Expanding the Solvable Space of Polycube-Map via Validity-Enhanced Construction

Lu He* Na Lei* Ziliang Wang* Chen Wang* Xiaopeng Zheng† Zhongxuan Luo*

Abstract

Polycube-maps are highly valuable in computer graphics, particularly concerning hexahedral meshes. Currently, the validity of polycube is determined based on Steinitz and Eppstein's approach, which only addresses 3-connected graphs. However, extending the features of polycubes to encompass 4- and 6-connected graphs holds significant practical importance. We strengthen the validity structure of the polycube-map, providing a polycube validity theorem based on the Gauss-Bonnet theorem, offering both global and local conditions. Our theorem allows for 4- and 6-connected cases, expanding the existing solvable space of 3- and 5-singularity polycube-map to include 3-, 5-, and 6-singularity scenarios. Our theorem effectively addresses models that, while not meeting the existing valid polycube criteria, are indeed valid polycube polyhedra in practice. Additionally, within the label optimization framework, we introduce the Immune Genetic Algorithm (I-GA algorithm) tailored to our theorem, enabling the robust generation of polycube-map. We evaluate our method using thingi10k and ABC datasets. Results demonstrate that our validity theorem expands the solvable space of polycubes while achieving higher quality all-hexahedral meshing for models with a singularity level of six. Furthermore, we discuss the limitations associated with our proposed method.

1 Introduction.

The polycube-map plays a crucial role in the field of computer graphics, as it serves multiple purposes such as texture mapping [1, 2, 3], All-hexahedral mesh generation [4, 5, 6], and trivariate spline fitting [7, 8]. The polycube has a highly regular topology and a distinct global parameter domain. It represents a specialized geometry wherein each surface element aligns with one of the principal axes ($\pm xyz$). Moreover, the chart of the polycube can be identified through clusters of adjacent elements sharing identical labels. The valid polycube topology, or structure, specifies the feasible solution space of polycube-maps. Therefore, it is meaningful to study the valid polycube topology and how to obtain a high-quality polycube-map under the validity constraint.

The typical definition of the validity of polycube is based on the Steinitz Theorem [9, 10] and the Eppstein Theorem [11]. Steinitz's theorem asserts that a graph is the skeleton graph of a convex polyhedron if and only if it is 3-connected planar. Eppstein, on the other hand, proposed a specific analogue theorem for non-convex polyhedra, known as simple orthogonal polyhedra, which satisfy three conditions: 1) sphere topology, 2) singly connected faces, and 3) precisely three mutually orthogonal axis-parallel edges meet at each vertex.

A 3-connected vertex is a vertex where precisely three axis-parallel edges meet. Similarly, 4-connected and 6-connected denote vertices where four and six-axis-parallel edges meet, respectively. Due to the similarity between the graph properties established by Steinitz and Eppstein and the shapes of polycube polyhedron, scholars traditionally defined a valid polycube topology as 3-connected graphs, with the additional requirement that neighbouring charts cannot be assigned opposing labels. However, this validity definition has limitations, as illustrated in fig. 1 . There exist instances of polycube polyhedron that do not adhere to this definition of polycube validity topology. Despite [12] expansion of the polycube definition, which considered multiple genus and non-singly connected graph properties based on Eppstein's work, the restriction to 3-connected graphs remains in place.

As mentioned in the [13], extending polycube features to 4- and 6-connected graphs holds significant practical importance, especially for CAD designs, where three or more sharp features frequently converge simultaneously. Such layouts cannot be captured within traditional polycube structures, inevitably leading to a lack of feature preservation and, very likely, severe and unnecessary geometric distortion. Thus far, achieving comprehensive graph characteristics for generating polycubes has proven elusive. This remains an open problem not only for mesh generation practitioners but also for the broader community.

We propose a more encompassing validity topol-

*Dalian University of Technology.
†Corresponding author. zhengxp@dlut.edu.cn.

Copyright © 2024 by SIAM
Unauthorized reproduction of this article is prohibited

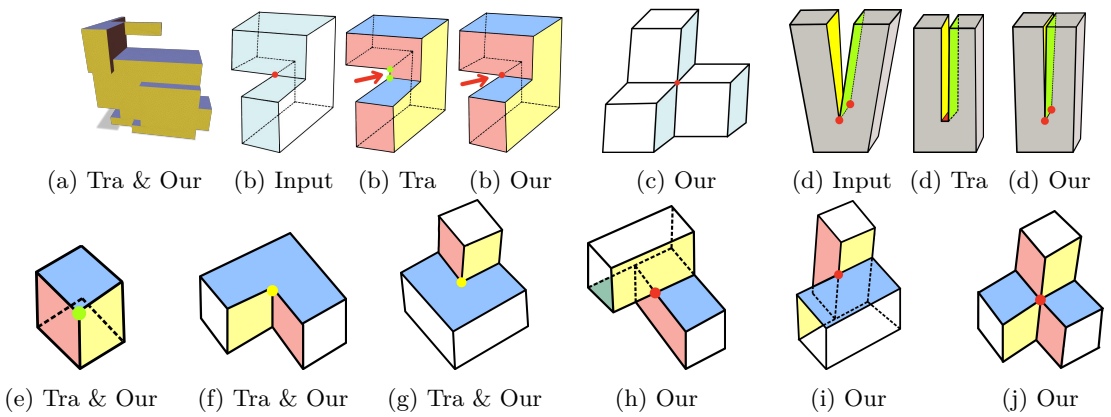


Figure 1: Comparison between the traditional polycube solvable space and our approach: (a) Traditional methods can solve the 3-connected polycube, as shown in (a). A 3-connected vertex has precisely three axis-parallel edges meeting at it. Similarly, 4-connected and 6-connected vertices involve four and six-axis-parallel edges meeting, respectively. (b). For the red-marked 4-connected vertices in (b)-input, traditional polycube mapping methods can only decompose them into 3-connected components, while our method can directly solve 4-connected vertices. (c)Traditional methods cannot handle 6-connected vertices, as exemplified by the red-marked vertex in (c). Nevertheless, our validity construction is capable of addressing models of this nature. (d) In cases where adjacent chart regions are labelled with opposing labels, traditional methods are deemed invalid polycube, which needs label repairs. In contrast, our validity construction allows for such cases, resulting in improved polycube mapping. (e) ~ (j) enumerate various corner scenarios. Traditional methods can only handle cases (e) ~(g) (3-connected), while our validity construction can handle cases (e) ~(j) (3-, 4-, and 6-connected).

ogy for polycube polyhedron. Our defined Validity-Enhanced topology inherently considers the properties of polycube polyhedron, allowing for 4- and 6-connected graph scenarios and permitting adjacent charts to have opposing principal axes within the confines of our provided conditions.

With the definition of the valid polycube topology in place, it can be employed to guide the generation of corresponding polycube structures for either surface or volumetric meshes. There are primarily two methods for generating polycubes: deformation-based and label optimization-based approaches. The former utilizes dedicated deformation energies that iteratively modify the object, causing surface normal to rotate until they align with the global coordinate axes. The latter method assigns a label to each surface element within the input (tetrahedral) mesh, representing one of the six global axes ($\pm$X, $\pm$Y, $\pm$Z).

However, it's worth noting that not every labelling allows for a corresponding polycube. To address this issue, various correction procedures have been suggested, which can be applied either as a post-processing step or integrated seamlessly with the labelling process The work [14] employs a rotation-driven approach and position-driven deformation to achieve planarity and alignment with the six global axes. Additionally, the authors of [15] and [2] achieve this by designing deformation energies that gradually deform the object. This process causes surface normals to rotate until they align precisely with the global coordinate axes. However, it's important to note that enforcing strict constraints against flips could potentially restrict the available deformation space. In some cases, this restriction may make it difficult to find a suitable mapping that complies with the specified boundary-alignment constraints unless further mesh refinement capabilities are introduced. The work by [16] integrates multi-label graph-cut optimization into a local search algorithm to guarantee the validity of polycube constructions. However, their algorithm may incur higher time complexity due to its reliance on local and greedy search strategies.

Our approach leverages label optimization, enhanced by a heuristic algorithm for efficiency and robustness. Recent work by [17] utilized a genetic algorithm(GA) for label refinement, benefiting from its natural selection and crossover processes to swiftly address local label inaccuracies. We enhanced the GA by integrating a sorting mechanism inspired by the immune algorithm. This improved version, referred to as the Immune Genetic Algorithm (I-GA), exhibits an increased capability to consider both the global and local properties of solutions.

Our primary contributions are as follows:

1) We provide a more comprehensive set of valid

Copyright © 2024 by SIAM
Unauthorized reproduction of this article is prohibited

topology conditions for the polycube-map, referred to as the "Validity-Enhanced topology."

2) Leveraging the validity judgments derived from the forenamed theorems, we expand the solvable space of polycube in practical model solving by allowing for 4- and 6-connected graphs and neighbouring charts to be assigned opposite labels.

3) The I-GA algorithm is introduced to enhance the overall and local properties of solutions, improving the efficiency of solution space exploration.

2 Related Work.

Generating high-quality hexahedral meshes is a core problem in the field of mesh generation. A robust approach is to employ polycube mapping [18] for hexahedral mesh generation. This method establishes a low-distortion mapping between the original mesh and a regular polycube mesh, allowing for the transformation of regular polycube hexahedra onto the original mesh.

Deformation-Based Methods. Polycube deformation, based on object tetrahedralization, aims to minimize an energy term penalizing misaligned surface normals with base axes. Early methods by [14] lacked effective penalization of distortions, degeneracies, and flipped elements. In [19], a l_1-norm energy and a variational method were introduced for polycube mesh generation. Despite its effectiveness, this method has a non-linear system that's sensitive to mesh orientation, necessitating extra energy for global orientation optimization and a post-processing step to handle topological degeneracies. In the quest for superior volumetric deformation, Fu introduced the 'normal-driven volumetric deformation and mesh segmentation' algorithm [15], which incorporates a deformation energy integrating the AMIPS term [20]. This energy tends towards infinity when degenerate or inverted elements are present. Similar flip-preventing energies have been proposed in recent studies [21, 22], showing potential in this context. Nonetheless, it's important to note that [23] highlighted the non-local nature of degenerate conditions, making it a challenge to compute a globally valid structure.

Label optimization-based polycube. Many different labeling approaches have been explored in the field of polycube generation. These methods range from purely local strategies, where each surface element is assigned the axis closest to its normal [14], to context-aware techniques, such as employing a modified centroidal Voronoi tessellation in the normal space [24], and incremental procedures [25]. The common objective is to deform the input mesh volumetrically, aligning each surface element with its assigned label. In the process of label optimization, local conditions exist for verifying whether a labeling's graph corresponds to that of an orthogonal polyhedron [11]. Still, these conditions are neither universally applicable nor entirely sufficient. The suitability of the graph formed by the labeling does not guarantee the suitability of the labeling itself, as noted by [25]. Sufficient conditions tend to be of a global nature, as exemplified by [23], who describes a complex post-processing procedure to adapt labeling for polycube suitability. Interactive tools for user-assisted polycube construction or modification have also been developed [6, 7, 26].

Polycube validity topology. The validity of polycube topology is fundamentally anchored in Eppstein's definition of simple orthogonal polyhedra [11]. The authors of PolyCut [16] have laid down pivotal topological conditions for ensuring the validity of polycubes. However, it's worth noting that this method does not guarantee convergence to a valid polycube structure. In instances of invalid polycube structures, the [23] method strategically introduces additional corners, guided by normal constraints, to establish a topologically sound global structure, thereby rectifying it into a valid form. [15] contend that the validity of polycube structures can be determined through a straightforward mesh segmentation algorithm. Nevertheless, their definition primarily hinges on chart connectivity, offering criteria that are sufficient rather than necessary for polycubes. Furthermore, [12] extended the theorems of Steinitz and Eppstein to encompass a broader category of polyhedral complexes, encompassing non-simply connected faces with any genus. They also introduced a necessary condition for the skeleton graph of polycube polyhedra. However, their adherence to 3-connected structures persists, and they address topological adjustments for 4-connected vertex polycubes by programming them as 3-connected. Therefore, the quest for necessary conditions for polycubes remains of paramount importance. In [23], the authors emphasized the coexistence of both local and global conditions with respect to structural validity. This coexistence potentially renders it challenging to completely eradicate invalid cases.

Approaching from a differential geometry standpoint, we present the essential and comprehensive criterion for a valid polycube structure. This is achieved through an in-depth examination of the global and local connectivity of singularities within polycube quad meshes. Our validity topology is rooted in the Discrete Gauss-Bonnet theorem, applied to both global and local contexts. This framework accommodates both 4- and 6-connectivity and even allows adjacent charts to bear opposing labels.

Copyright © 2024 by SIAM
Unauthorized reproduction of this article is prohibited

3 Validity Topology

In the previous section, we outlined the definitions of polycube validity in existing research [23, 16, 15, 27, 25]. We now summarize the most ideal validity proxies identified in these studies as follows:

1. The connectivity of each corner of the polycube is 3-connected.
2. The labels of each adjacent chart must not be opposite, such as +Y and -Y.
3. The number of adjacent charts for each chart should be a minimum of four, and the count of neighbouring charts must be an even number.

We conducted research on polycube polyhedra to establish a more rigorous set of validity conditions. The polycube polyhedron, as defined by [12], requires each vertex to have precisely three axis-parallel edges that are mutually perpendicular. However, to encompass a wider range of polycube polyhedra, including those permitting 4- and 6-connectivity, we have formulated the following definitions.

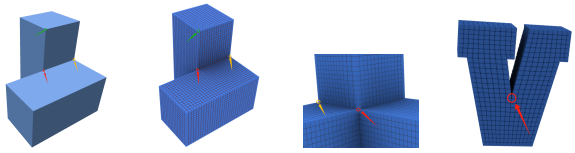
Definition 3.1. Polycube Polyhedron*: A polycube polyhedron is a three-dimensional polyhedron (not necessarily convex) in which every edge at each vertex is parallel to one of the main axes. It allows for situations with 3-, 4-, and 6-connected vertices. In the case of 3- and 4-connected vertices, edges can be either perpendicular or parallel to each other. In the case of 6-connected vertices, six edges are along the $\pm xyz$ axes.*

To investigate the properties of all corners of a polycube polyhedron, we will discuss corners in the context of a quad mesh. This is because there is a close connection between quadrilateral meshes and polycubes [28]. Moreover, Lei et al.'s recent research on the Abel-Jacobi conditions for the singularities of quad-meshes establishes the theoretical equivalence between quad-meshes and meromorphic quartic differentials. In their work [29], it is pointed out that the configuration of singular vertices in the quad-mesh directly corresponds to the arrangement of poles and zeros (divisor) in the meromorphic differential. Historically, there was a common belief that generating cross fields over the surface domain implied equivalence with quad-meshes. However, Lei et al. [30], focusing on singular points, debunked this notion by introducing theorems that precisely delineate singularity configurations for both cross fields and quad-meshes. These collective studies underscore the significance of singular point analysis in quad-mesh construction.

A regular quad mesh generated on a polycube polyhedron is referred to as a "polycube quad mesh," where each quadrilateral represents a standard unit square. Our observations of the polycube quad mesh reveal that each singularity is situated at the corners of the polycube polyhedron, as illustrated in fig. 2.

The polycube quad mesh exhibits singularities exclusively at its corners, comprising 3-, 5-, and 6-singularities. To effectively utilize the information about singular points in the study of polycube polyhedron, we apply the concept of singular points from the polycube quad mesh to the corresponding polycube polyhedron. We classify the corners based on singularity properties and provide the following definition.

Definition 3.2. V_i Point in the Polycube Polyhedron*: The corners of the polycube polyhedron can be categorized as V_3, V_5, and V_6 points, corresponding to 3-, 5-, and 6-singularities in the polycube quad mesh, respectively. Among these, both the V_3 point and V_5 point are 3-connected vertices. The V_6 point encompasses 3-, 4-, and 6-connected vertices.*



(a) (b) (c) (d)

Figure 2: The relationship between vertex connectivity and V_i. (a) Represents a polycube polyhedron, (b) Depicts the corresponding polycube quad mesh. Among these, the green-marked vertex in (b) belongs to V_3, indicating 3-connected. Additionally, the yellow-marked vertices in both (b) and (c) belong to V_5, signifying 3-connected vertices. Moreover, the vertices highlighted in red in (b), (c), and (d) are designated as V_6. In (b), the red vertex displays a 4-connected structure, while in (c), the corresponding red vertex exhibits 6-connectedness. Moving on to (d), the red-marked vertex, also identified as V_6 (with six quadrilaterals in its vicinity), demonstrates 3-connectedness.

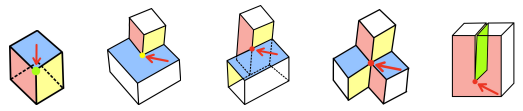
In the case of a polycube polyhedron, every edge on each chart is horizontally or vertically aligned. In the context of a 2D plane or local tangent plane, each edge is parallel to the x or y-axis, forming an orthogonal polygon. Therefore, for each vertex within each chart, we have the following definition:

Definition 3.3. T_i Point in the Chart*: The corners in the chart can be classified as T_i points, where $i \in$*

Copyright © 2024 by SIAM
Unauthorized reproduction of this article is prohibited

$\{1, 2, 3, 4\}$. *This classification is related to the angle θ formed by the two edges at the corner within the chart. The relationship between i and θ in T_i is given by:* $\theta = i \cdot \frac{\pi}{2}$.

The angle θ describes, within a specific chart, the angle formed by the two adjacent edges connecting at the corner as they traverse within that chart. The same corner may be defined as a different T_i type in different charts, as illustrated in fig. 3. Moreover, it can be observed that the chart essentially allows the existence of T_1 to T_4 points. We permit the occurrence of T_2 points because, in the case of a 4-connected V_6 singularity, it forms an angle of π within a chart, corresponding to a T_2 point. Additionally, we allow for the presence of T_4 points, as shown in fig. 2 (d)(V model). During the polycube mapping, they can be mapped as T_4 points when two edges overlap in the parameter domain.



(a) (b) (c) (d) (e)

Figure 3: The relationship between vertex connectivity, V_i, and T_i. (a) The green-marked vertex is 3-connected, belonging to V_3. Additionally, it is a T_1 point in any given chart. (b) The yellow-marked point is 3-connected, belonging to V_5. It is a T_1 point in both the yellow and red charts but a T_3 point in the blue chart. (c) The red point is 4-connected, belonging to V_6. In the blue chart, it is a T_1 point, but in the yellow and red charts, it is a T_2 point. (d) The red-marked point is 6-connected, belonging to V_6. It is a T_1 point in any given chart. (e) The red point is 3-connected, belonging to V_6. In the red chart, it is a T_4 point, and in the green chart, it is a T_1 point.

Definition 3.4. Boundary Loop in the Chart*: The edges of the polycube polyhedron connect to form individual closed loops. A closed loop on a chart is defined as the Boundary Loop of that chart.*

Outer Boundary Loop :*The outer boundary loop represents the outermost edge of the entire chart, separating it from the external environment.*

Inner Boundary Loop :*The inner boundary loop is situated within the chart and specifically demarcates the hole from the main body of the chart.*

Hole :*A hole is a void or empty space within the chart that is not considered part of the main body of the chart, and it is connected to other charts.*

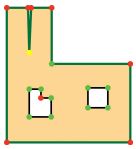


Figure 4: Outer/inner boundary loop. For the orange chart, the green loop is the outer boundary loop, and the black loop is the inner boundary loop. The region enclosed by the black loop with respect to this chart is considered a hole.

For a given chart, the corresponding Outer and Inner Boundary Loops can be observed in fig. 4.

3.1 Polycube Validity Topology Condition For a polycube polyhedron that satisfies the validity conditions, it possesses V3, V5, and V6 singularities. Globally, it must satisfy the following quantity relationship, which we refer to as the polycube validity condition:

Polycube validity topology condition 1:

$$n(V_3) - n(V_5) - 2 \cdot n(V_6) = 8 - 8g; \quad V_3 \geq 8, \text{ and } V_3 \neq 9 \tag{3.1}$$

Where $n(V_i)$ represents the total count of singularities V_i in the polycube and g is the genus.

Proof. Let M be a compact, smooth, and bounded surface with Gaussian curvature K and Euler characteristic χ. According to the Gauss-Bonnet theorem, it holds that:

$$\int_M K dA + \int_{\partial M} k_g ds = 2\pi\chi(M) \tag{3.2}$$

Where dA represents the element of area of the surface, k_g is the geodesic curvature on the boundary, and ds is the line element along the boundary of M. For a compact surface (a 2-dimensional manifold), only vertices, edges, and faces are involved, so the Euler-Poincaré formula becomes:

$$\chi(M) = 2 - 2g = V - E + F \tag{3.3}$$

where V, E, and F are the numbers of vertices, edges, and faces in the polycube quad mesh, respectively.

At singularities V_i, the Gaussian curvature is

$$K(V_i) = (4 - i) \cdot \frac{\pi}{2} \tag{3.4}$$

When considering a polycube polyhedron generated by a polycube-map, we can treat it as a closed discrete mesh. Therefore, the boundary term on the left-hand side of the equation $\int_{\partial M} k_g, ds = 0$. Moreover,

Copyright © 2024 by SIAM
Unauthorized reproduction of this article is prohibited

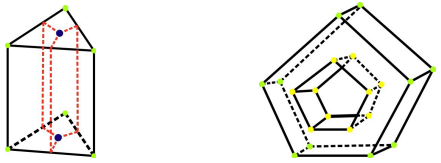
since V_i only has 3-, 4-, and 6-singularities, we have $\int_M K, dA = \sum_{i=3,5,6} n(V_i) \cdot K(V_i)$. Furthermore, combining eq. (3.2), eq. (3.4) and eq. (3.3), we obtain:

$$\sum_{i=3,5,6} n(V_i) \cdot K(V_i) = 2\pi \cdot (2 - 2g) \tag{3.5}$$

Substituting eq. (3.4) into eq. (3.5), we have $n(V_3) - n(V_5) - 2 \cdot n(V_6) = 8 - 8g$

□

Condition 1 is a necessary condition for polycube validity, but it is far from being sufficiently strong. As shown in fig. 5 (a), after generating a quadrilateral mesh for the Triangular Prism, there are 8 V_3 points (6 green points and 2 black points). These points satisfy Condition 1 but do not form a valid polycube map. Therefore, we further propose the following validity conditions:



(a) Triangular prism (b) Hollow pentagon prism

Figure 5: There exist cases where the validity condition 1~3 is satisfied but which are not polycubes

Polycube validity topology condition 2:

Every singularity must reside on a boundary loop and cannot exist in isolation.

Proof: Validity condition 2 is primarily based on geometric observations and induction.

For polycube quadrilaterals, each chart's interior consists of structured full quadrilateral meshes with no singularities inside and the singularities are distributed at the corners. As Definition 1, each corner is either a V3, V5, or V6 singularity. Thus, there is a bijective mapping relationship between singularities and corners. We observe that for a polycube polyhedron, its singularities must be distributed in a closed loop. This condition does not allow singularities to exist in isolation within any chart, as exemplified by the triangular prism in the above figure.

However, situations may still arise that satisfy only conditions 1 2 but do not form a polycube polyhedron. As illustrated in fig. 5 (b), where $n(V_3) = 10$, $n(V_5) = 10$, and $g = 1$, it satisfies eq. (3.1). Additionally, each singular point lies on the boundary loop. Nevertheless, it does not constitute a valid polycube map. This discrepancy arises because the Gauss–Bonnet theorem considers properties from a global perspective. As explained in [23], polycube validity necessitates considering both global and local properties.

In the realm of differential manifolds, each local region serves as a tangent space equipped with charts and transition functions. Notably, charts labeled with coordinates can also be interpreted as those in differential geometry, providing comprehensive coverage of the entire polycube. Consequently, our approach involves a detailed analysis of each individual chart to capture the local characteristics of the polycube.

The validity condition requires that a polycube, in each of its chart's inner and outer bounding loops, must satisfy a specific quantitative relation between its T_i points. This condition is referred to as Validity Condition 3 and can be described as follows:

Polycube validity topology condition 3: For a given chart, any of its boundary loops must satisfy the following condition:

$$n(T_1) - 0 \cdot n(T_2) - n(T_3) - 2 \cdot n(T_4) = 4 \cdot \delta \tag{3.6}$$

Where $n(T_i)$ represents the count of points belonging to T_i in a boundary loop. When the loop is an inner boundary loop, $\delta = -1$, and when it's an outer boundary loop, $\delta = 1$.

Proof. We first consider all boundaries together. For a chart with a boundary count of b and genus 0, its Euler characteristic can be expressed as $\chi = 2 - 2g - b = 2 - b$. Therefore, for a compact, smooth, two-dimensional surface with a boundary, the Gauss-Bonnet theorem can be stated as follows:

$$\int_M K \, dA + \int_{\partial M} k_g \, ds = 2\pi(2 - b) \tag{3.7}$$

For each chart, since it is a flat surface, the Gaussian curvature K at all points is 0. The boundary ∂M of chart being piecewise smooth, we interpret the integral $\int_{\partial M} k_g \, ds$ as the sum of the corresponding integrals along the smooth portions of the boundary, plus the sum of the angles by which the smooth portions turn at the corners T_i of the boundary. In this case, we have:

$$k_g(T_i) = (2 - i) \cdot \frac{\pi}{2} \tag{3.8}$$

Therefore, we can derive:

$$\sum_{i=1}^{4} n(T_i) \cdot k_g(T_i) = 2\pi(2 - b) \tag{3.9}$$

Substituting eq. (3.8), we can obtain:

$$n(T_1) - 0 \cdot n(T_2) - n(T_3) - 2 \cdot n(T_4) = 8 - 4b \tag{3.10}$$

Copyright © 2024 by SIAM
Unauthorized reproduction of this article is prohibited

It is important to note why we do not directly use eq. (3.10) as the polycube validity condition. The reason is that it is a necessary but not a sufficient condition for a polycube polyhedron. In other words, there exist cases that satisfy eq. (3.10) but do not form a valid polycube polyhedron. For example, consider the shape of a pentagon with a bolt-like structure, as shown in fig. 5 (b). If we set the points on its outer boundary as T_1 points and the points on the inner boundary as T_3 points, then $T_1 = T_3 = 5$, which satisfies eq. (3.10). However, it is evident that it cannot constitute a valid polycube polyhedron. Therefore, we further expand eq. (3.10) and consider each boundary separately.

Let's consider a chart with 1 outer boundary and k inner boundaries, and we proceed with the induction as follows:

- When $k = 0$, we have $b = 1$. This leads to the equation $n(T_1) - 0 \cdot n(T_2) - n(T_3) - 2 \cdot n(T_4) = 4$. (Equation 1)
- For $k = 1$, with $b = 2$, we have $n(T_1) - 0 \cdot n(T_2) - n(T_3) - 2 \cdot n(T_4) = 0$. (Equation 2)
  Subtracting Equation 1 from Equation 2, we obtain that the inner boundary i satisfies: $n(T_1) - 0 \cdot n(T_2) - n(T_3) - 2 \cdot n(T_4) = -4$.
- Assuming that for $k = i$, every inner boundary loop satisfies $n(T_1) - 0 \cdot n(T_2) - n(T_3) - 2 \cdot n(T_4) = -4$ (Equation 3);
  Then, for $k = i + 1$, by applying eq. (3.7) to Equation 3, we find that the $i+1$th inner boundary loop satisfies: $T_1 - 0 \cdot T_2 - T_3 - 2 \cdot T_4 = -4$.

□

Based on the four conditions provided above, we present the necessity theorem for the topological conditions of polycube validity as follows:

THEOREM 3.1. *A valid polycube polyhedron must satisfy the following conditions:*

1) For any genus of a polycube polyhedron, the global singularities must satisfy $n(V_3) - n(V_5) - 2 \cdot n(V_6) = 8 - 8g; V_3 \geq 8, and V_3 \neq 9$, *where g represents the genus, and* $n(V_i)$ *represents the number of singularities with valence i.*

2) Every singularity must be on a closed boundary loop and cannot exist in isolation.

3) For a given chart, any of its boundary loops must satisfy: $n(T_1) - 0 \cdot n(T_2) - n(T_3) - 2 \cdot n(T_4) = 4 \cdot \delta$.

Based on the above theorems, for any segmentation result, we can quickly determine whether its segmentation meets the polycube conditions. For regions that do not meet the polycube conditions, we can further utilize the validity formula in label optimization algorithms to correct erroneous labels, as described in the referenced material section 4.

We find that our validity conditions for polycubes can encompass existing conditions outlined in previous research papers such as [12, 15, 16, 19, 25]. However, our conditions, especially Conditions 1 and 3, offer a more precise and formalized approach to evaluating polycube validity compared to the intuitive judgment methods discussed in other studies. Incorporating these conditions into computer algorithms allows for more accurate assessments than manual comparisons. Moreover, our conditions consider both global and local aspects of mesh validity, helping to prevent instances where local validity might lead to global invalidity [23, 25].

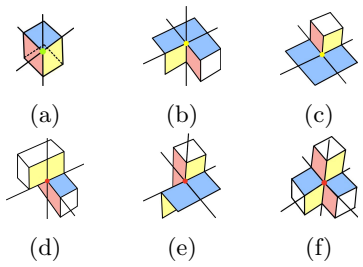


Figure 6: The solvable space of polycube. The conventional method can only solve cases (a) ~ (c), while we are able to solve all situations listed above from (a)~ (f).

3.2 Solvable Space Through our theorem, we are able to make our validity criteria more inclusive. Furthermore, we present in tabular form the range of polycube solution spaces for both our approach and existing methods.

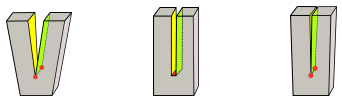
Table 1: Polycube feasible space

	connectivity	singularity	chart angle
Our	3, **4, 6**	V_3,V_5,$\boldsymbol{V_6}$	T_1,$\boldsymbol{T_2}$,T_3,$\boldsymbol{T_4}$
Traditional	3	V_3,V_5	T_1,T_3

As shown in table 1, based on the validity characterization criteria we provided, we allow for both 4-connected and 6-connected cases, whereas traditional methods restrict solutions to the 3-connected case. Regarding singularities, we expand the solvable space to allow for 6 singular points, in contrast to the traditional approach that only permits 3- and 5-singularity solutions. Concerning the turning angles of charts, our validity criteria encompass not only the solutions for T_1 and T_3 but also consider cases where angles can be

Copyright © 2024 by SIAM
Unauthorized reproduction of this article is prohibited

mapped to polycube, such as T_2 and T_4. Finally, our criteria allow for situations where adjacent faces have opposite orientations (e.g., the letter "V" pattern), which would be considered invalid in traditional approaches.



(a) input (b) Tra (c) Our

Figure 7: V model. We can get a better polycube-map by allowing adjacent charts to assign opposite labels

In summary, we achieve a solvable space space through precise mathematical definitions. These definitions are utilized as energy terms in our label optimization algorithm, as described in section 4.

4 Labelling Optimization.

In the label optimization process, we initially need an initial labeling for refinement. Subsequently, we use these initial labels as input to obtain the optimal solution through the immune genetic algorithm (I-GA). Within the I-GA algorithm, we incorporate the proposed validity topology condition to design the validity energy, which serves as a component of the fitness function.

4.1 Initial Solution In order to obtain a suitable initial labeling, we followed the methods from the previous studies [16] and [25]. We use the graph-cut multi-label optimization method presented by [16].

Additionally, we also need to compute turning points on boundary edges. Turning points represent non-monotonic points along the boundary edges. Similar to [16], we employ the graph-cut algorithm to integrate each edge for calculating turning points.

For the initial labeling, it is typically an invalid polycube topology, meaning that this labeling result cannot be mapped to polycube hexahedra. We need to utilize the I-GA algorithm for label optimization.

4.2 Obtaining Information for T_i and V_i For the initial labeling or the label set l_i obtained at each iteration of the I-GA optimization process, we need to acquire information regarding T_i and V_i to calculate our validity energy E_v based on them. To do so, we specifically utilize the following approach for computation.

We outline how our validity criteria are practically implemented in engineering applications. Using the aforementioned energy measures, we can obtain an initial solution for the labeling algorithm. we initially obtain the chart set through a breadth-first search algorithm, followed by the recursive derivation of the corner set.

Concerning corners, we further distinguish between their T_i and V_i attributes, specifying four possible cases for corners: 3, 5, 6 singular points, and error-vertices. In the case of corners with three neighboring faces marked, they may be classified as V_3,V_5, or V_6 points. We determine whether they are V_3 or V_5 points by computing the cross product of the unit normals of the neighboring triangles in a counterclockwise (CCW) direction. The right-hand rule is applied to make this determination. All V_3 points are labeled as T_1 points across all charts. For V_5 points, there are three possible angles in each of the three charts, and we select the largest angle θ as the T_3 point, while the remaining corners are labeled as T_1 points. If the cross product of adjacent principal axes results in zero and the maximum angle θ in the neighborhood is greater than $\frac{3}{2}\pi$, the corner is classified as a V_6 point. Otherwise, it is identified as an error-vertex and requires label repair. In cases where corners have six neighboring labels, they are categorized as V_6 points.

In scenarios where corners have four neighboring labels, we calculate the cross products of adjacent label principal axes in CCW order. If these cross-product results are consistent, the corner is designated as an error vertex; otherwise, it is classified as a V_6 point. Given that V_6 points have four possible T_i points, we select two of them with angles close to π as T_2 points, while the remaining two are labeled as T_1 points.

4.3 I-GA Algorithm We employ an enhanced genetic algorithm inspired by the previous genetic algorithm proposed in [25]. In contrast to [25], which relies solely on probability distributions for modeling, our I-GA algorithm incorporates the sorting mechanism from the immune algorithm. This addition enables us to better consider both the global and local properties of solutions. Consequently, it helps in reducing excessive computations and potential waste of computational resources.

The I-GA algorithm incorporates random selection and crossover of candidate solutions to simultaneously explore multiple search directions. This strategy aims to reduce sensitivity to local minima. Furthermore, the polycube validity topology conditions (presented in section 3) and the fitness function we define establish standards and criteria for the search process.We compute the fitness of the I-GA algorithm as follows:

Firstly, To strike a balance between mapping distortion and singularity counts, we employed the fidelity (E_F) and compactness (E_C) energy from [16] to mini-

Copyright © 2024 by SIAM
Unauthorized reproduction of this article is prohibited

Algorithm 1 I-GA Algorithm

Require: Polycube C
Require: Maximum number of iterations t
Require: $M, N \in \mathrm{N}, M \geq N$
  $X \longleftarrow$ First partition result using the [16] method
  $B \longleftarrow \emptyset$
  for $i = 1$ to t **do**
    $F \longleftarrow$ Use equation (4.11) to compute the population fitness with respect to X.
    $F \longleftarrow Sort(F)$
    $M \longleftarrow$ Selection of the first M individuals
    if B is $\emptyset$ **then**
      $X \longleftarrow$ The set of first N individuals of M
    else
      $M \longleftarrow mutation(M)$
      $N \longleftarrow$ Crossing using sets consisting of M and N
    end if
  end for
  $x^\star \longleftarrow$ One of the best individuals in N fitness.
  return $x^\star$

mize normal and boundary energies, respectively. Secondly, we employ the energy derived from the singular values of the Jacobian of the mapping, as discussed in [25], to represent the per-triangle distortion energy (E_W). Finally, we devised the computation of validity energy (E_v) based on our Validity topological condition, which is expressed as eq. (4.11).

By combining the above metrics, we define a fitness function that can be embedded in our I-GA algorithms framework:

(4.12)
$$fitness(l) = \omega_1 E_v(l) + \omega_2 E_W(l) + \omega_3 E_F(l) + \omega_4 E_C(l)$$

Where, l represents the set of labels assigned to the mesh, signifying the result of the polycube map. The constants ω_1, ω_2, ω_3, and ω_4 are coefficients. This function effectively distinguishes between candidate solutions and promotes search directions, minimizing parameterization distortion, as assessed by our metrics.

After the initial labeling by section 4.1, we employ the following steps for the search:

1) Archiving: We use an archiving system to keep track of the best solutions obtained so far.

2) Crossover: The crossover operator takes two solutions, l1 and l2, as input and generates a new individual that incorporates mutations from both parents.

3) Generations: According to our $fitness(l)$, N candidate solutions are selected from the archive at the beginning of each generation. According to our stochastic model, the top-ranked solutions are selected multiple times.

4) Mutation: Two individuals are randomly selected for mutation.

5) Selection: New solutions with sufficient scores are added to the archive, while unsuitable ones are discarded, terminating the generation.

This enhanced genetic algorithm not only considers multiple search directions but also evaluates global and local properties more accurately through the sorting mechanism inspired by the immune algorithm. The corresponding pseudocode is shown in algorithm 1. By incorporating the sorting mechanism from the immune algorithm, our genetic algorithm can effectively handle multiple search directions and reduce the risk of falling into local minima. The introduction of validity conditions ensures that the solutions obtained are reasonable and feasible. The key to this algorithm lies in the design of the $fitness(l)$, which comprehensively considers multiple aspects of solutions, including validity and optimization criteria.

Our genetic algorithm is an adaptive process that adjusts search directions based on the results of each generation, enabling more effective exploration of the solution space. Additionally, we implement an archiving system to retain the best solutions obtained so far. This approach helps us escape local optima, increasing the likelihood of finding the global optimum. By retaining highly evaluated solutions and introducing a certain level of randomness in each generation, our algorithm maintains diversity during the search process, enabling a more thorough exploration of the solution space.

In the crossover operation, we combine two solutions, generating a new individual. This operation retains the advantages of parental solutions while introducing a certain degree of variation, expanding the breadth of the search space. The introduction of the crossover operation accelerates the convergence speed of the algorithm, allowing for faster discovery of high-quality solutions.

(4.11)
$$E_v = \begin{cases} 1000, & \text{if } n(V_3) < 8 \text{ or } n(V_3) = 9 \\ |n(V_3) - n(V_5) - 2 \cdot n(V_6) - 8 + 8g| + \sum_{i \in \text{boundary loop}} |(n(T_1^i) - n(T_3^i) - 2 \cdot n(T_4^i) - 4 \cdot \delta_i)|, & \text{otherwise} \end{cases}$$

Copyright © 2024 by SIAM
Unauthorized reproduction of this article is prohibited

In each generation of the algorithm, a certain number of candidate solutions are selected from the archive based on the $fitness(l)$. This selection process considers both the quality of solutions and introduces a level of randomness, enhancing the diversity of the algorithm. The selected individuals then undergo random mutations, introducing even more diversity. Finally, new solutions with sufficiently high scores are added to the archive, while unsuitable solutions are discarded, leading to the termination of the generation.

5 Results.

To evaluate the performance of our proposed polycube method, we conducted extensive quantitative and qualitative experiments. Our experimental data mainly originated from the Thingi10k dataset, along with data from the commonly used ABC dataset in state-of-the-art methods. We selected data with topological structures similar to those shown in fig. 1, specifically from (b) to (d), which contain 6 singularities. We ensured that this dataset covered various levels of complexity, diversity, and genus numbers.

The experimental data were categorized into two types: artistic and industrial, to reflect different usage scenarios. We compared our algorithm against other state-of-the-art approaches. Our algorithm takes triangular meshes as input, and by comparing polycube maps

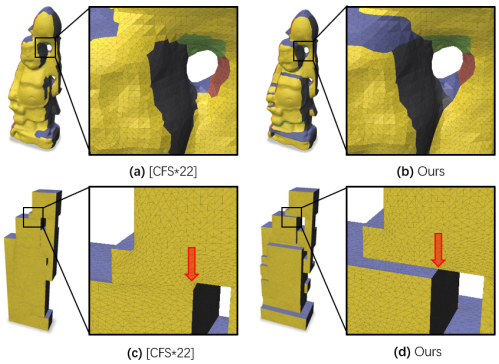


Figure 9: The vertices indicated by arrows in the figure represent 4-connected graphs with degree of 6. (a) and (b) depict a comparison of the initial mesh labels between [CFS*22][25] and our method, while (c) and (d) show a comparison of the generated polycube map, with the details magnified in the black boxes. Due to the expanded solvable space enabled by our approach, structures similar to the one highlighted in (d) can emerge, resulting in improved polycube results. In the generated all-hex mesh, our average weighted Jacobian is 0.873, surpassing the 0.775 achieved by the method in [CFS*22][25].

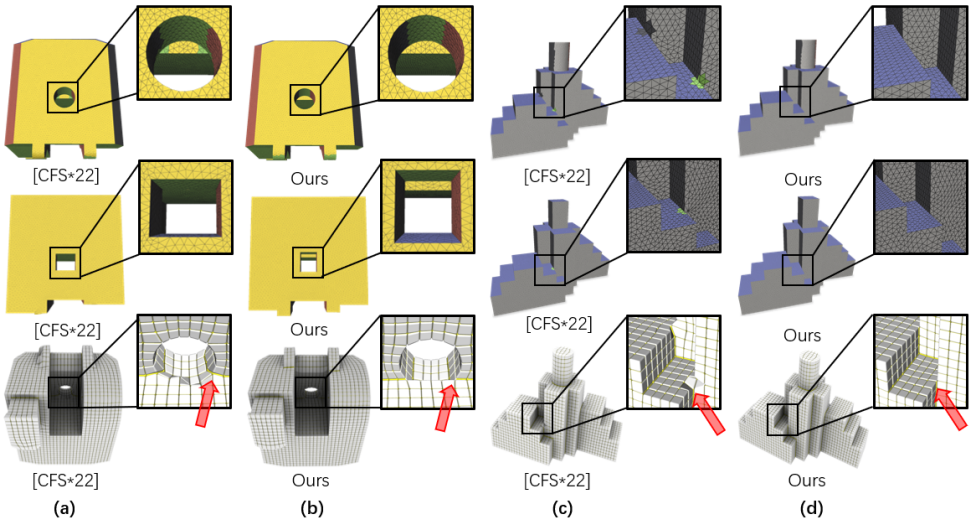


Figure 8: The vertices indicated by arrows from (a) to (d) contain 4-connected graphs with degree of 6. (a) and (c) were generated using the method from [CFS*22][25], while (b) and (d) represent our results. The first row shows the initial triangular mesh with labels, followed by the polycube maps and the all-hex meshes. It can be observed that the method from [CFS*22][25] struggles to handle this structure and tends to treat such points as singular points with a degree of 5. This leads to errors in the labels on the initial triangular mesh and worse corresponding polycube map, ultimately resulting in significant distortions in the generated all-hex mesh.

Copyright © 2024 by SIAM
Unauthorized reproduction of this article is prohibited

and the final hexahedral mesh results, we demonstrated the superiority of our proposed method.

5.1 Polycube Labeling The generation of polycube maps typically involves labeling the initial triangular mesh with seamless textures of six different colors, corresponding to the six principal axis directions. This process partitions the surface and establishes a topological structure that conforms to the polycube conditions. For meshes that are polyhedra and meet the criteria for polycubes but do not adhere to the traditional definition, as shown in fig. 1, scholars typically resort to forcibly altering the principal axis assignments of the non-compliant local triangles to make them conform to the criteria. However, such forced corrections inevitably introduce mapping distortions, resulting in suboptimal quality in the generated all-hex meshes.

It can be observed that vertices similar to the one indicated by the arrow in fig. 8 (a) cannot be mapped to 6-singularities by other state-of-the-art methods. This is because traditional validity definitions enforce a requirement of 3-connectivity, making it impossible for traditional methods to solve this, or forcing them to handle such vertices as singularities with degrees of 3 or 5 through label optimization, resulting in significant distortion. Due to the expansion of the solvable space by our method, we can handle such situations, as shown by the arrow in fig. 8 (b), and our approach yields the optimal results.

Fig. 9 presents the initial triangular mesh with labels for the famous 'monk' mesh, a typical representative of complex artworks, along with the polycube maps. We conducted a comparative analysis with the methods proposed in [25]. In our polycube maps, we allow structures similar to those shown in fig. 1 (b) to emerge, thereby expanding the solvable space of the polycube method. Notably, for the structure depicted in fig. 1 (b), the method in [16] encounters difficulties during generation and experiences issues during the deformation stages. The mesh-containing structure in fig. 1 (c) poses significant challenges for the [16] approach, and although the fig. 1 (d) structure can be generated successfully, it exhibits noticeable label inaccuracies. In contrast, the method in [25] encounters label inaccuracies in all three cases, adversely affecting the quality of the resulting all-hex meshes.

5.2 All-hexahedra Mesh Generation In light of the considerable potential that polycubes offer within the realm of hexahedral mesh generation tasks, we proceeded to undertake a thorough evaluation to assess the effectiveness of our approach in this context.

[0]The additional quantitative evaluation results can be found in Table 2, provided in the supplementary materials.

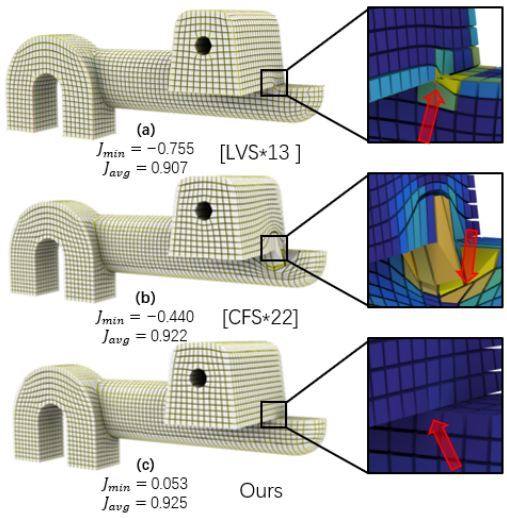


Figure 10: A comparison on complex industrial components. The area pointed to by the arrow represents a 4-connected graph with a degree of 6. In the figure, the closer the color is to blue, the higher the weighted Jacobian value, indicating superior hexahedral mesh quality. Our method yields the best results.

In fig. 10 (c), the results of our algorithm applied to industrial parts are displayed. In contrast, the other two methods, (a)[16] and (b)[25], cannot handle singular points with a degree of 6. Both methods treat these points as degree-5 vertices, resulting in significant distortion within the black-bordered region. Our method achieves excellent results in handling the structure within the black-bordered area, avoiding distortion and yielding a weighted Jacobian value close to 1.

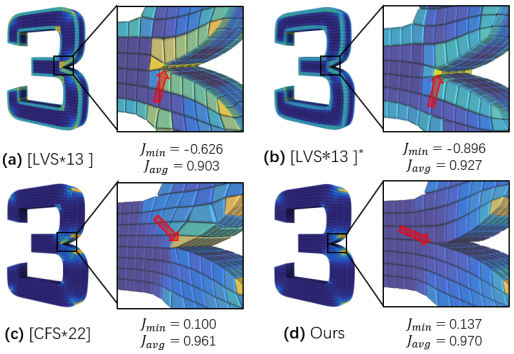


Figure 11: The arrow points to a 3-connected vertex with a degree of 6. In the figure, closer to blue indicates higher weighted Jacobian values. * represents the result after iterative optimization

Copyright © 2024 by SIAM
Unauthorized reproduction of this article is prohibited

Figure 12: Gallery of our all-hex meshes and polycube maps.

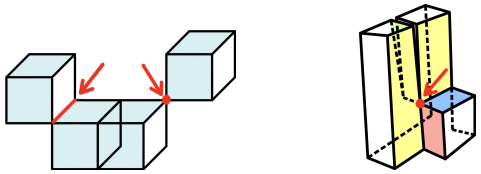
Furthermore, fig. 11 shows a comparison on artworks where our approach consistently delivers top-quality results. with [16] and [25] used as the reference group. * denotes the result of the method after complex post-processing steps. Although it achieves higher weighted Jacobian values at certain Curved surface, it deviates from the surface of the input triangular mesh. The hexahedral mesh generated after this step has a surface that differs from the original triangular mesh. Under the condition of the fewest hexahedra and without post-processing, our method achieved the best results.

Fig.12 provides additional polycube-maps and all-hex meshing results, along with the names of the meshes. In this figure, we have highlighted degree-6 3-connected, 4-connected, and 6-connected graphs using yellow boxes.

6 Conclusion and Future Work.

We have introduced a validity-enhanced approach to expanding the solvable space of polycube maps. The four validity topology conditions we propose support the mapping of 3, 5, and 6 singularities. When integrated into the label optimization algorithm of the I-GA method, it can detect and rectify invalid label regions. Our method effectively handles models with 3-, 4-, and 6-connected components and extensive experiments have been conducted on various models. The resulting polycubes exhibit meaningful complexity and yield high-quality hex meshes across a diverse range of input geometries. This observation aligns with the findings in our own research.

Our method has a few limitations. As shown in fig. 13, our approach cannot handle non-manifold meshes like (a). Additionally, in a broader sense, polycube maps may also encompass vertex singularities with a valence of 7. While such models are rare, our validity topology does not account for this scenario, which we will address in our future work.



(a) Non-manifold (b) 7 Singularitie

Figure 13: Scenarios Beyond the Capability of Validity-Enhanced Construction

Acknowledgements

This work is supported by National Natural Science Foundation of China No. 62272080, LiaoNing Revitalization Talents Program of China No. 2022RG04 and National Science Fund for Distinguished Young Scholars of China No. T2225012.

Copyright © 2024 by SIAM
Unauthorized reproduction of this article is prohibited

References

[1] C.-C. Chang and C.-Y. Lin. Texture tiling on 3d models using automatic polycube-maps and wang tiles. *Journal of Information Science & Engineering*, 26(1), 2010.

[2] Y. Yang, X.-M. Fu, and L. Liu. Computing surface polycube-maps by constrained voxelization. In *Computer Graphics Forum*, volume 38, pages 299–309. Wiley Online Library, 2019.

[3] C. Yuksel, S. Lefebvre, and M. Tarini. Rethinking texture mapping. In *Computer Graphics Forum*, volume 38, pages 535–551. Wiley Online Library, 2019.

[4] X. Fang, W. Xu, H. Bao, and J. Huang. All-hex meshing using closed-form induced polycube. *ACM Transactions on Graphics (TOG)*, 35(4):1–9, 2016.

[5] L. Chen, G. Xu, S. Wang, Z. Shi, and J. Huang. Constructing volumetric parameterization based on directed graph simplification of 1 polycube structure from complex shapes. *Computer Methods in Applied Mechanics and Engineering*, 351:422–440, 2019.

[6] Y. Yu, J. G. Liu, and Y. J. Zhang. Hexdom: Polycube-based hexahedral-dominant mesh generation. In *Mesh Generation and Adaptation: Cutting-Edge Techniques*, pages 137–155. Springer, 2022.

[7] Y. Yu, X. Wei, A. Li, J. G. Liu, J. He, and Y. J. Zhang. Hexgen and hex2spline: Polycube-based hexahedral mesh generation and spline modeling for isogeometric analysis applications in ls-dyna. In *Geometric Challenges in Isogeometric Analysis*, pages 333–363. Springer, 2022.

[8] Q.-C. Guan, H. Liu, Y.-C. Wang, and X.-M. Fu. Error-bounded unstructured t-spline surface fitting with low distortion. *Journal of Graphics*, 43(6):1104, 2022.

[9] G. M. Ziegler. *Lectures on Polytopes*, volume 152. Springer Science & Business Media, 2012.

[10] B. Grünbaum, V. Klee, M. A. Perles, and G. C. Shephard. *Convex Polytopes*, volume 16. Springer, 1967.

[11] D. Eppstein and E. Mumford. Steinitz theorems for orthogonal polyhedra. In *Twenty-sixth Symposium on Computational Geometry*, 2009.

[12] H. Zhao, X. Li, W. Wang, X. Wang, and X. Gu. Polycube shape space. *Computer Graphics Forum*, 38(7):311–322, 2019.

[13] N. Pietroni, M. Campen, A. Sheffer, G. Cherchi, D. Bommes, X. Gao, R. Scateni, F. Ledoux, J. Remacle, and M. Livesu. Hex-mesh generation and processing: A survey. *ACM Transactions on Graphics*, 42(2):1–44, 2022.

[14] J. Gregson, A. Sheffer, and E. Zhang. All-hex mesh generation via volumetric polycube deformation. *Computer Graphics Forum*, 2011.

[15] C.-Y. Bai, Y. Liu, and X.-M. Fu. Efficient volumetric polycube-map construction. *Computer Graphics Forum: Journal of the European Association for Computer Graphics*, 2016.

[16] M. Livesu, N. Vining, A. Sheffer, J. Gregson, and R. Scateni. Polycut: Monotone graph-cuts for polycube base-complex construction. *ACM Transactions on Graphics (TOG)*, 32(6):1–12, 2013.

[17] F. Protais, M. Reberol, N. Ray, E. Corman, F. Ledoux, and D. Sokolov. Robust quantization for polycube maps. *Computer-Aided Design*, page 150, 2022.

[18] M. Tarini, K. Hormann, P. Cignoni, and C. Montani. Polycube-maps. 2004.

[19] J. Huang, T. Jiang, Z. Shi, Y. Tong, H. Bao, and M. Desbrun. l1-based construction of polycube maps from complex shapes. *ACM Transactions on Graphics (TOG)*, 33(3):1–11, 2014.

[20] X.-M. Fu, Y. Liu, and B. Guo. Computing locally injective mappings by advanced mips. *ACM Transactions on Graphics (TOG)*, 34(4):1–12, 2015.

[21] X.-M. Fu and Y. Liu. Computing inversion-free mappings by simplex assembly. *ACM Transactions on Graphics (TOG)*, 35(6):1–12, 2016.

[22] M. Rabinovich, R. Poranne, D. Panozzo, and O. Sorkine-Hornung. Scalable locally injective mappings. *ACM Transactions on Graphics (TOG)*, 36(4):1, 2017.

[23] D. Sokolov and N. Ray. *Fixing Normal Constraints for Generation of Polycubes*. PhD thesis, LORIA, 2015.

[24] K. Hu and Y. J. Zhang. Centroidal voronoi tessellation based polycube construction for adaptive all-hexahedral mesh generation. *Computer Methods in Applied Mechanics and Engineering*, 305:405–421, 2016.

[25] C. Dumery, F. Protais, S. Mestrallet, C. Bourcier, and F. Ledoux. Evocube: A genetic labeling framework for polycube-maps. 2022.

[26] L. Li, P. Zhang, D. Smirnov, S. M. Abulnaga, and J. Solomon. Interactive all-hex meshing via cuboid decomposition. *Association for Computing Machinery (ACM)*, 40(6):1–17, 2021.

[27] H. Zhao, N. Lei, X. Li, P. Zeng, K. Xu, and X. Gu. Robust edge-preserving surface mesh polycube deformation. *Computational Visual Media*, 4(1):1–10, 2018.

[28] M. Baumeister and L. Kobbelt. How close is a quad mesh to a polycube? *Computational Geometry*, 111:101978, 2023.

[29] N. Lei, Zheng X., Z. Luo, F. Luo, and X Gu. Quadrilateral mesh generation ii: Meromorphic quartic differentials and abel-jacobi condition. *Computer Methods in Applied Mechanics and Engineering*, 2020.

[30] X. Gu, N. Lei, Y. Zhu, X. Zheng, H. Si, and Z. Luo. Why cross fields not equivalent to quadrilateral meshes. *Available at SSRN 4460747*.

Copyright © 2024 by SIAM
Unauthorized reproduction of this article is prohibited