–18.