LIMITS AND PROSPECTS OF POLYCUBE LABELINGS

Sébastien Mestrallet[1,4] François Protais[2,3] Christophe Bourcier[1] Franck Ledoux[4,5]

[1] *Université Paris-Saclay, CEA, Service de Génie Logiciel pour la Simulation, 91191, Gif-sur-Yvette, France.*
[2] *Université de Lorraine, CNRS, Inria, LORIA.*
[3] *Siemens Digital Industries Software.*
[4] *Université Paris-Saclay, CEA, Laboratoire en Informatique Haute Performance pour le Calcul et la simulation, 91680, Bruyères le Chatel, France.*
[5] *CEA, DAM, DIF, F-91297 Arpajon, France.*

ABSTRACT

Polycubes have been a fruitful approach for all-hexahedral mesh generation, thanks to an attractive robustness/quality trade-off. Starting from a tetrahedral mesh of the input shape, a polycube can be easily represented by associating one of the six signed orthogonal direction $\pm\{X, Y, Z\}$ to boundary triangles, called labeling. Not all labelings induce polycubes, therefore validity criteria have been proposed. Despite satisfactory in most cases, they are neither necessary nor sufficient. By presenting failure cases, we open the discussion towards new approaches to discriminate between valid and invalid polycube labelings.

Keywords: polycube, hexahedral meshing, mesh generation, computational geometry

1. INTRODUCTION

1.1 Polycube-maps

Polycube-maps were introduced by Tarini et al. [1] as an efficient texture storage technique for 3D objects. The polycube domain is a set of unit cubes, and the projection into the polycube domain aims at mapping the 3D surface to a set of square patches, trivially stored in memory.

Polycube-maps prove to be a useful intermediate representation for hexahedral meshing, because an hex-mesh can be robustly extracted from a polycube [2, 3] (figure 1). This way, the complex hex-mesh generation problem can be simplified to a polycube-map optimization, under two main limitations: axis-aligment constraint and singularity-free ouput mesh. Note that recent works proposed new approaches alleviating these obstacles with polycube domain warping or integer-grid maps [4, 5].

1.2 Polycube labeling

The simplest way to construct a polycube-map is with a supporting tetrahedral mesh. The triangular surface mesh is extracted, and each triangle is labeled with one of the six signed directions $\pm\{X, Y, Z\}$. The ordered set of labels is called the **labeling** (figure 1). In the figures below, unsigned directions $\{X, Y, Z\}$ are colored in white, blue, and red respectively. The labeling where each triangle is assigned to the direction nearest to its normal is called **naive labeling** and is often the starting point of labeling optimization algorithms. Not all labelings can produce a valid polycube, this is why several works have been conducted to formulate the necessary and sufficient criteria, or, for want of anything better, criteria that work in most cases [6, 7].

It must be ensured that the input tetrahedral mesh has enough elements to capture small features, like narrow holes, and to provide enough freedom to satisfy validity constraints, especially near acute feature edges (see figure 4 left). The volumetric mesh is not

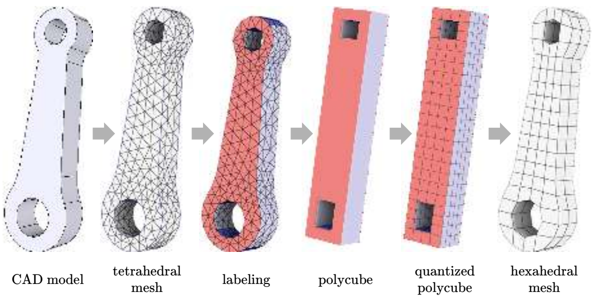


Figure 1: Hex mesh generation pipeline using a polycube and a labeled tetrahedral mesh.

used in current optimization methods (see section 3.5 for implications), but it is required by the hexahedral mesh extraction step.

1.3 Labeling terminology

The labeling can be segmented into **charts**, or patches, by grouping together adjacent triangles of same label. A **boundary**, or border, is a set of all edges between two adjacent charts. Finally, **corners** are defined as vertices where three or more boundaries meet. The number of adjacent boundaries of a given corner is called its **valence**.

2. VALIDITY CRITERIA

2.1 Simple orthogonal polyhedra

Eppstein and Mumford [6] characterized orthogonal polyhedra as graphs, extending Steinitz's theorem. They defined simple orthogonal polyhedra, and 3 criteria to discriminate them (figure 2):

- Charts with opposite direction of the same axis must not share a boundary. Else, it is an **invalid boundary**;
- Charts must have 4 or more neighbors. Else, they are **invalid charts**;
- Corners must have 3 neighbors. Else, they are **high valence corners**.

Indeed, an orthogonal polyhedron is not guaranteed when opposite sides of the cube are made adjacent, when a chart has less sides than a cube's face, and when corners do not represent a concave/convex cube corner. These criteria are sufficient for genus-0 shapes when the labeling is unsigned. That is why figure 6a has a corresponding simple orthogonal polyhedra (see figure 3.2 of [8]) while having no valid polycube when the sign is enforced.

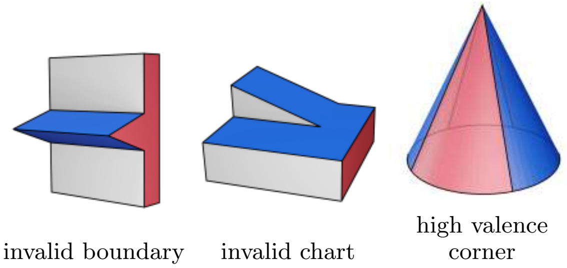


Figure 2: Example labelings that do not lead to simple orthogonal polyhedra. Illustration from [9].

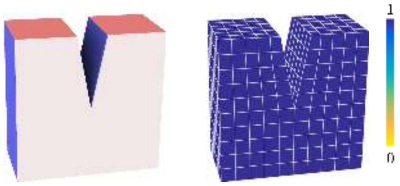


Figure 3: Model inducing an invalid boundary in the naive labeling, and an hex mesh extracted with [2]. Hexahedra are colored by their Scaled Jacobian.

2.2 Usage in heuristics

Although not all polycubes may be simple orthogonal polyhedra, several polycube generation approaches take these criteria into account due to their low computational cost. Two of them are purely local constraints. In *PolyCut* [10], Livesu et al. keep track of the criteria, along with distortion and compactness metrics, in a iterative hill-climbing optimization process with tweaked graph-cut steps. Fu et al [11] proposed a deformation before the labeling, and required solutions to comply with the 3 criteria. They further align the mesh to the axis directions for invalid ones. Guo et al. [12] restricted the topology of polycube shape to these criteria as well. In the *Evocube* genetic framework [9], criteria from Eppstein and Mumford are used in the objective function, as validity proxy.

3. LIMITATIONS

3.1 Labelings wrongly considered invalid

As specified in [6], in some cases these criteria are too strict. Invalid labelings can lead to satisfactory hex meshes. A simple example regarding invalid boundaries is shown in figure 3. However, if the interior angle is small (figure 4 left), the cuboid is flat and no hex mesh can be extracted.

High valence corner can also lead to satisfactory polycubes. The valence-4 case was known since the formulation of Eppstein and Mumford's criteria. In fact,

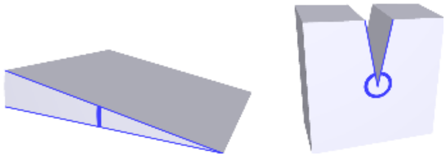


Figure 4: Interior angle higher than 180° (right) could lead to an acceptable boundary between opposite direction, ones lower than 180° (left, from [13]) are always invalid.

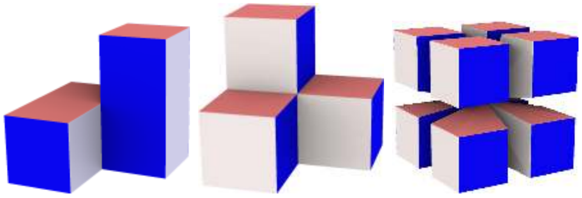


Figure 5: Minimal examples with valence-4 corner (left), valence-6 corner (middle) and valence-24 corners.

some corners of valence of 6 – and up to 24 – can produce valid polycube representations (see figure 5). The extreme case is not an unique piece, and is obtained with "invalid boundaries" meeting in the same corner. Corners of intermediate valence can be obtained with less boundaries of this kind.

3.2 Labelings wrongly considered valid

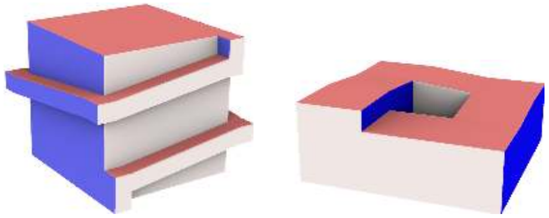
Sokolov and Ray [14] presented a **genus-0** model (figure 6a) where validity criteria are respected (locally valid) despite not allowing for a polycube without further modifications because of the conflicting normal constraints on the z-axis (globally invalid). Therefore, when the labeling is signed, criteria of [6] become insufficient even for genus-0 shapes.

A simple **genus-1** problematic shape is a cuboid torus with a step (figure 6b). The naive labeling is locally valid as well, but as is, the step will be crushed in the iso-z plane in red.

The **genus-2** model in figure 6c has a locally and globally valid labeling that does not capture the twist of one of the holes. The associated polycube cannot generate a satisfactory hex mesh. The grooves on the blue faces prevent hex mesh extraction algorithms from untwisting the holes.

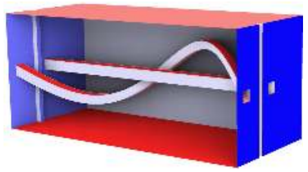
3.3 Requirements for polycube hex-mesh extraction

HexEx [2] uses global parametrizations, so it handles well local subtlety like opposite directions on adjacent charts. On the other hand, [3] has special global constraints in order to improve quantization robustness, requiring overlap-free polycubes (see figure 7).



(a) Labeled genus-0 shape wrongly considered valid

(b) Labeled genus-1 shape wrongly considered valid



(c) Labeled genus-2 shape wrongly considered valid. The front face has been removed.

Figure 6: Cases showing validity criteria are insufficient.

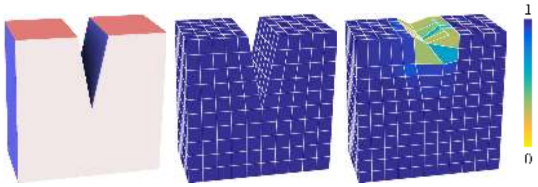


Figure 7: Labeling (left), hex mesh extracted by [2] (center) and [3] (right), where the induced polycube has overlapping faces due to opposite directions made adjacent. Hexahedra are colored by their Scaled Jacobian.

Because tolerance varies with the hex-mesh extractor used on the polycube, the "invalid boundary" case cannot be accepted systematically.

3.4 Appearance in academic and industrial shapes

Failure cases shown are pretty rare: in *Evocube* [9], over the 1309 models tested, only 16 of them outputted an invalid labeling because of overconstrainted validity, and about 8 outputted a valid but unsatisfactory labeling because the optimization algorithm tried to avoid these configurations. Other causes of invalid/unsatisfactory labeling are a too-coarse input mesh and challenging geometry for polycubes (poorly aligned features). Figure 8 showcases prob-

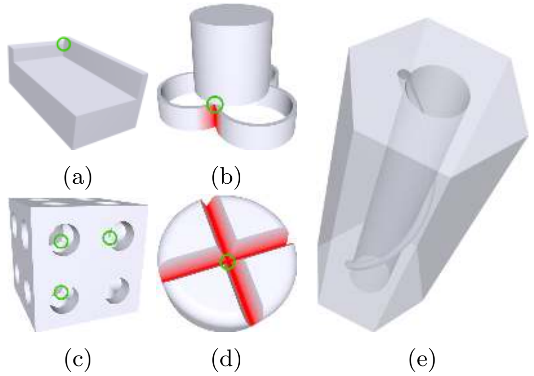


Figure 8: Academic and industrial shapes leading to invalid boundaries (red), high-valence corners (green) or global invalidity: (a) ABC8298 (b) ABC5185 (c) cheese2 (d) simplified sub-part of ABC7165 (e) simplified fuel rod assembly.

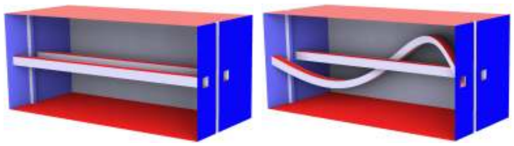


Figure 9: Two labeled shapes having the same labeling graph: (right) figure 6c (left) the same shape with straight holes. The front faces have been removed.

lematic cases from academic [15, 13, 16] and industrial datasets. Fully valid labelings can be achieved on these shapes with additional charts, but it induces artifacts on the hex mesh.

3.5 Limits of boundary information

Figure 9 shows that boundary information will never be enough to discriminate between valid and invalid polycube representations: if the holes are not twisted, the naive labeling gives the same polycube.

Invalidities we encountered can be sorted by easiness of detection: local ones comes from the neighborhood of triangles or charts, global ones from the overall charts connectivity (i.e. when normal constraints are incompatible [14]), and in-volume ones are undetectable without volume awareness. Table 1 summarizes the kind of invalidity that can be encountered according to the genus of the shape.

4. PROSPECTS

4.1 Fixing normal constraints

This limit is considered in [14]: Sokolov and Ray proposed an algorithm editing the meta-mesh representation, allowing a globally valid polycube transformation on challenging models like figure 6a and 6b. This post-processing step is said to be fast, and because volume foldovers such as figure 6c are very rare, it can be a satisfactory follow-up to labeling optimization algorithms [10, 11, 12, 9].

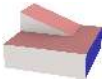
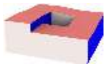
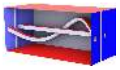
	genus 0	genus 1	genus 2
locally invalid	 Figure 2 middle	possible but not of interest	possible but not of interest
locally valid but globally invalid	 Figure 6a	 Figure 6b	possible but not of interest
globally valid but in-volume invalid	impossible	impossible	 Figure 6c

Table 1: Example models of different genus with different level of invalid labeling.

4.2 Pre-deformation as evaluation

Instead of estimating polycube feasibility from the labeling, a quick surface polycube [17, 18] can be generated, and the induced deformation be used as quality criteria in heuristics. Invalid configurations will involve high distortion and therefore be dismissed. Because the polycube is surface-only, the volume foldovers issue remains unsolved, and a complete volumetric polycube is still needed to extract an hex-mesh at the end.

4.3 Toward validity-aware operators

Going away from iterative processes dealing with validity and quality simultaneously, work could be heading towards the quality optimization only, of valid-only solutions. This implies the generation of valid – but geometrically unsatisfactory – initial polycube(s), as well as the definition of validity-aware operators, improving the quality over time.

5. CONCLUSION

Being a convenient way to represent solutions, polycube labelings gained popularity despite the growing number of counter-examples regarding validity. We pave the way for new approaches to handle validity of this representation, still simple enough to be embedded in evolutionary algorithms.

References

[1] Tarini M., Hormann K., Cignoni P., Montani C. "PolyCube-Maps." *ACM Trans. Graph.*, vol. 23, no. 3, 853–860, aug 2004. URL `https://doi.org/10.1145/1015706.1015810`

[2] Lyon M., Bommes D., Kobbelt L. "HexEx: Robust Hexahedral Mesh Extraction." *ACM Trans. Graph.*, 2016. URL `http://dx.doi.org/10.1145/2897824.2925976`

[3] Protais F., Reberol M., Ray N., Corman E., Ledoux F., Sokolov D. "Robust Quantization for Polycube Maps." *Computer-Aided Design*, vol. Volume 150, no. 103321, Sep. 2022. URL `https://hal.archives-ouvertes.fr/hal-03076711`

[4] Mandad M., Chen R., Bommes D., Campen M. "Intrinsic mixed-integer polycubes for hexahedral meshing." *Computer Aided Geometric Design*, vol. 94, 102078, 2022. URL `https://doi.org/10.1016/j.cagd.2022.102078`

[5] Brückler H., Bommes D., Campen M. "Volume Parametrization Quantization for Hexahedral Meshing." *ACM Trans. Graph.*, vol. 41, no. 4, jul 2022. URL `https://doi.org/10.1145/3528223.3530123`

[6] Eppstein D., Mumford E. "Steinitz Theorems for Orthogonal Polyhedra." *Proceedings of the Twenty-Sixth Annual Symposium on Computational Geometry*, SoCG '10, p. 429–438. Association for Computing Machinery, New York, NY, USA, 2010. URL `https://doi.org/10.1145/1810959.1811030`

[7] Zhao H., Li X., Wang W., Wang X., Wang S., Lei N., Gu X. "Polycube Shape Space." *Computer Graphics Forum*, vol. 38, 07 2019

[8] Sokolov D. *Modélisation Géométrique*. HDR, Université de Lorraine, 2016. URL `https://members.loria.fr/DSokolov/hdr/`

[9] Dumery C., Protais F., Mestrallet S., Bourcier C., Ledoux F. "Evocube: a Genetic Labeling Framework for Polycube-Maps." *Computer Graphics Forum*, 2022

[10] Livesu M., Vining N., Sheffer A., Gregson J., Scateni R. "PolyCut: Monotone Graph-Cuts for PolyCube Base-Complex Construction." *Transactions on Graphics (Proc. SIGGRAPH ASIA 2013)*, vol. 32, no. 6, 2013

[11] Fu X.M., Bai C.Y., Liu Y. "Efficient Volumetric PolyCube-Map Construction." *Computer Graphics Forum*, vol. 35, no. 7, 97–106, 2016. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13007`

[12] Guo H.X., Liu X., Yan D.M., Liu Y. "Cut-Enhanced PolyCube-Maps for Feature-Aware All-Hex Meshing." *ACM Trans. Graph.*, vol. 39, no. 4, aug 2020. URL `https://doi.org/10.1145/3386569.3392378`

[13] Ledoux F. "MAMBO: Model dAtabase Mesh BlOcking.", 2019. URL `https://gitlab.com/franck.ledoux/mambo`

[14] Sokolov D., Ray N. "Fixing normal constraints for generation of polycubes." Research report, LORIA, Oct. 2015. URL `https://hal.inria.fr/hal-01211408`

[15] Koch S., Matveev A., Jiang Z., Williams F., Artemov A., Burnaev E., Alexa M., Zorin D., Panozzo D. "ABC: A Big CAD Model Dataset For Geometric Deep Learning." *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019

[16] Gao X., Shen H., Panozzo D. "Feature Preserving Octree-Based Hexahedral Meshing - Dataset.", 2019. URL `https://cims.nyu.edu/gcl/papers/2019-OctreeMeshing.zip`

[17] Protais F. "fastbndpolycube.", 2022. URL `https://github.com/fprotais/fastbndpolycube`

[18] Zhao H., Lei N., Li X., Peng Z., Xu K., Gu X. "Robust edge-preserving surface mesh polycube deformation." *Computational Visual Media*, vol. 4, 01 2018