Polycube Shape Space

Hui Zhao, Xuan Li, Wencheng Wang, Xiaoling Wang, Shaodong Wang, Na Lei, Xianfeng Gu

State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences

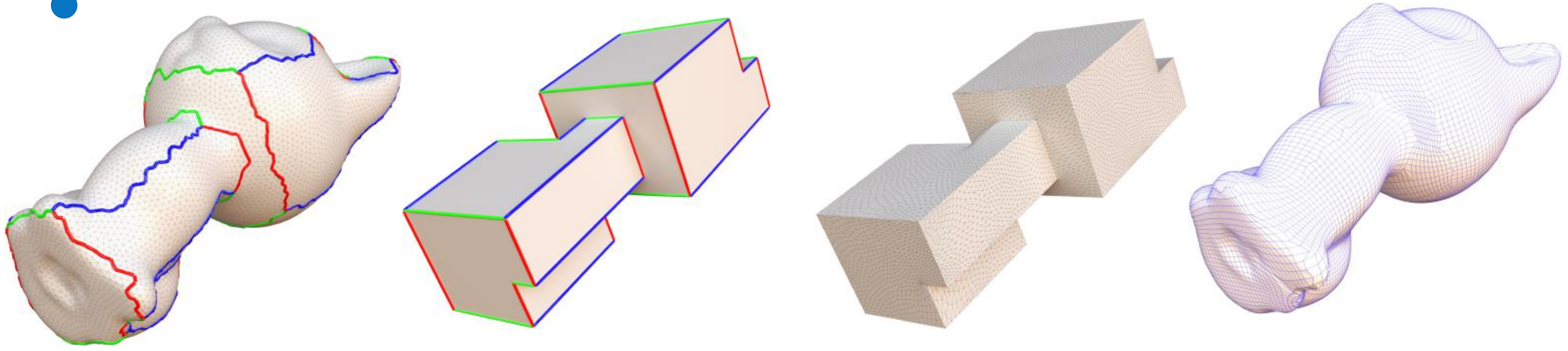
University of Chinese Academy of Sciences

University of Science and Technology Beijing

State University of New York at Stony Brook

Dalian University of Technology

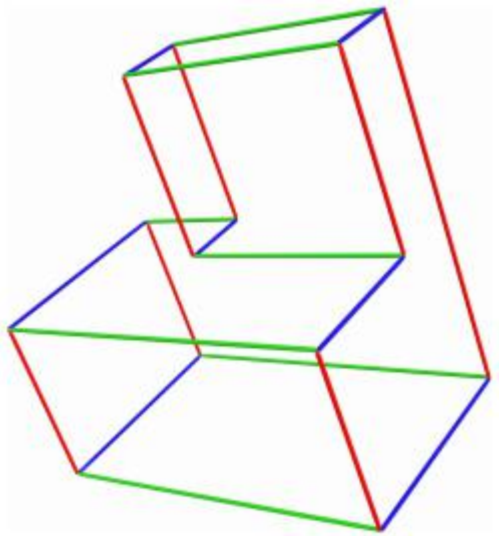
PACIFIC GRAPHICS 2019



Introduction

Given a polyhedron

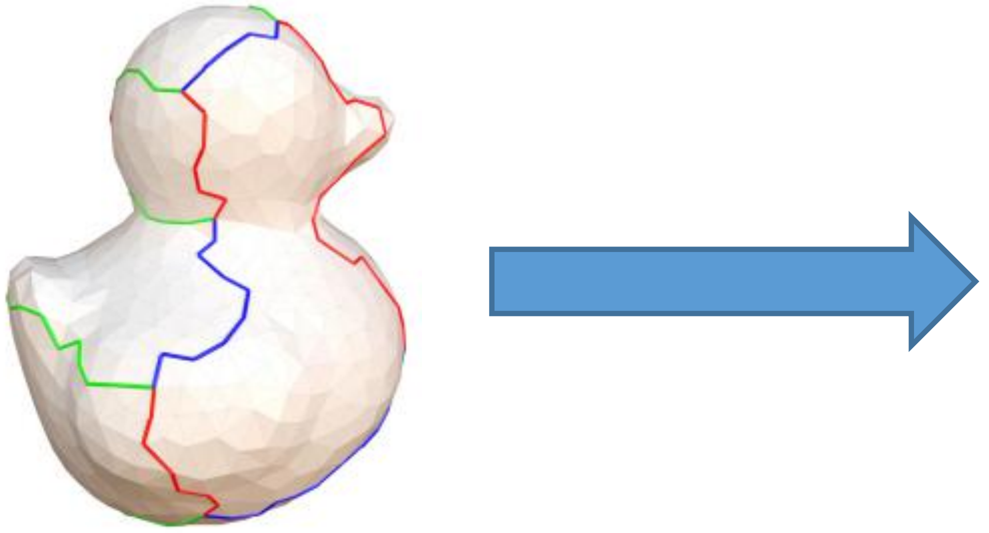
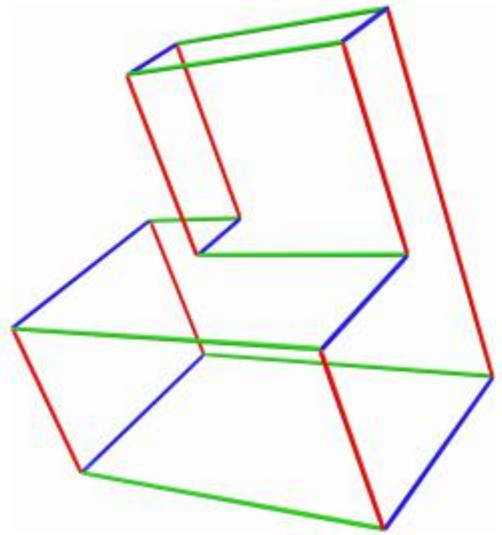
It is easy to extract its skeleton graph



Given a graph

Three problem:

1. Is there a polyhedron whose graph is the given one?
2. If so, how to generate the polyhedron?
3. Any applications in mesh processing for this kinds of relationship?

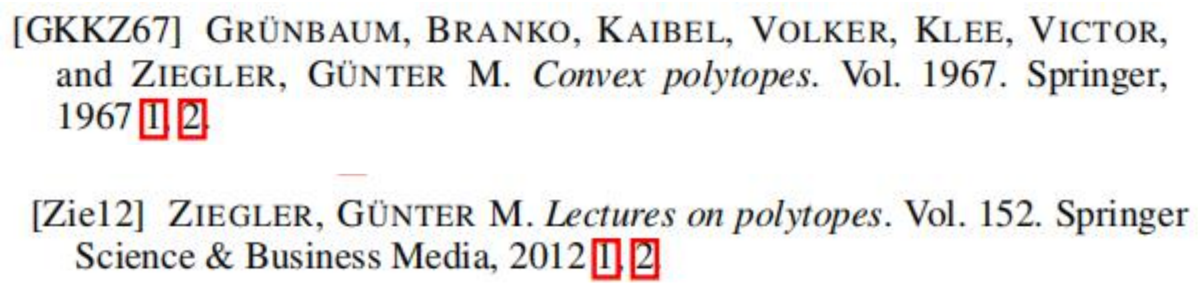
Outline

Introduction

Steintiz’s Theorem:

a graph is the skeleton graph of a convex polyhedron if and only if it is 3-connected and planar.

Branko Grünbaum call it : “ the most important and deepest known result on 3-polytopes ”



[GKKZ67] Grünbaum, Branko, Kaibel, Volker, Klee, Victor, and Ziegler, Günter M. *Convex polytopes*. Vol. 1967. Springer, 1967 [1], [2].

[Zie12] Ziegler, Günter M. *Lectures on polytopes*. Vol. 152. Springer Science & Business Media, 2012 [1], [2].

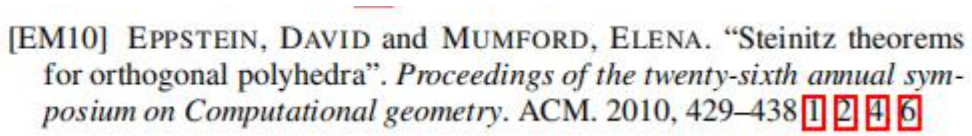
Introduction

Eppstein et al. :

"a graph is the skeleton graph of a simple orthogonal polyhedron if and only if it is bipartite, planar, 3-regular and removal of any 2 of its vertices disconnects it into at most 2 components."

simple orthogonal polyhedral :

1) spherical topology;
2) simply-connected faces;
3) three mutually-perpendicular axis-parallel edges meeting at each vertex;
4) non-convex polyhedral


[EM10] EPPSTEIN, DAVID and MUMFORD, ELENA. "Steinitz theorems for orthogonal polyhedra". *Proceedings of the twenty-sixth annual symposium on Computational geometry*. ACM. 2010, 429–438 1 2 4 6

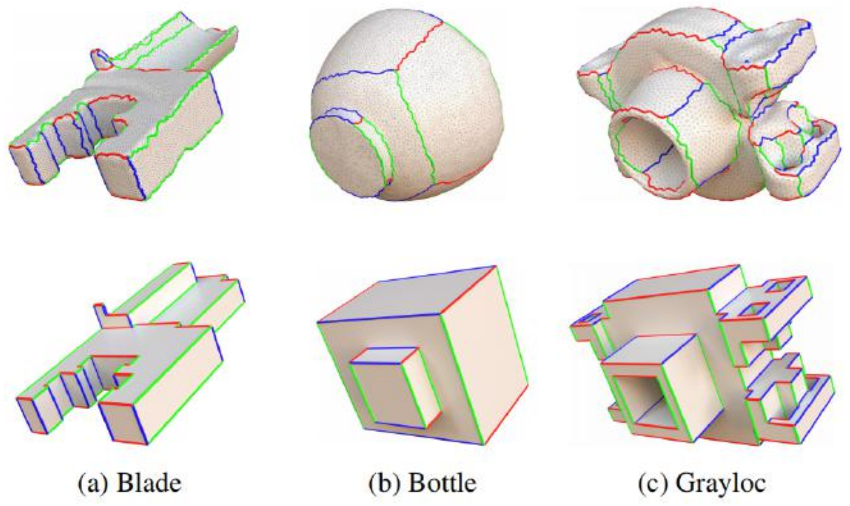
Introduction

Our works:

" a polycube graph is a skeleton graph of a polycube polyhedron if the patch number of the graph is bigger than 3."

Polycube polyhedral of any genus with non-simply connected faces.

A linear system to generate a space of polycube polyhedra from a graph.



Introduction

Definition 3.1 A layout graph is called a *polycube graph* $G = (N, A, P)$ with the nodes N, the arcs A and the patches P, if it satisfies two conditions: 1) each node $n_i \in N$ is of valence 3, i.e. there are 3 arcs incident to it; 2) each patch $p_i \in P$ has an even valence, i.e. there are even number of nodes along the border of each patch.

Definition 3.2 A *polycube polyhedron* is a three-dimensional polyhedron (not necessarily convex) which has exactly three mutually-perpendicular axis-parallel edges meeting at every vertex.

Definition 3.3 If a *polycube graph* has an embedding in $\mathbb{R}^3$ to form a *polycube polyhedron*, such an embedding is called a *polycube embedding* of the polycube graph.

Non-valid graphs for our approach

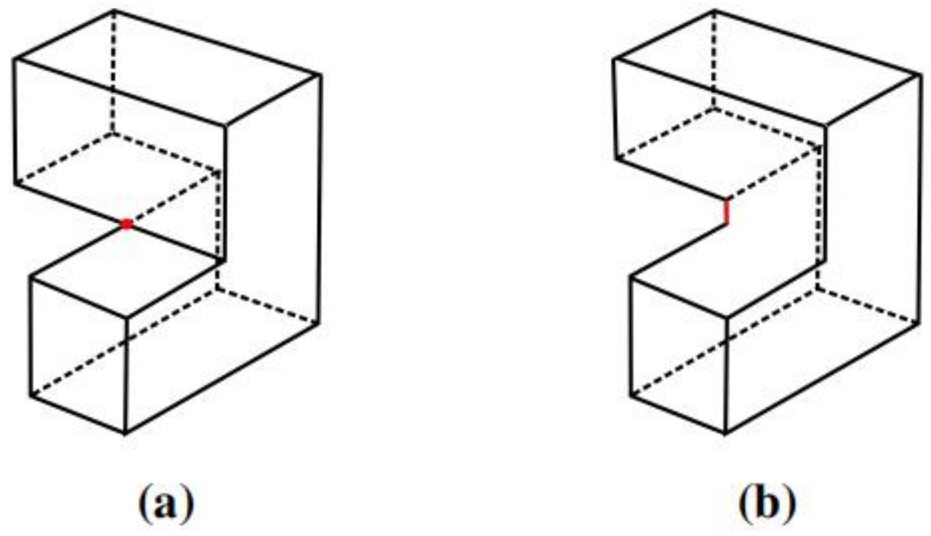


Figure 3: *(a) A polycube that violates our requirements; (b) modified valid version.*

Polycube Shape Space

Introduction

Intuitive idea: every edge assigned a vector, then the sum of the vectors around any loop is equal to zero.

We model this fact by the language of differential one-form, and results in a linear system.

Introduction

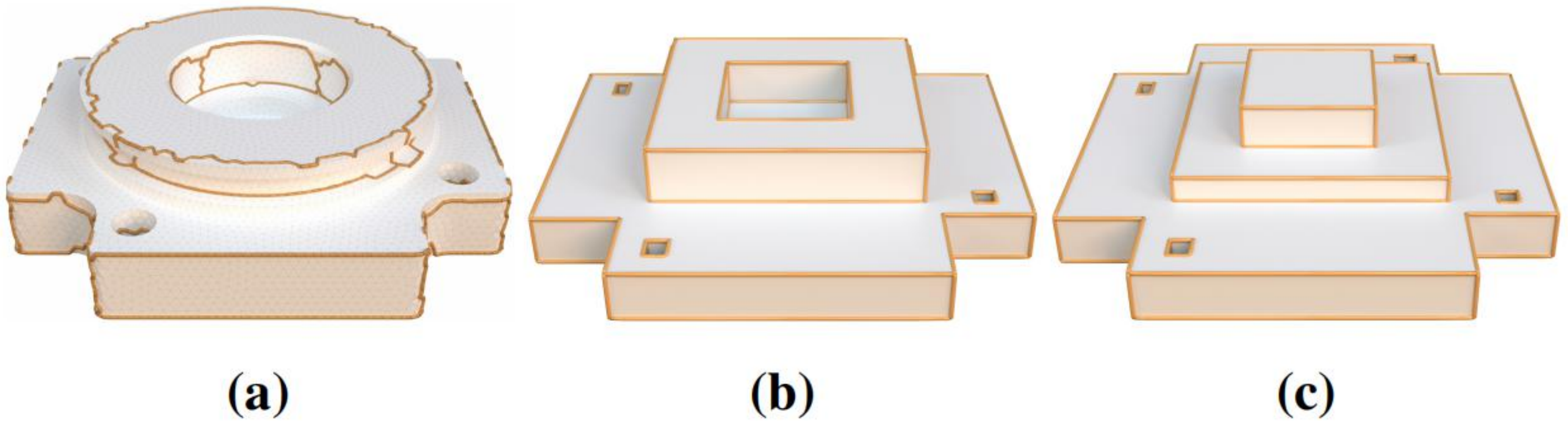


Figure 8: *Two different* polycube polyhedra *(b), (c) correspond to the same* polycube graph *(a).*

Introduction

Given an valid input graph, we build a “polycube shape space” with discrete differential one-form.

“polycube shape space” means all polycubes whose layout graphs are the input one.

Theory

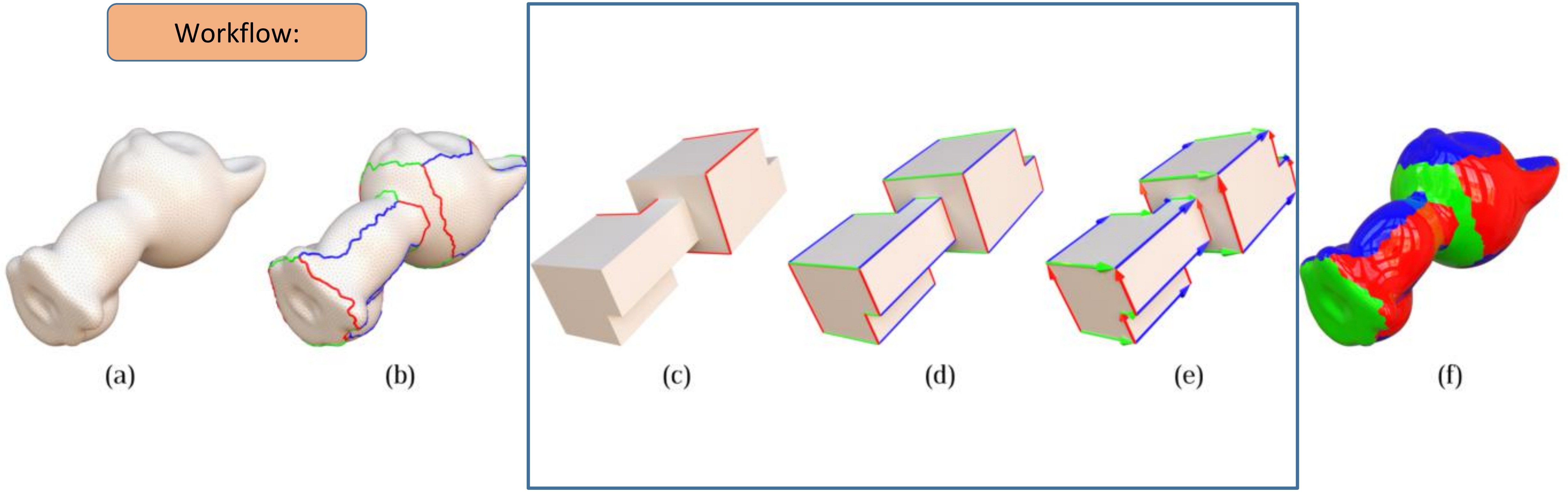
How to get the vectors on edges of graph?

1. arc colorization: determine the coordinate axis (X, Y or Z) for the vectors

2. Linear system: determine the size of the vectors

Introduction

Workflow:

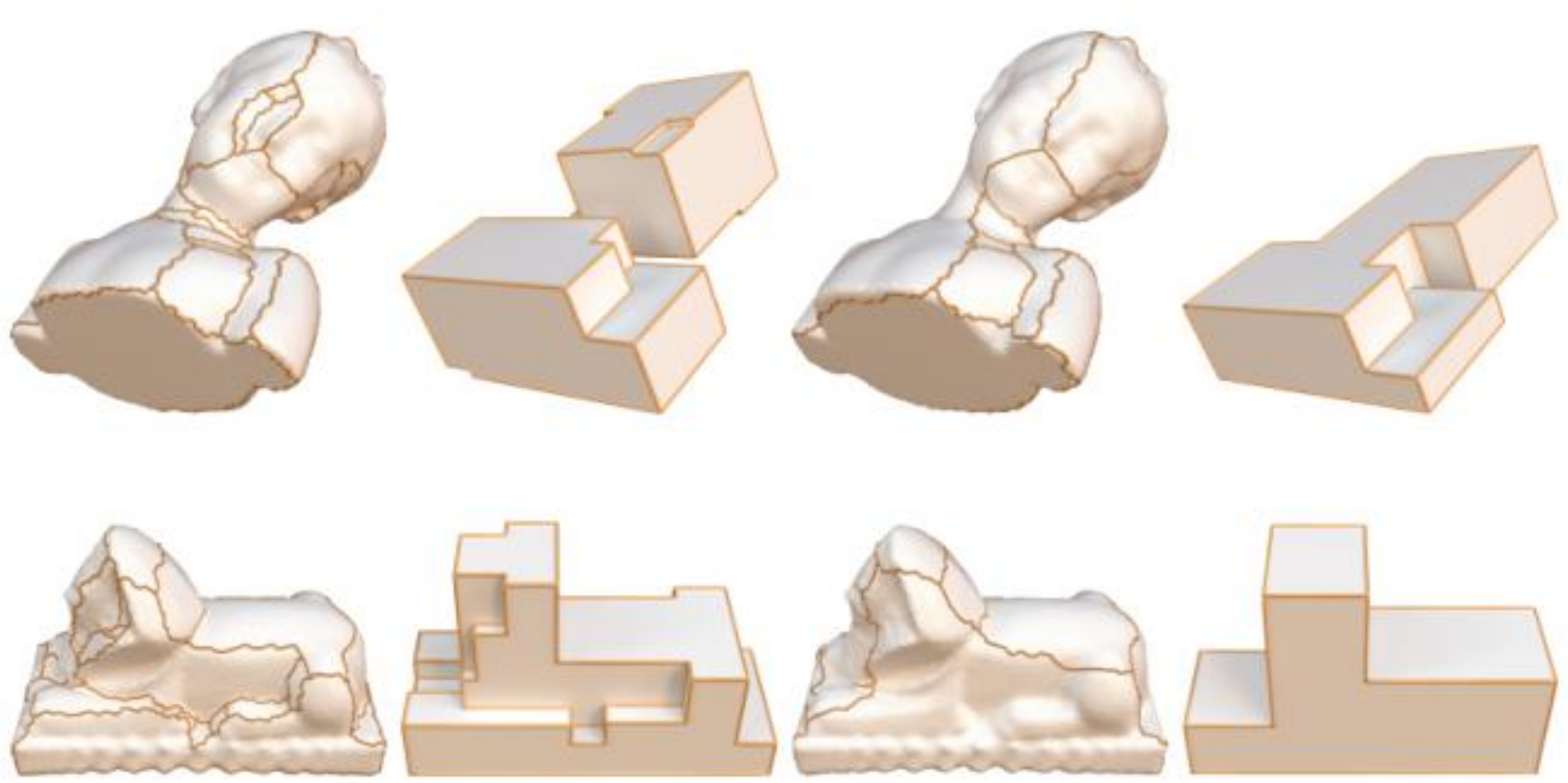


three steps to compute polycube shape steps

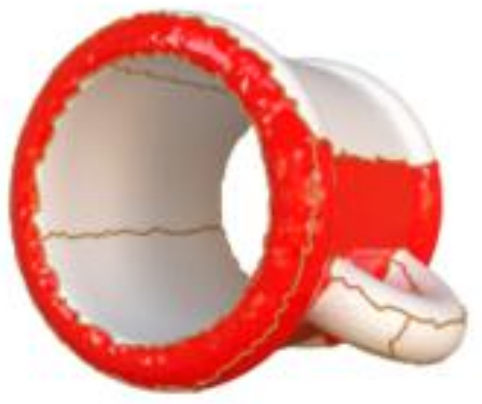
Theory

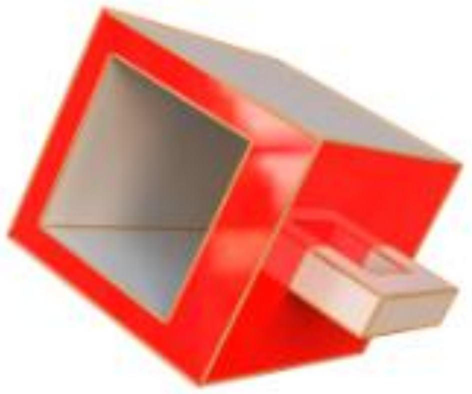
Input graph

Input graphs

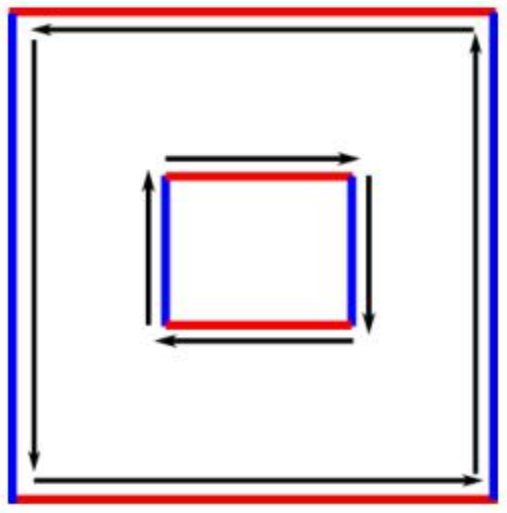

(a) layout 1 (b) polycube 1 (c) layout 2 (d) polycube 2

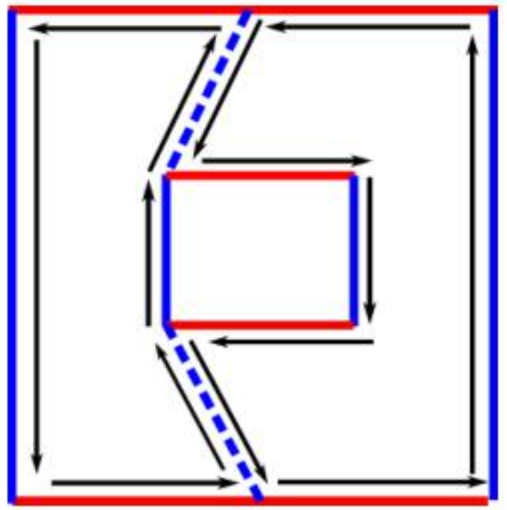
Change the graph to remove Non simply connected faces


(a) model

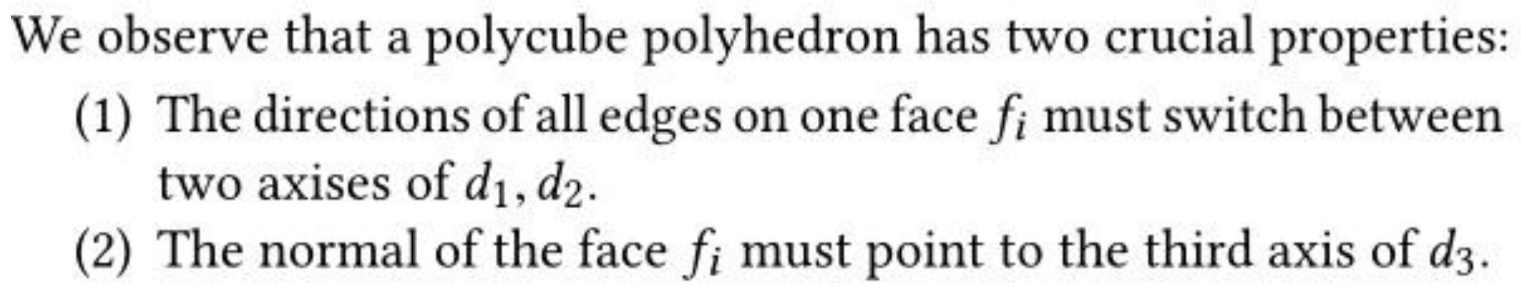

(b) polycube

The non-simply connected faces.


(a) Non-simply


(b) simply

Arc Colorization



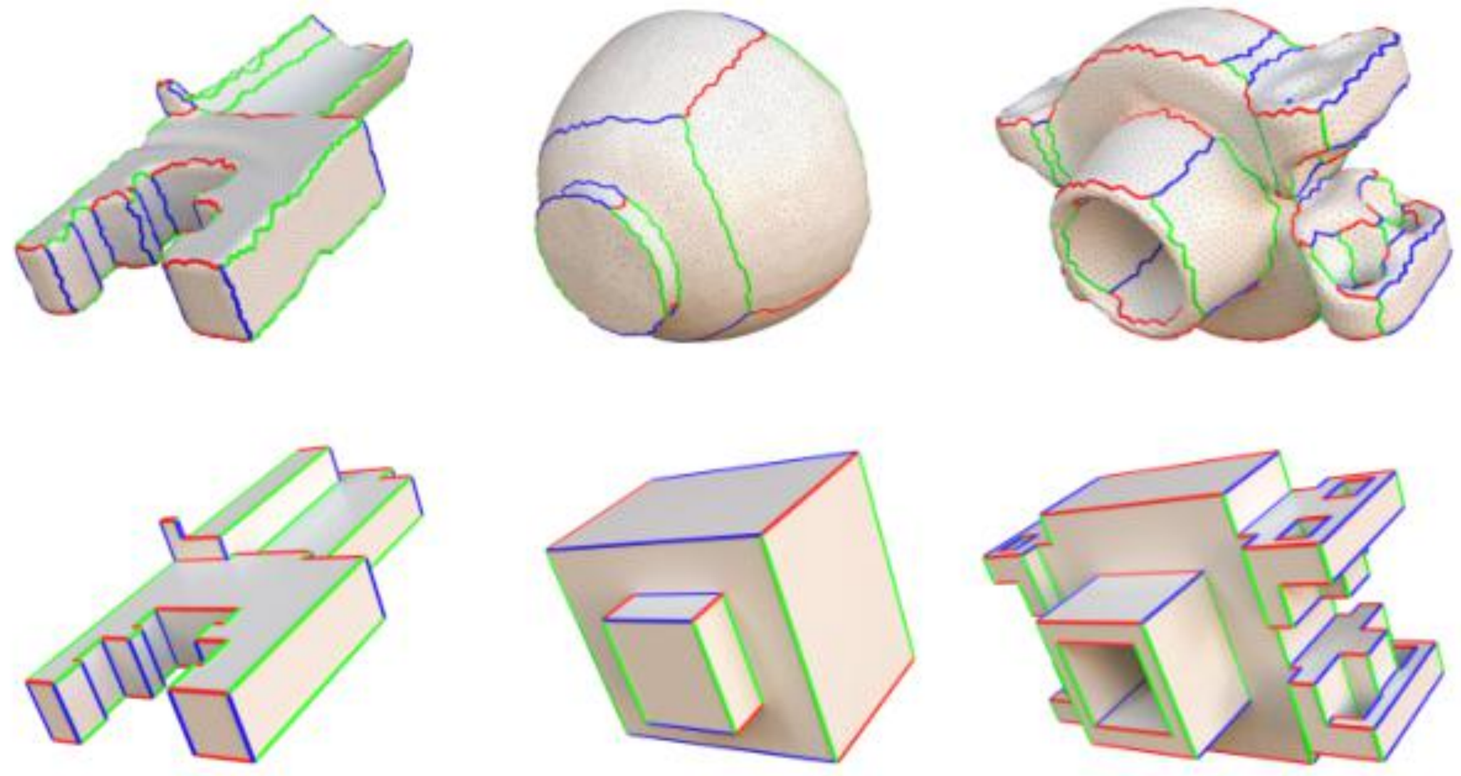
We observe that a polycube polyhedron has two crucial properties:

(1) The directions of all edges on one face f_i must switch between two axises of d_1, d_2.

(2) The normal of the face f_i must point to the third axis of d_3.

1. Three colors have six kinds of permutations
2. a colorization for every permutation
3. six colorizations for a polycube layout graph in total.
4. all of them are equivalent under rotation and refection transformations.
5. choose any one as our input for the next step.

Arc Colorization


(a) blade (b) bottle (c) grayloc

Theory

Polycube shape space by exact one-form

1. The color label on one arc can denote the axis which the arc belongs to
2. but it can not distinguish the positive or negative directions of the axis

Assign a vector to every edges

Polycube shape space by exact one-form

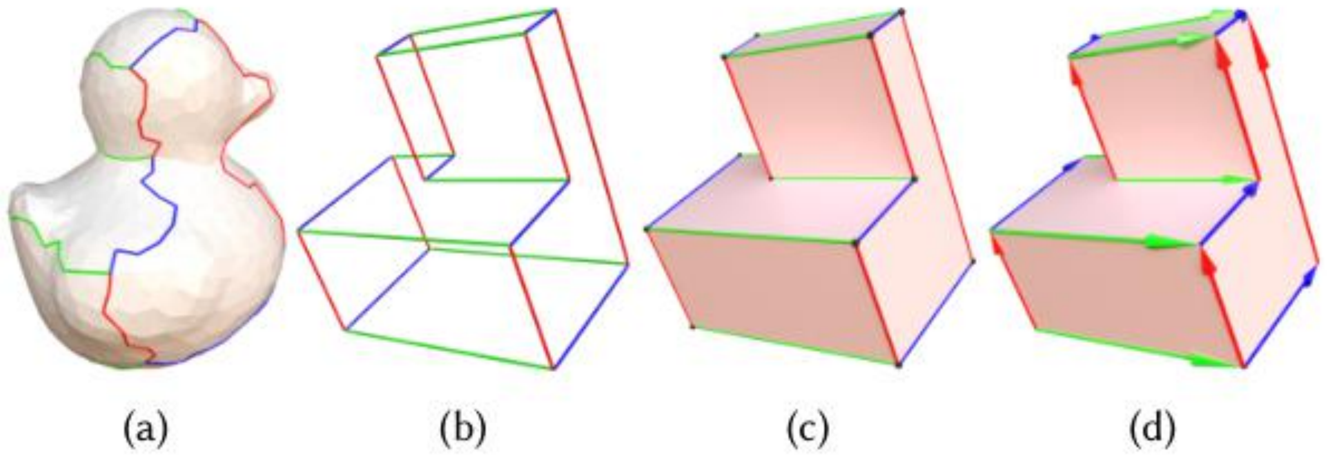
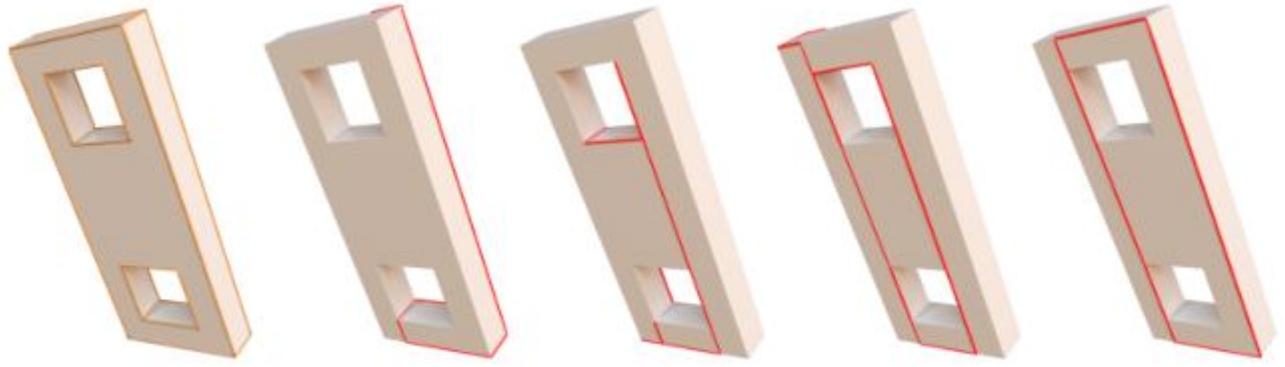


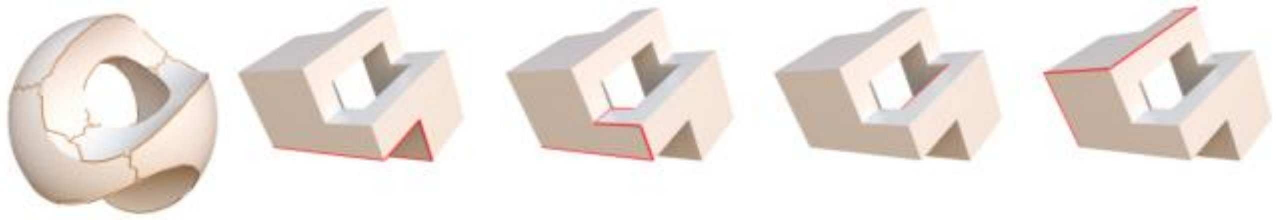
Fig. 10. The colorized arcs are shown in (a), (b), (c), and the initial directions are exhibited in (d).

The sum of edge vectors around any face is zero

Theory

The sum of edge vectors around homology loop is zero


(a) graph (b) loop1 (c) loop2 (d) loop3 (e) loop4


(a) graph (b) loop1 (c) loop2 (d) loop3 (e) loop4

W is a matrix, L is the set of vectors on edges

$$W \cdot L = 0$$

If the dimension of the kernel space *K* of the matrix *W* is zero,

it means that there is no valid *polycube polyhedron*.

Theorem 4.1 If a polycube graph can realize a polycube polyhedron embedded in $\mathbb{R}^3$, then the dimension of the corresponding kernel space K is not zero.

Six basis of the keneral space

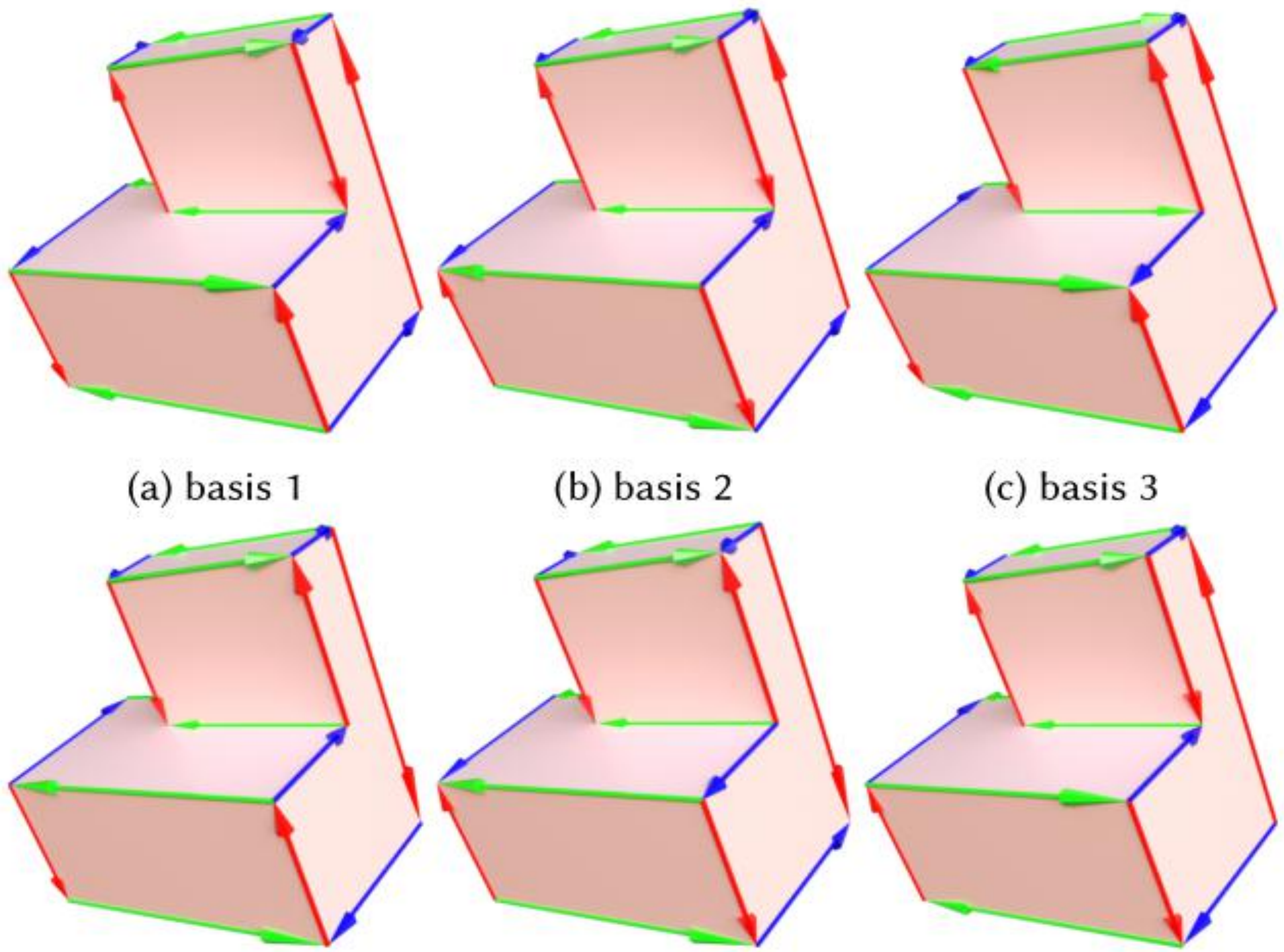

(a) basis 1 (b) basis 2 (c) basis 3
(d) basis 4 (e) basis 5 (f) basis 6

Fig. 14. The basis of the polycube shape space for the "duck" mesh and its layout graph.

Theory

We proved:

Theorem 4.2 For a polycube graph with the number of nodes $|N|$, arcs $|A|$, patches $|P|$ respectively, the dimension of its polycube shape space is at least $|P|-3$.

Extensive experiments validate our theorem

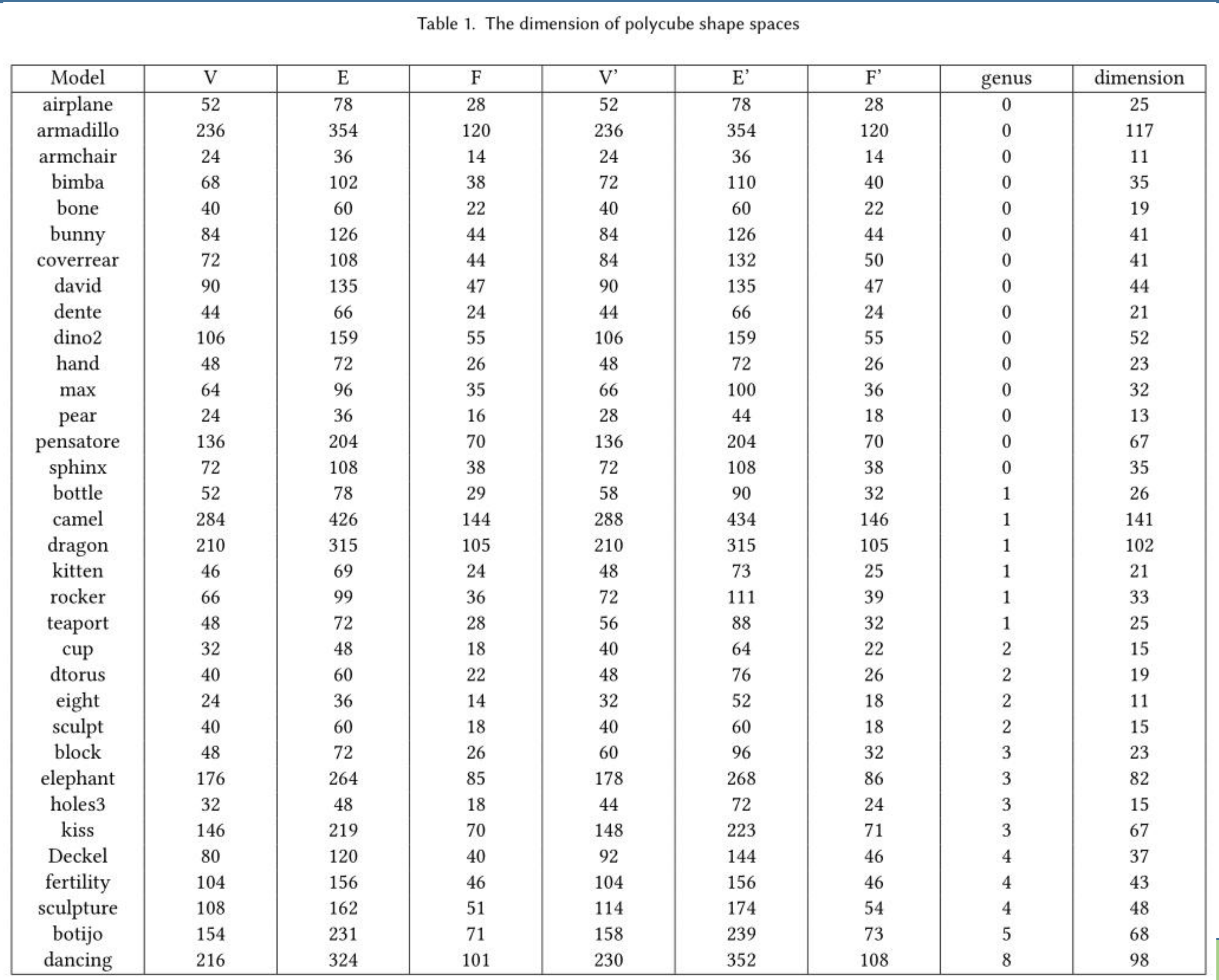


Table 1. The dimension of polycube shape spaces

Model	V	E	F	V'	E'	F'	genus	dimension
airplane	52	78	28	52	78	28	0	25
armadillo	236	354	120	236	354	120	0	117
armchair	24	36	14	24	36	14	0	11
bimba	68	102	38	72	110	40	0	35
bone	40	60	22	40	60	22	0	19
bunny	84	126	44	84	126	44	0	41
coverrear	72	108	44	84	132	50	0	41
david	90	135	47	90	135	47	0	44
dente	44	66	24	44	66	24	0	21
dino2	106	159	55	106	159	55	0	52
hand	48	72	26	48	72	26	0	23
max	64	96	35	66	100	36	0	32
pear	24	36	16	28	44	18	0	13
pensatore	136	204	70	136	204	70	0	67
sphinx	72	108	38	72	108	38	0	35
bottle	52	78	29	58	90	32	1	26
camel	284	426	144	288	434	146	1	141
dragon	210	315	105	210	315	105	1	102
kitten	46	69	24	48	73	25	1	21
rocker	66	99	36	72	111	39	1	33
teaport	48	72	28	56	88	32	1	25
cup	32	48	18	40	64	22	2	15
dtorus	40	60	22	48	76	26	2	19
eight	24	36	14	32	52	18	2	11
sculpt	40	60	18	40	60	18	2	15
block	48	72	26	60	96	32	3	23
elephant	176	264	85	178	268	86	3	82
holes3	32	48	18	44	72	24	3	15
kiss	146	219	70	148	223	71	3	67
Deckel	80	120	40	92	144	46	4	37
fertility	104	156	46	104	156	46	4	43
sculpture	108	162	51	114	174	54	4	48
botijo	154	231	71	158	239	73	5	68
dancing	216	324	101	230	352	108	8	98

Polycube Embedding

Find one optimal Embedding from the polycube shape space

1. Input a set of target positions for the nodes of the graph.

2. We look for an optimal embedding whose node positions approximate them

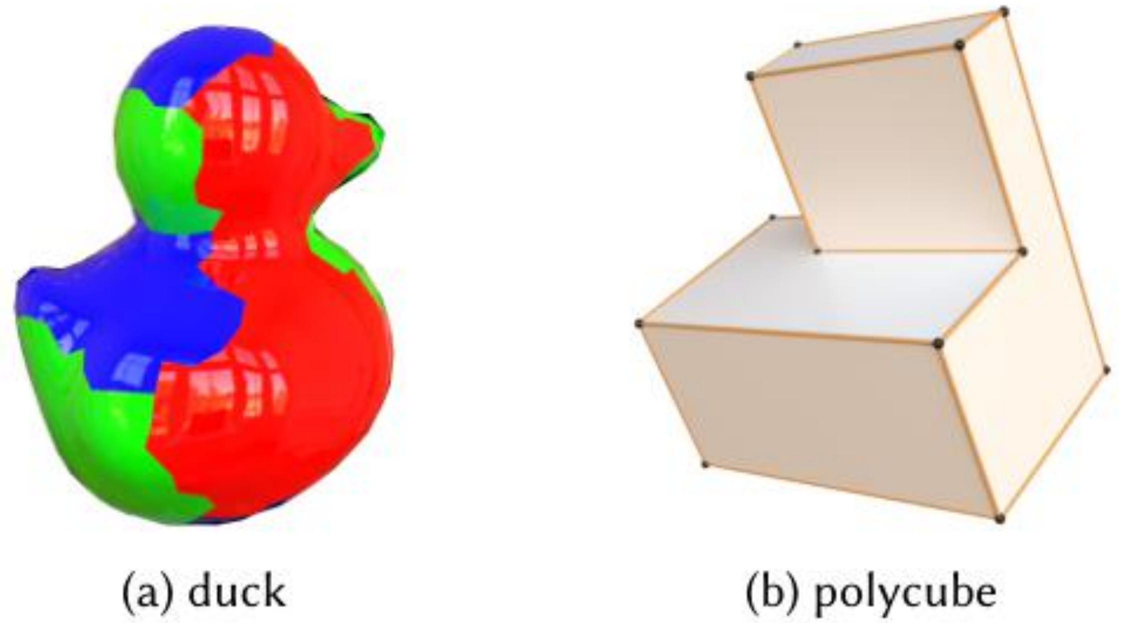


Fig. 15. The "duck" model and its optimal polycube polyhedron.

Theory

Algorithm:

1. a quadratic optimization method to obtain an optimal polycube polyhedron from this space.

2. employ linear inequality constraints to remove the degenerate cases.

3. add constraints on edge lengths to adjust the shape of the polycube polyhedron.

Polycube shape space contains degenerated embedding

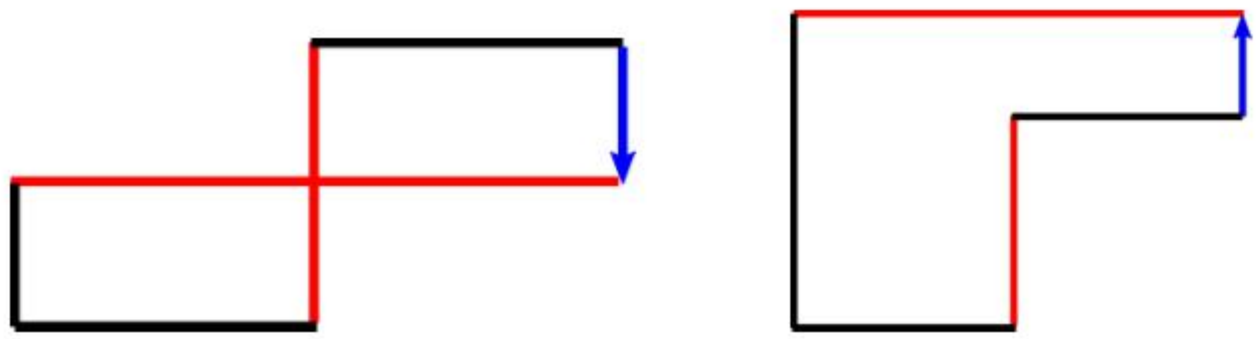

(a) degenerated (b) fixed

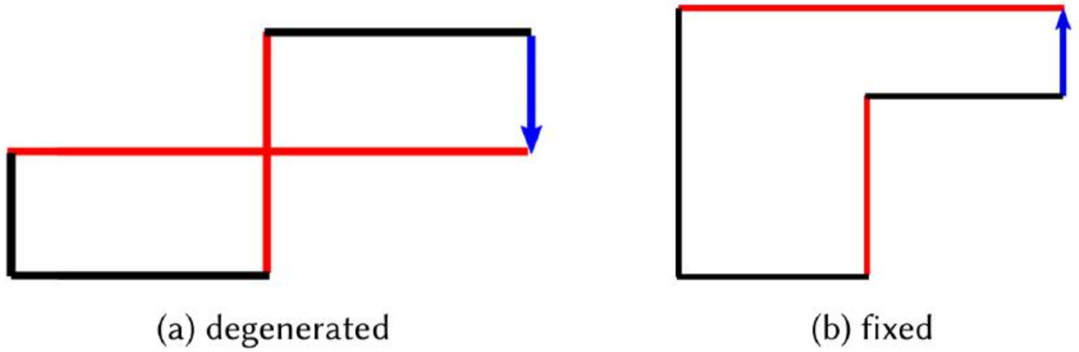
Fig. 17. The degenerated cases.

Quadratic programming

Coordinates on the original mesh

$$(S-\bar{S})^2 = \Big(\sum_{\omega^i \in N} c_i \cdot \int \omega^i + \mathbf{r}_0 - \bar{S}\Big)^2$$

$$\left.\begin{array}{l} l_i \geq \delta,\ or \\ l_i \leq -\delta\ or \\ (\omega_0(v_i)-\omega_0(v_j))\cdot \mathbf{n}_i \geq \delta,\ or \\ (\omega_0(v_i)-\omega_0(v_j))\cdot \mathbf{n}_i \leq -\delta \end{array}\right\} \longrightarrow$$


(a) degenerated (b) fixed

Edge Length constraints

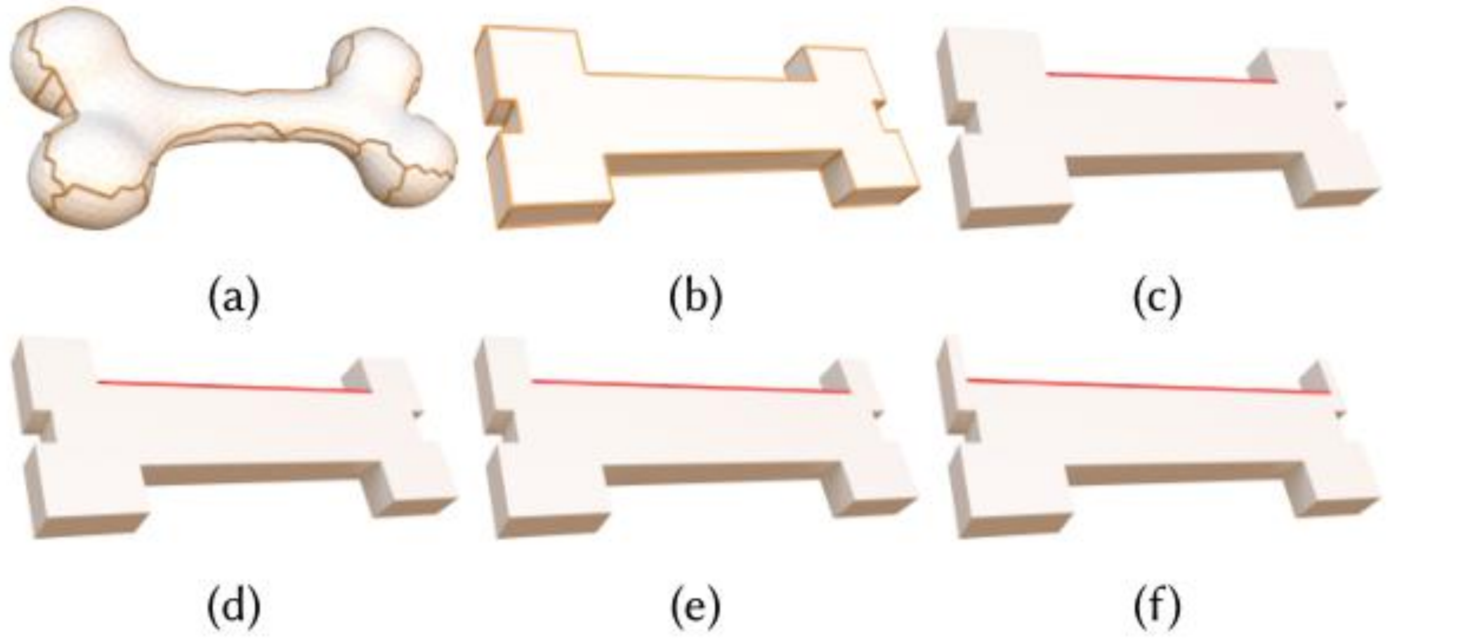


Fig. 16. The polycube embeddings with the different edge length constraints.

Quadratic programming

$$(S - \bar{S})^2 = (\sum_{\omega^i \in N} c_i \cdot \int \omega^i + \mathbf{r}_0 - \bar{S})^2$$

$$l_i \geq \delta,\ or$$

$$l_i \leq -\delta\ or$$

$$(\omega_0(v_i) - \omega_0(v_j)) \cdot \mathbf{n}_i \geq \delta,\ or$$

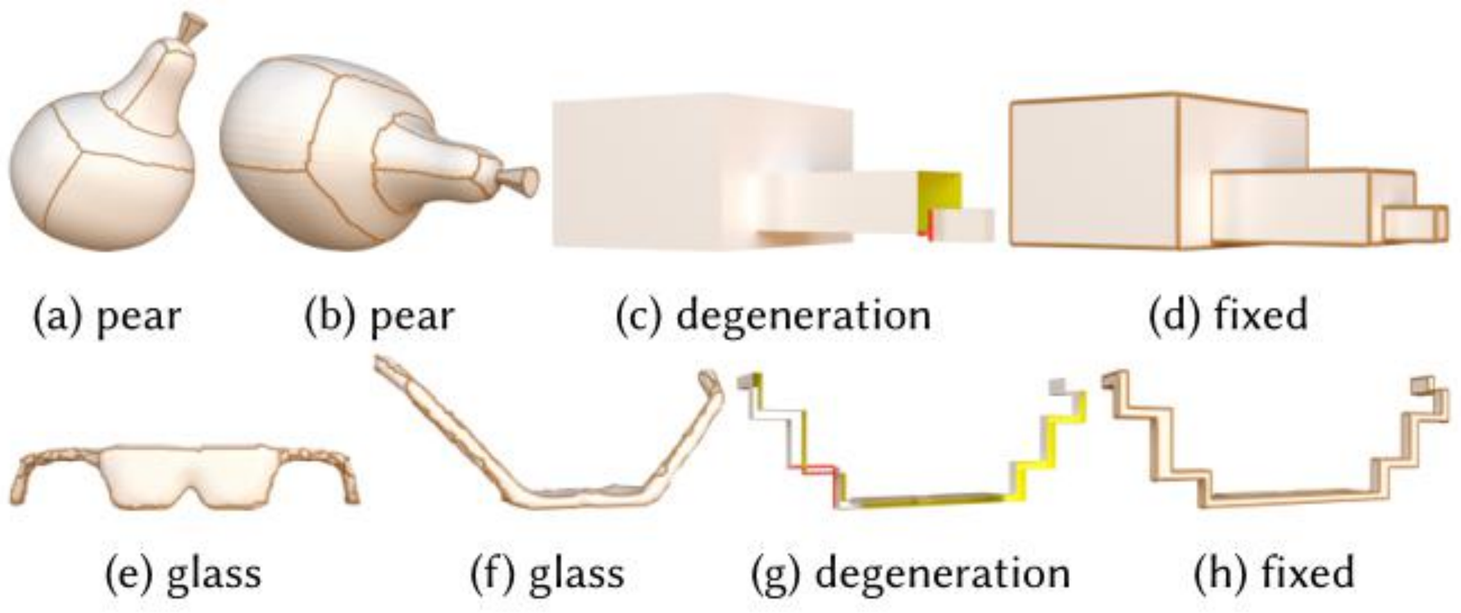
$$(\omega_0(v_i) - \omega_0(v_j)) \cdot \mathbf{n}_i \leq -\delta$$

where δ is some positive real number.

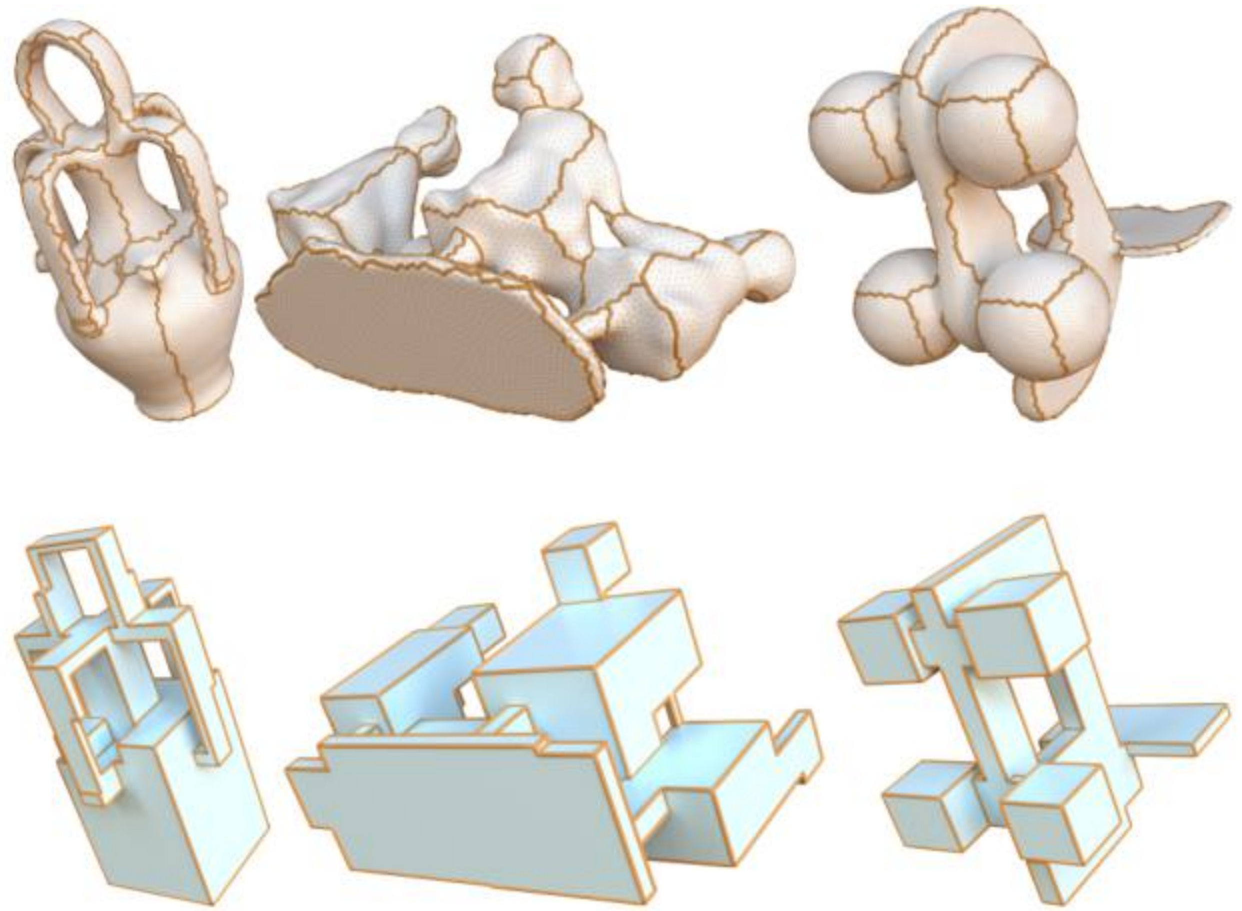
The inequality symbol is unknown, we use an interactive user interface to setup the inequality value and update the degenerated polycube to a valid one.

Interactive set the inequality constraints

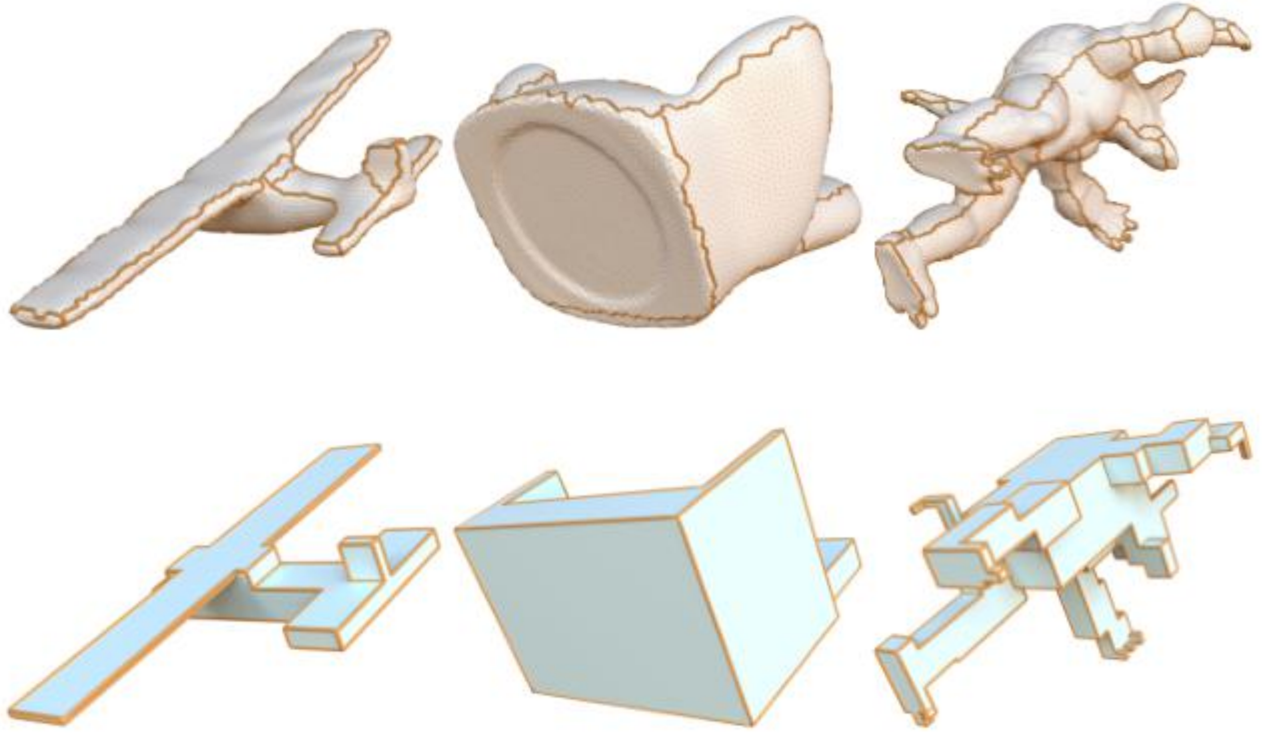
Degenerated cases



Experiments


(a) botijo (b) dancing children (c) elk

Experiments


(a) airplane (b) airchair (c) armodilo

Experiments

The applications of 1) polycube mesh, 2) quadrangulation, 3) hex-meshing

Related Works

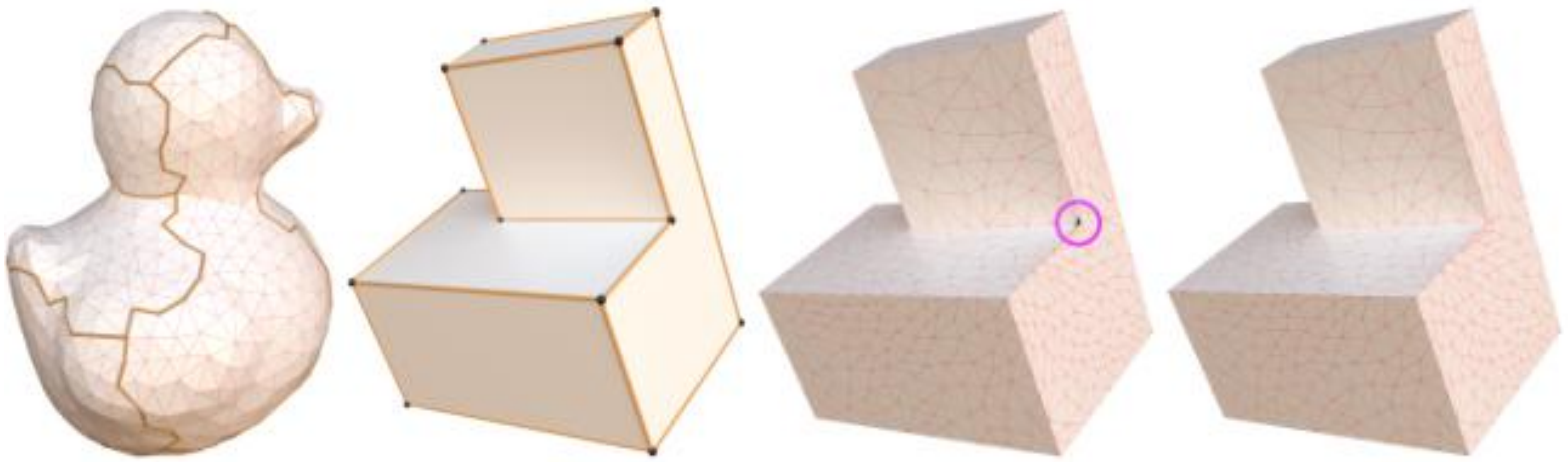
construct a polycube and a cross map at the same time.

- Deformation based: Fu et al. 2016; Gregson et al. 2011;Huang et al. 2014; Zhao et al. 2017
- Graph-cut based: Livesu et al. 2013.

Limitation: Only construct a single polycube.

Polycube parameterization

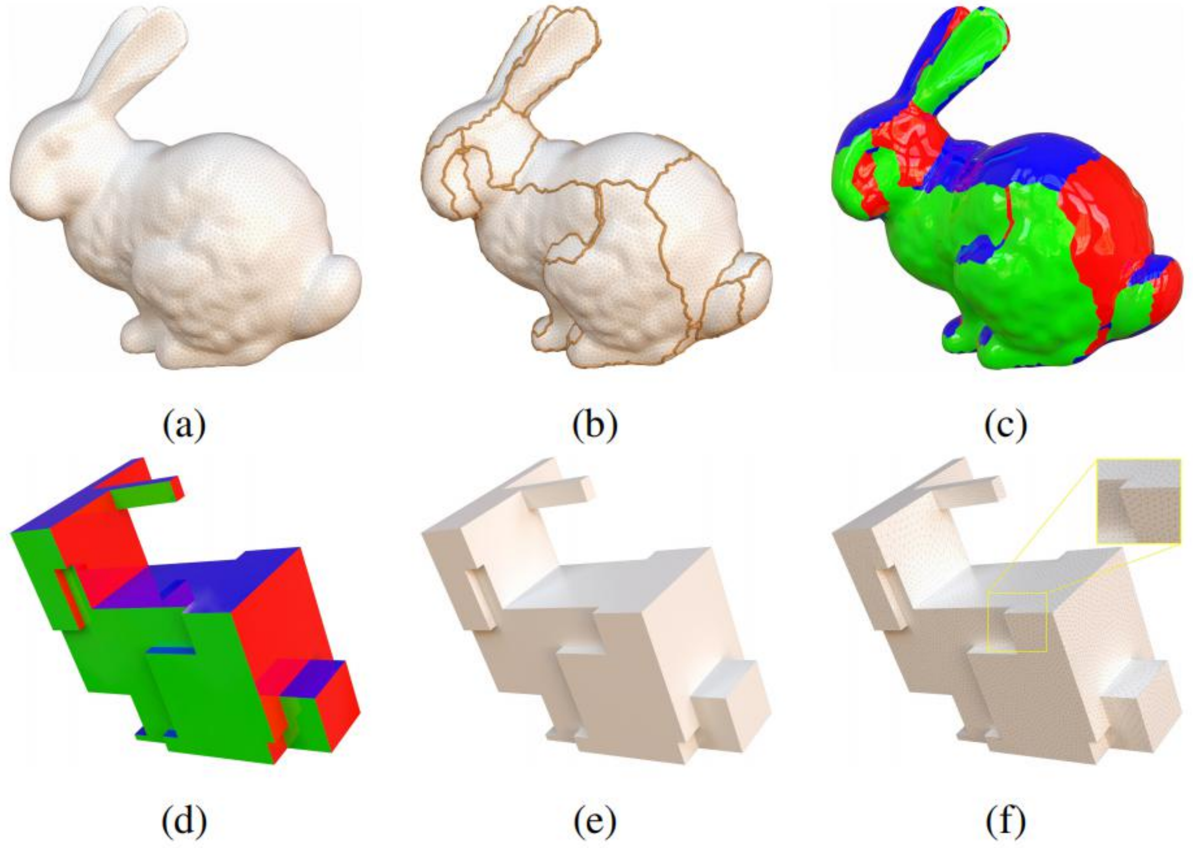
Map every mesh patch to polyhedron face one by one .



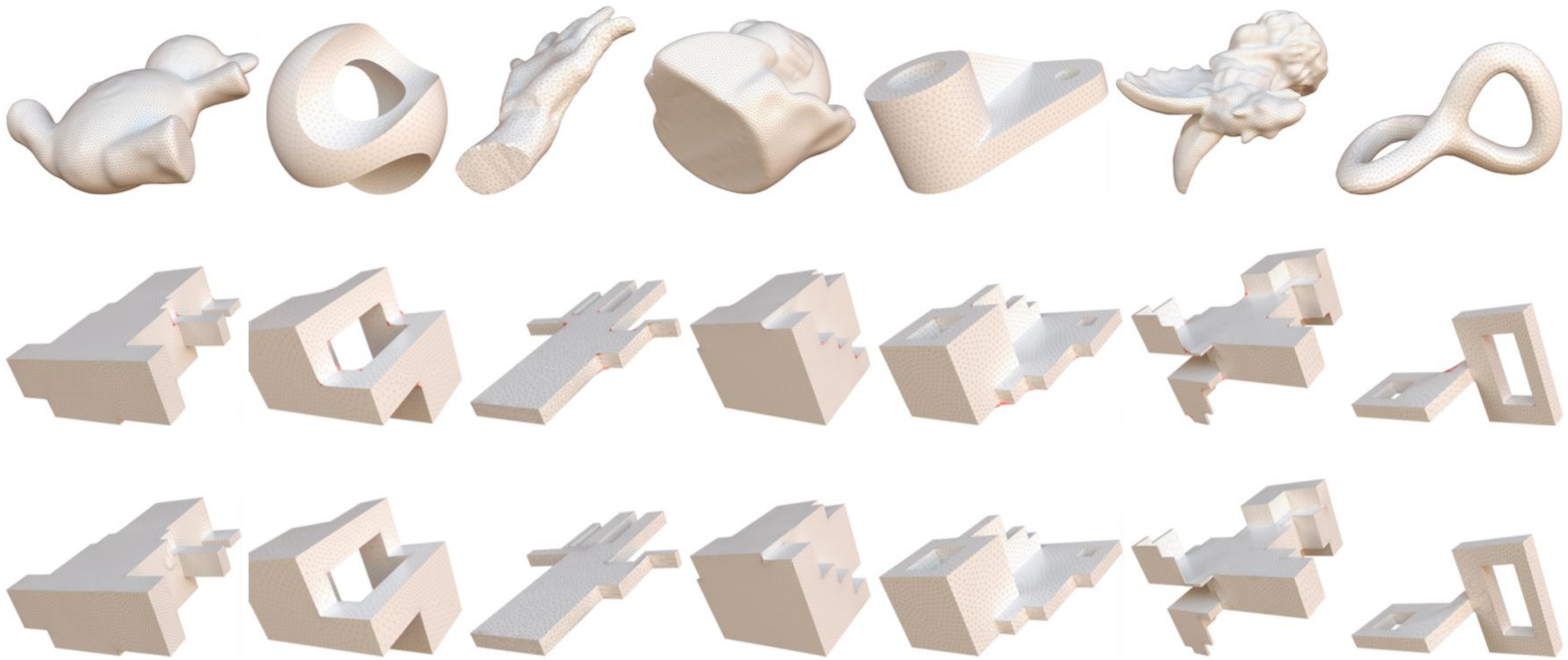
(a) duck (b) arc mapping (c) Tutte's (d) bijective

Fig. 19. The polycube parameterization.

Introduction



Polycube Mesh



Quad Meshing

Two steps:

1. Compute a quadrangular polycube mesh.
2. Pull back onto the original mesh by barycentric coordinates.

Experiments

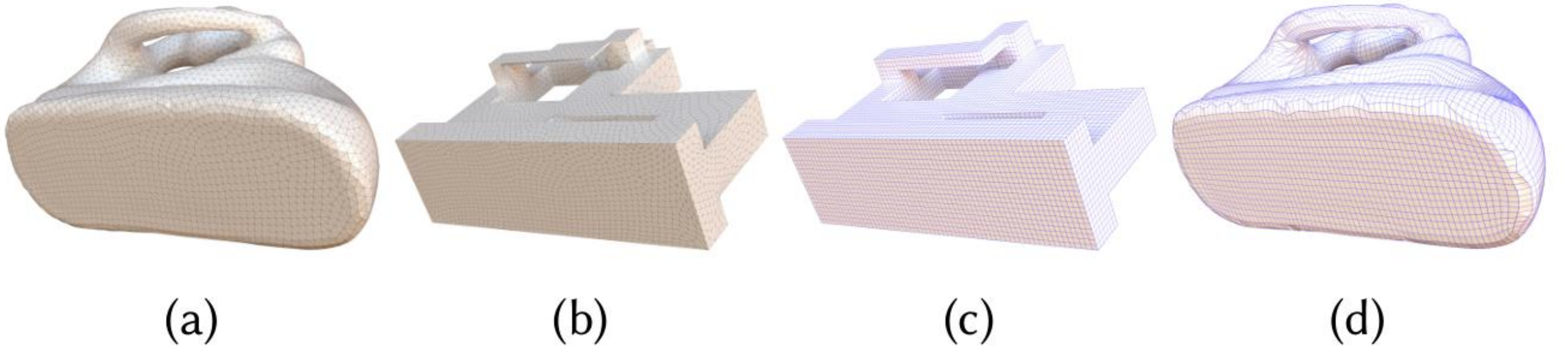


Fig. 22. The quad meshing work flow. (a),(b) are the original triangular mesh and its bijective cross-mapping polycube mesh, (c) is the quadrangular polycube mesh, (d) is the final quad meshing result.

Experiments

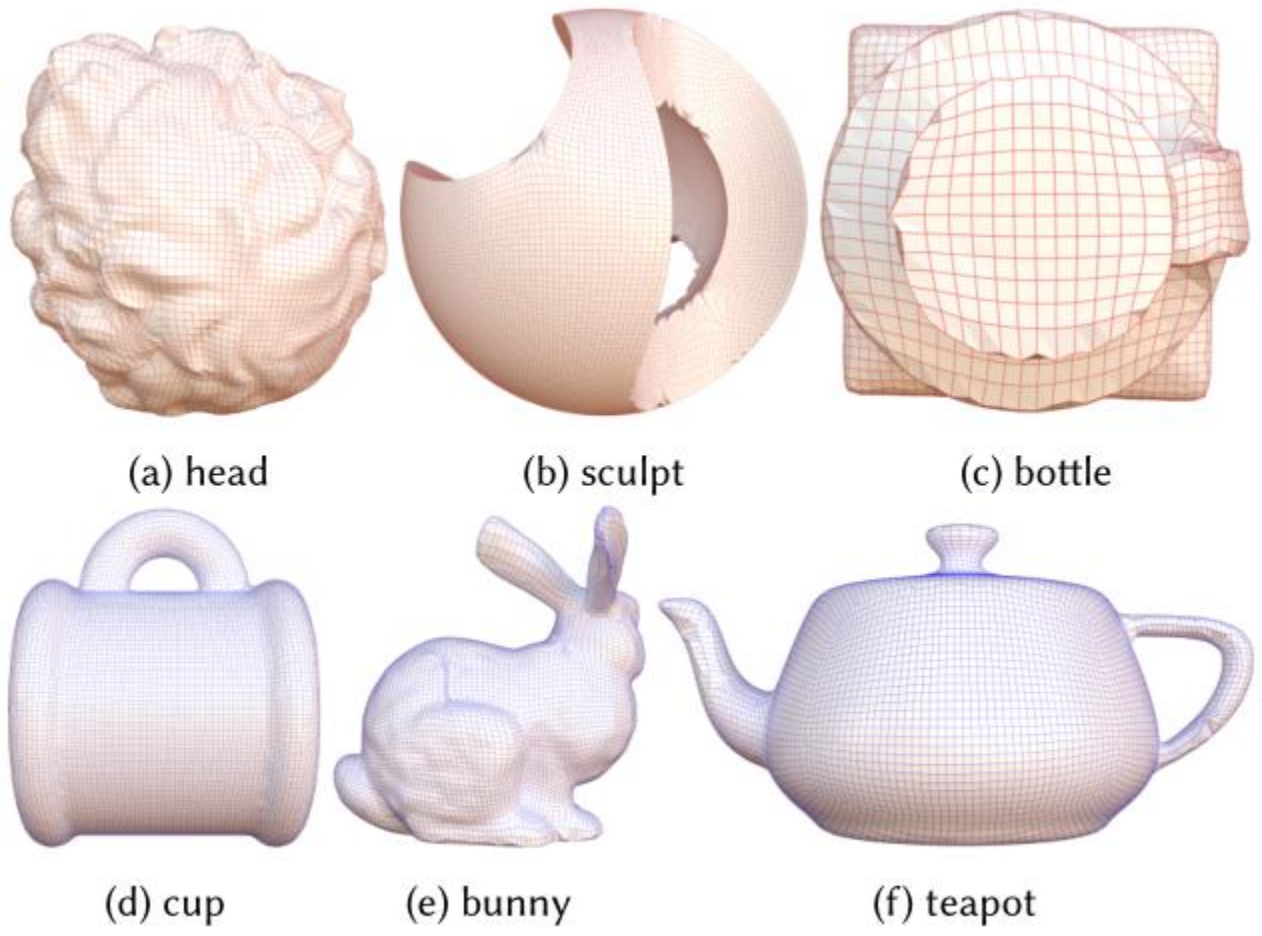
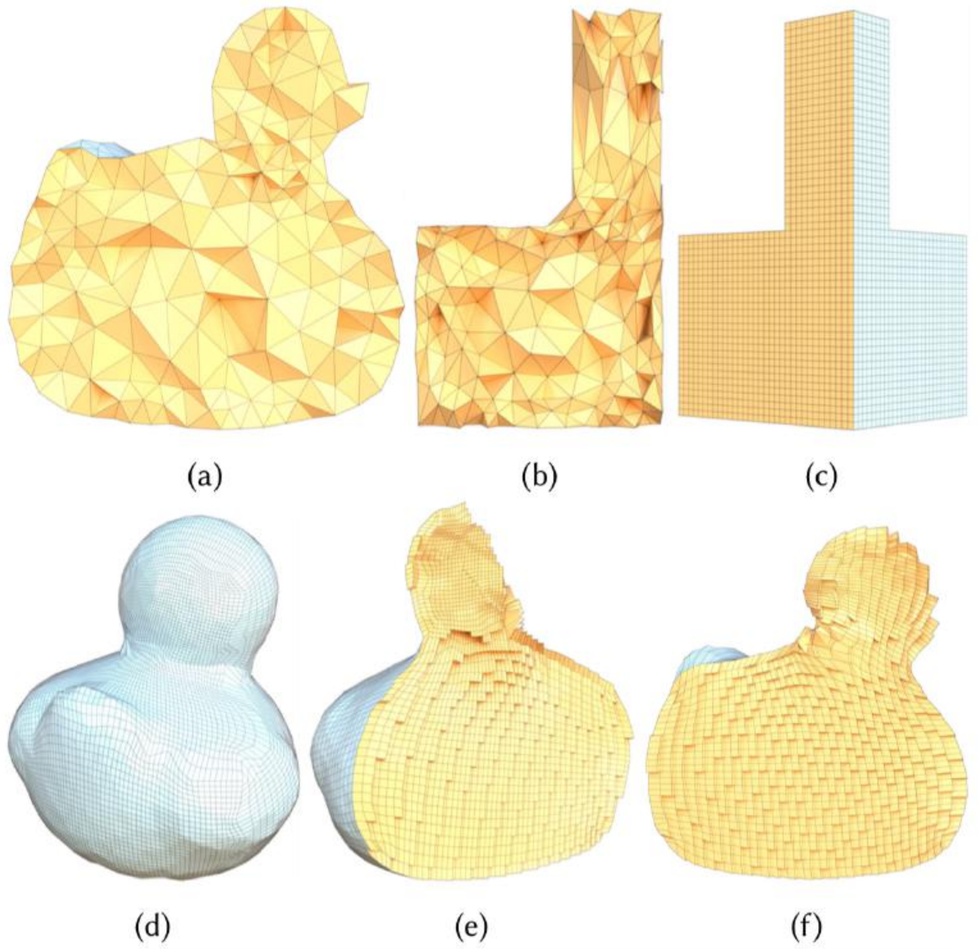


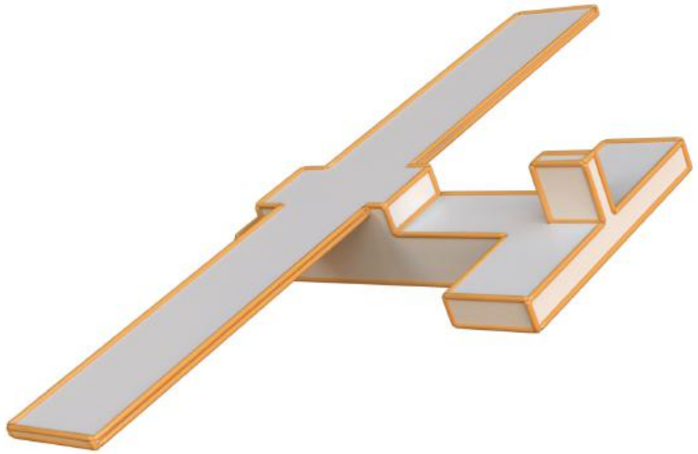
Fig. 23. The quad meshing results by our polycube method.

All-Hex Meshing

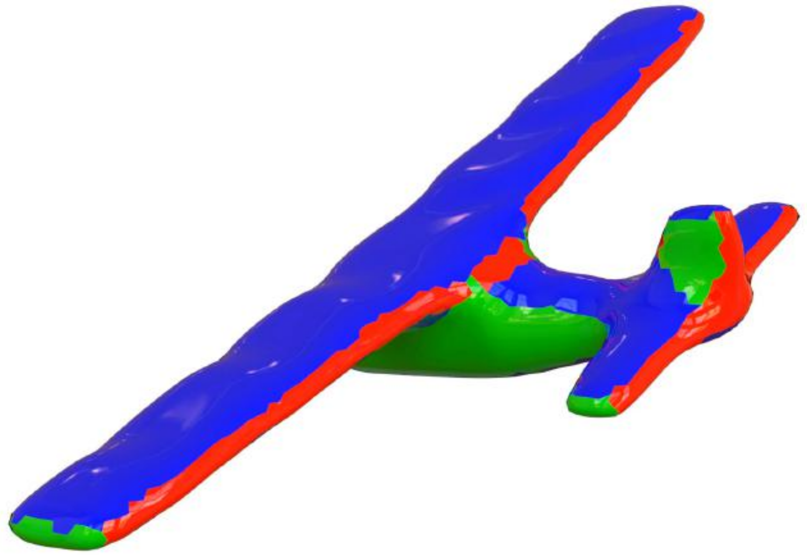
1. Tetrahedralize the original mesh.
2. Bijective map between original volumetric mesh and polycube volumetric mesh.
3. Tessellate the volume tetrahedral polycube mesh into a hexahedral polycube mesh.
4. Pull back by barycenter coordinates.

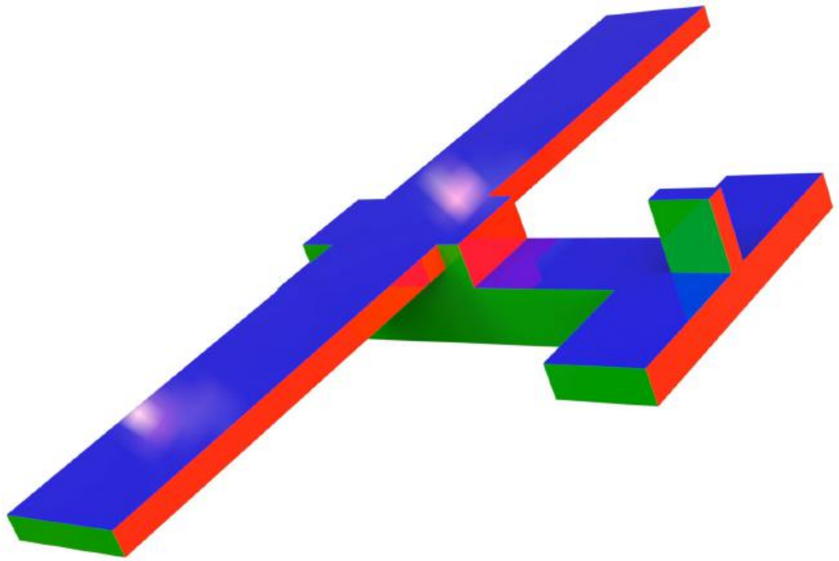


Experiments

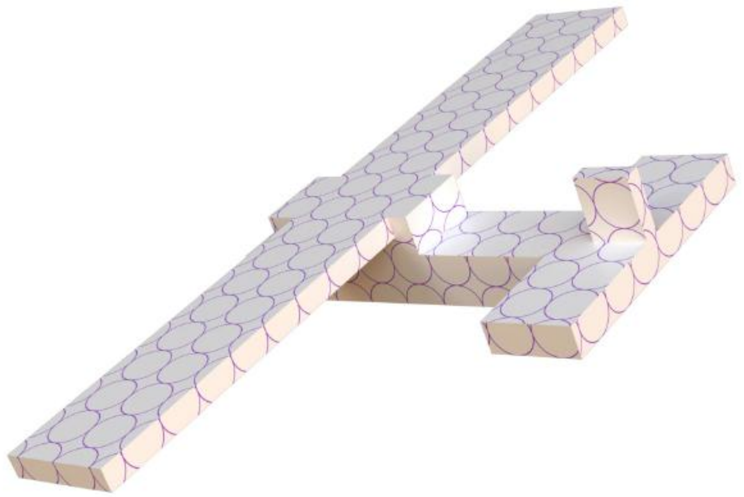


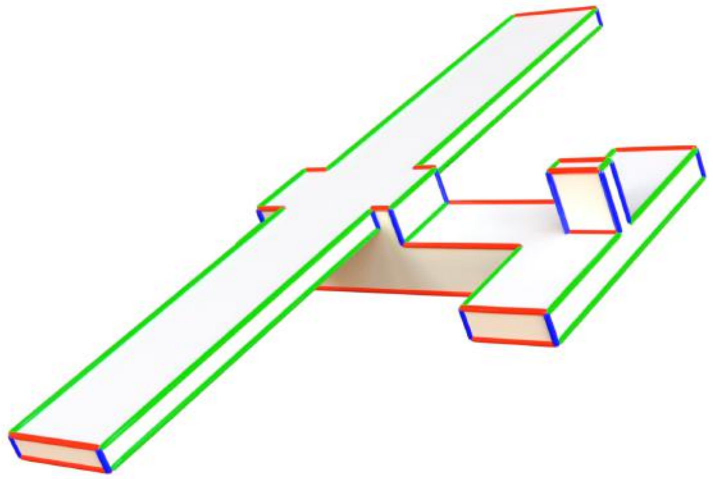
Experiments



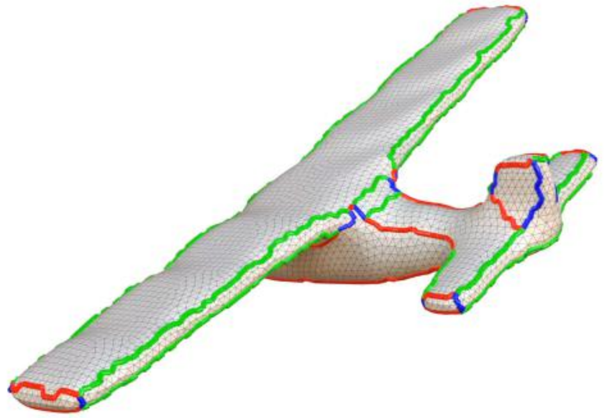
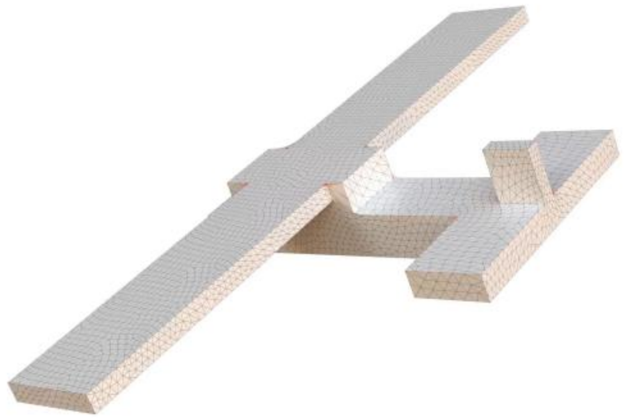
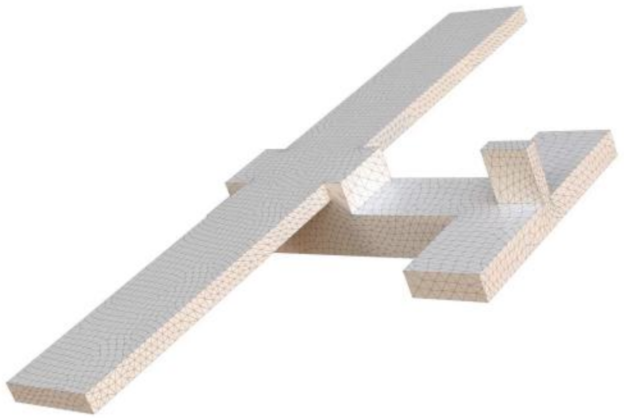


Experiments





Experiments

Future works

In this paper, we assume graphs are given,

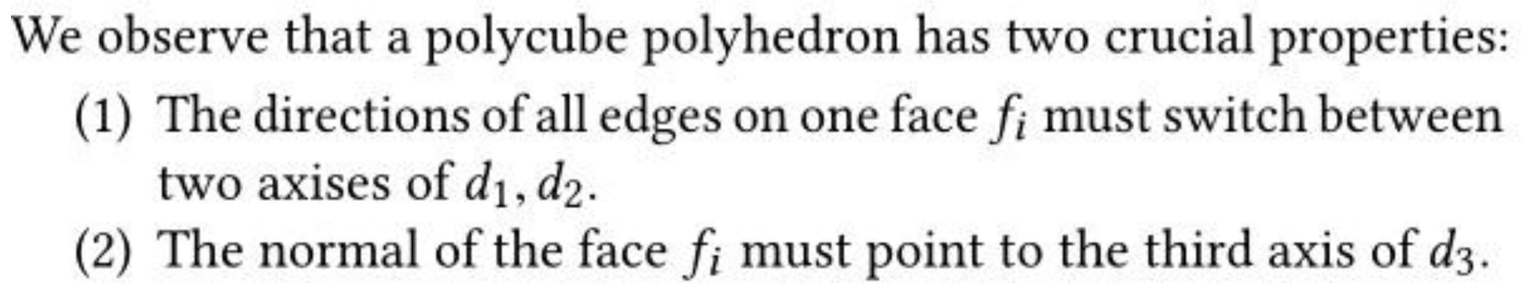
How to generate the optimal graph in terms of a certain kind of application?

Q&A

Appendix

Theory

Arc Colorization



We observe that a polycube polyhedron has two crucial properties:

(1) The directions of all edges on one face f_i must switch between two axises of d_1, d_2.

(2) The normal of the face f_i must point to the third axis of d_3.

1. Three colors have six kinds of permutations

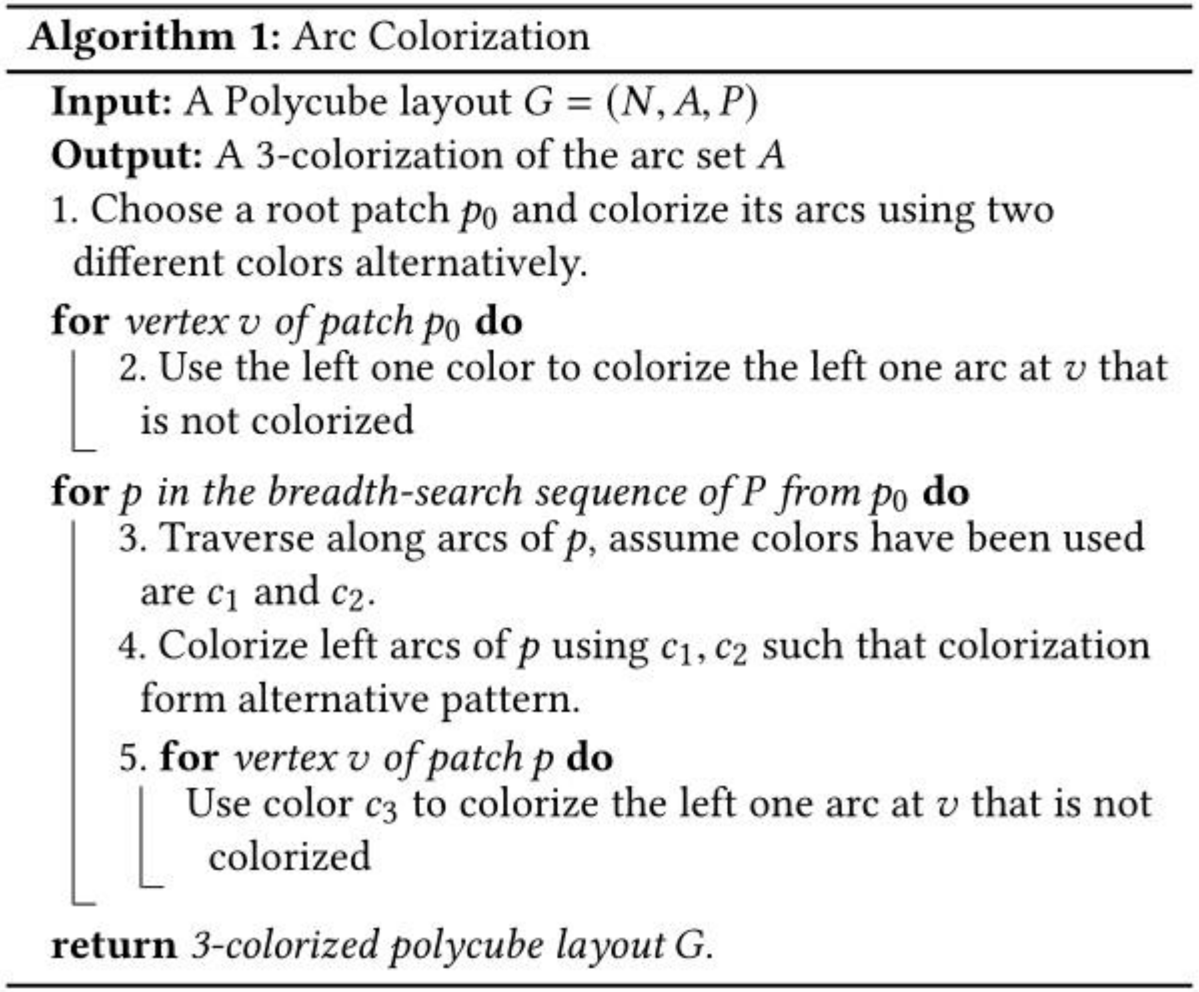
2. a colorization for every permutation

3. six colorizations for a polycube layout graph in total.

4. all of them are equivalent under rotation and refection transformations.

5. choose any one as our input for the next step.

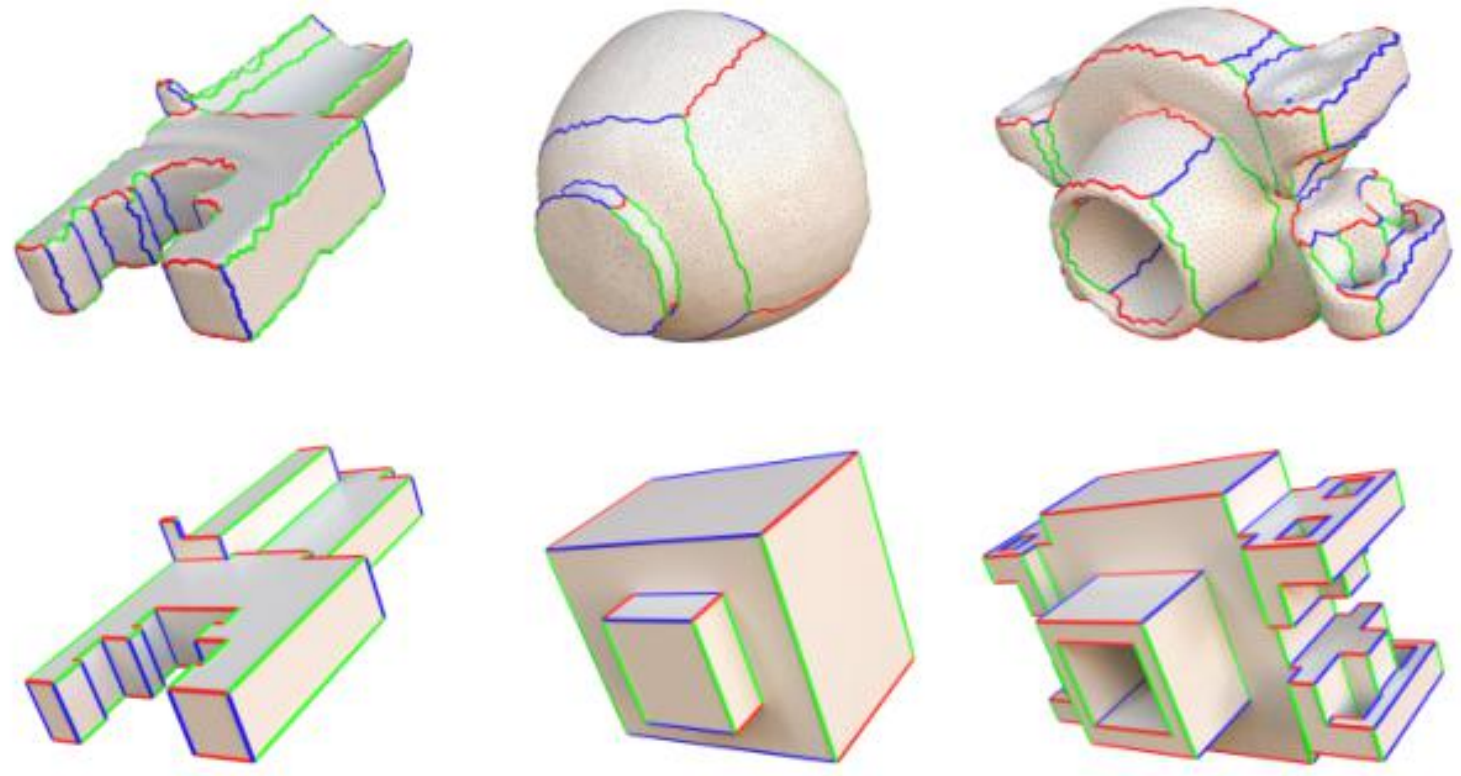
Arc Colorization



Algorithm 1: Arc Colorization

```
Input: A Polycube layout $G = (N, A, P)$
Output: A 3-colorization of the arc set $A$
1. Choose a root patch $p_0$ and colorize its arcs using two
   different colors alternatively.
for vertex $v$ of patch $p_0$ do
    2. Use the left one color to colorize the left one arc at $v$ that
       is not colorized
for $p$ in the breadth-search sequence of $P$ from $p_0$ do
    3. Traverse along arcs of $p$, assume colors have been used
       are $c_1$ and $c_2$.
    4. Colorize left arcs of $p$ using $c_1, c_2$ such that colorization
       form alternative pattern.
    5. for vertex $v$ of patch $p$ do
        Use color $c_3$ to colorize the left one arc at $v$ that is not
          colorized
return 3-colorized polycube layout $G$.
```

Arc Colorization


(a) blade (b) bottle (c) grayloc

Polycube space by exact one-form

1. The color label on one arc can denote the axis which the arc belongs to
2. but it can not distinguish the positive or negative directions of the axis

Intuitive idea: every edge assigned a vector, then the sum of the vectors around any loop is equal to zero.

We model this fact by differential one-forms, and results in a linear system.

Polycube shape space by exact one-form

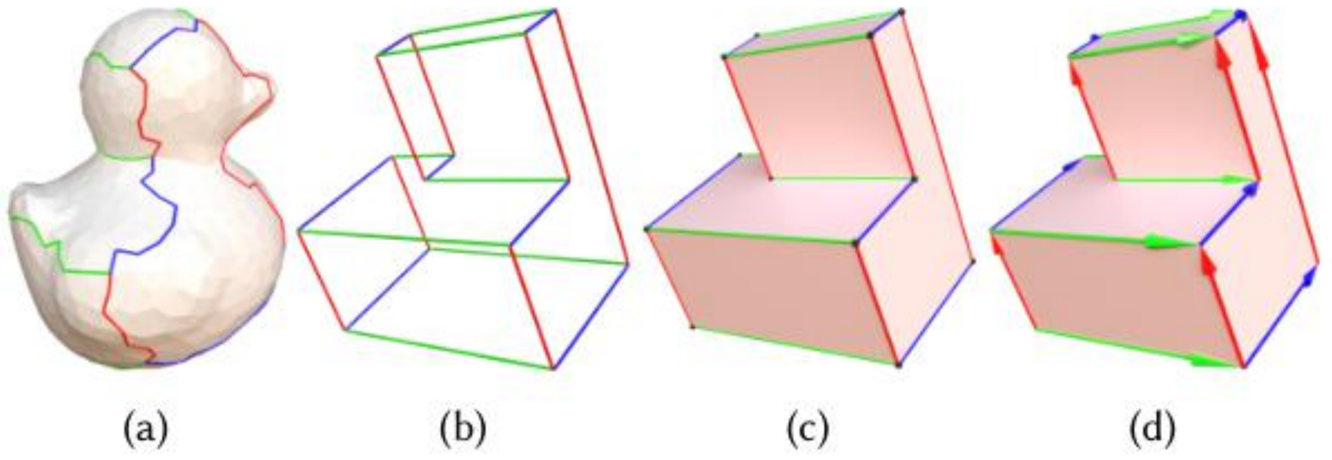


Fig. 10. The colorized arcs are shown in (a), (b), (c), and the initial directions are exhibited in (d).

Theory

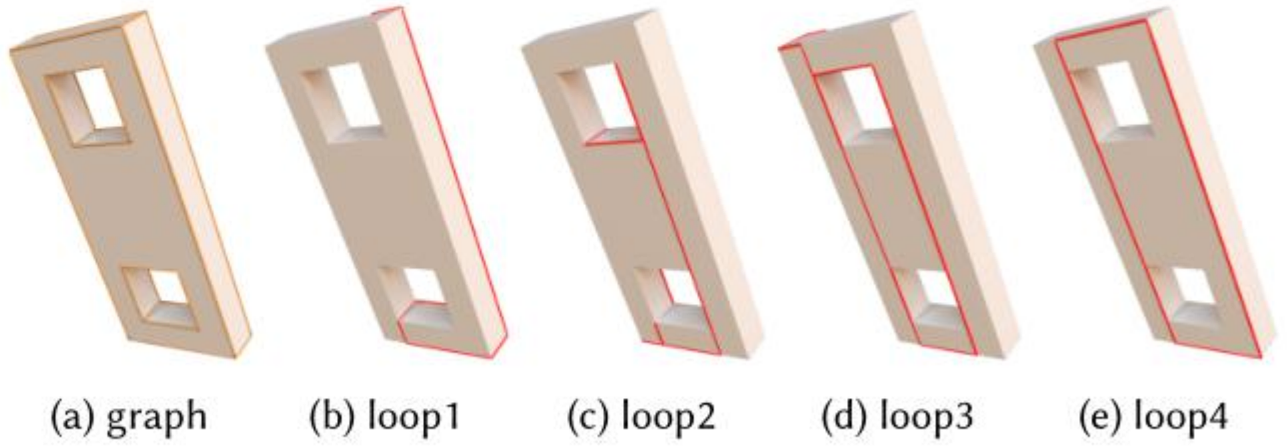


Fig. 12. The homology loops of the "double torus" model.

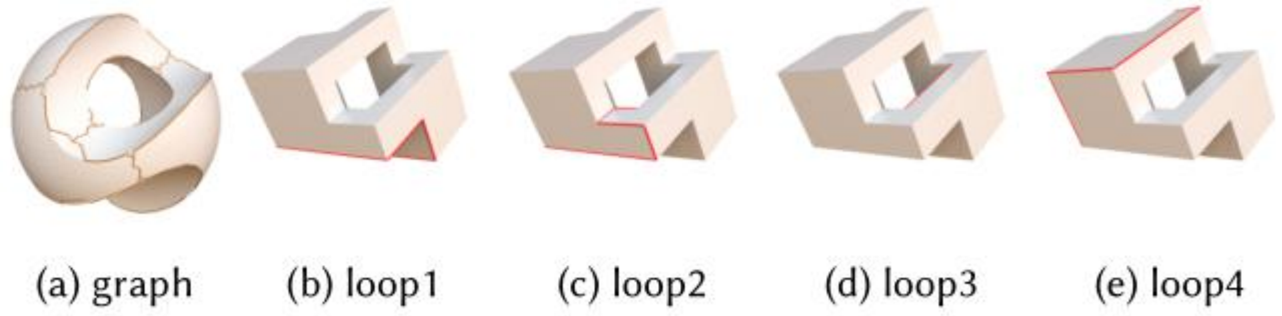


Fig. 13. The homology loops of the "sculpt" model.

Polycube Shape Space

Edge length: $l : A \to \mathbb{R}$

Edge orientation: $d : A \to \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$

Discrete 1-form: $\omega = l * d$

Closed 1-form: $d\omega(p) = \sum_{a \in p} \omega(a)0, \forall p \in P$

Exact 1-form: $\int_\Gamma \omega = \sum_{a \in \Gamma} \omega(a) = 0, \forall \Gamma \in H(G)$

Polycube shape space: The linear space of all possible closed and exact 1-forms under a fixed orientation. **The dimension is |P| - 3.**

Polycube space by exact one-form

1. assign a positive direction “d” for every colorized arc randomly

2. Modify non-simply connected faces into simply ones.

d

3. Define a vector-valued differential one-form by “edge length multiply direction”.

$$\omega = l * d,$$

4. As the polycubes are embedded in R^3 space, therefore this one-form is exact.

$$d\omega = \mathbf{0}.$$

Polycube space by exact one-form

5. Exact one-form integrate to zero on all closed loops, need only satisfy on all homology basis loops

6. Exact one-form conditions can be expressed as a linear system W

$$d\omega(p_j) = \sum_{arc_i \in p_j} (l_i \cdot d_i) = 0.$$

7. The dim of the kernel space P of the linear system W is dim of the polycube shape space

Theory

In matrix format: W is a matrix, L is a vector of lengths

$$W \cdot L = 0$$

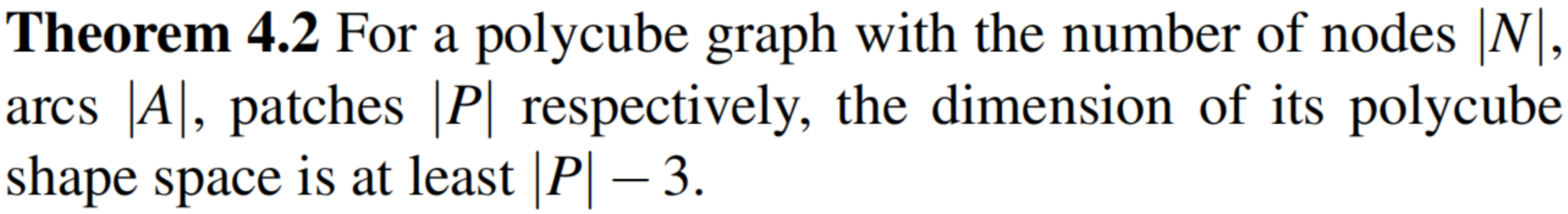
If the dimension of the kernel space *K* of the matrix *W* is zero,

it means that there is no valid *polycube polyhedron*.

Theorem 4.1 If a polycube graph can realize a polycube polyhedron embedded in $\mathbb{R}^3$, then the dimension of the corresponding kernel space K is not zero.

Theory

We proved:



Theorem 4.2 For a polycube graph with the number of nodes $|N|$, arcs $|A|$, patches $|P|$ respectively, the dimension of its polycube shape space is at least $|P|-3$.

Theory

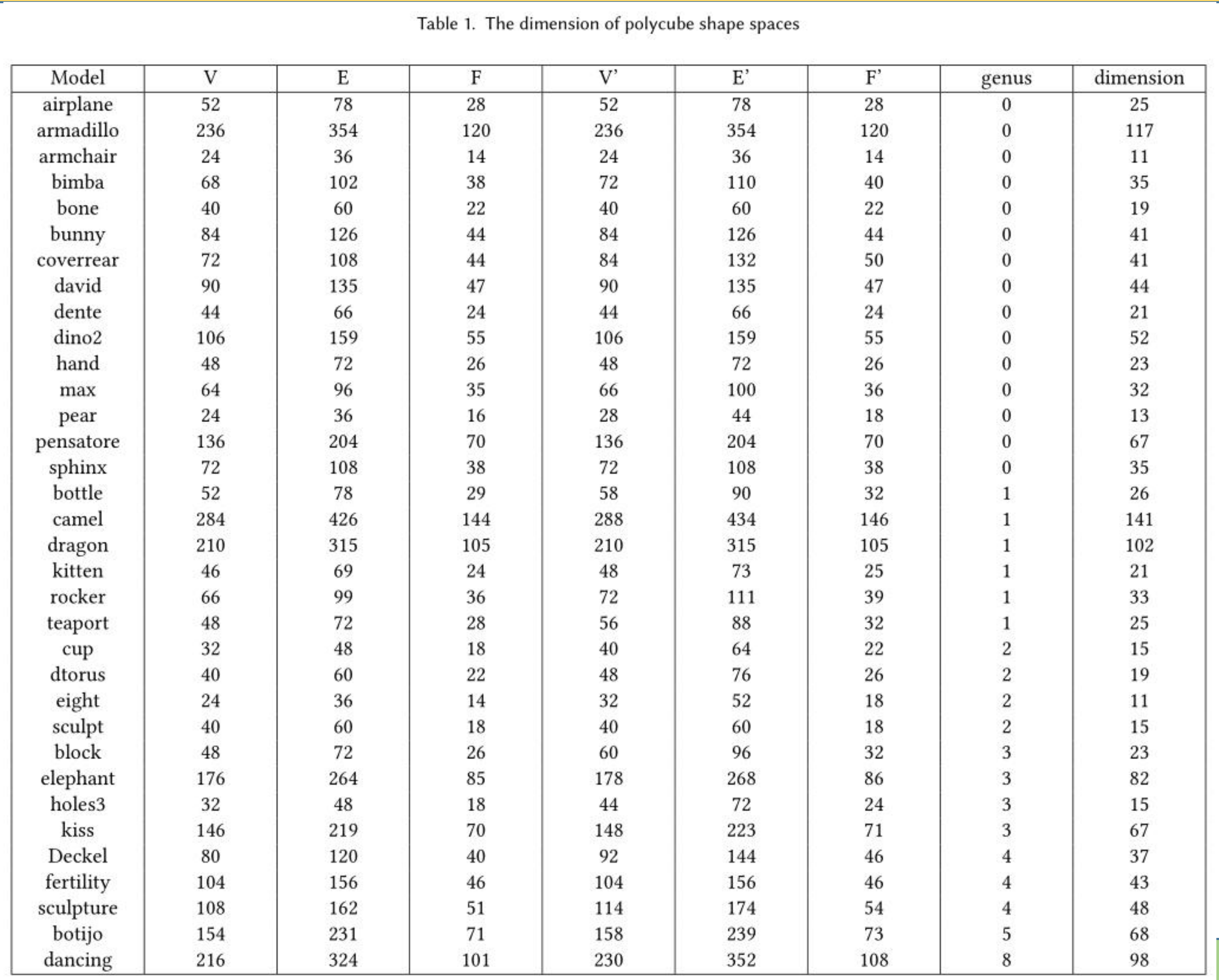


Table 1. The dimension of polycube shape spaces

Model	V	E	F	V'	E'	F'	genus	dimension
airplane	52	78	28	52	78	28	0	25
armadillo	236	354	120	236	354	120	0	117
armchair	24	36	14	24	36	14	0	11
bimba	68	102	38	72	110	40	0	35
bone	40	60	22	40	60	22	0	19
bunny	84	126	44	84	126	44	0	41
coverrear	72	108	44	84	132	50	0	41
david	90	135	47	90	135	47	0	44
dente	44	66	24	44	66	24	0	21
dino2	106	159	55	106	159	55	0	52
hand	48	72	26	48	72	26	0	23
max	64	96	35	66	100	36	0	32
pear	24	36	16	28	44	18	0	13
pensatore	136	204	70	136	204	70	0	67
sphinx	72	108	38	72	108	38	0	35
bottle	52	78	29	58	90	32	1	26
camel	284	426	144	288	434	146	1	141
dragon	210	315	105	210	315	105	1	102
kitten	46	69	24	48	73	25	1	21
rocker	66	99	36	72	111	39	1	33
teaport	48	72	28	56	88	32	1	25
cup	32	48	18	40	64	22	2	15
dtorus	40	60	22	48	76	26	2	19
eight	24	36	14	32	52	18	2	11
sculpt	40	60	18	40	60	18	2	15
block	48	72	26	60	96	32	3	23
elephant	176	264	85	178	268	86	3	82
holes3	32	48	18	44	72	24	3	15
kiss	146	219	70	148	223	71	3	67
Deckel	80	120	40	92	144	46	4	37
fertility	104	156	46	104	156	46	4	43
sculpture	108	162	51	114	174	54	4	48
botijo	154	231	71	158	239	73	5	68
dancing	216	324	101	230	352	108	8	98

Polycube Embedding

Optimal Embedding from the polycube shape space

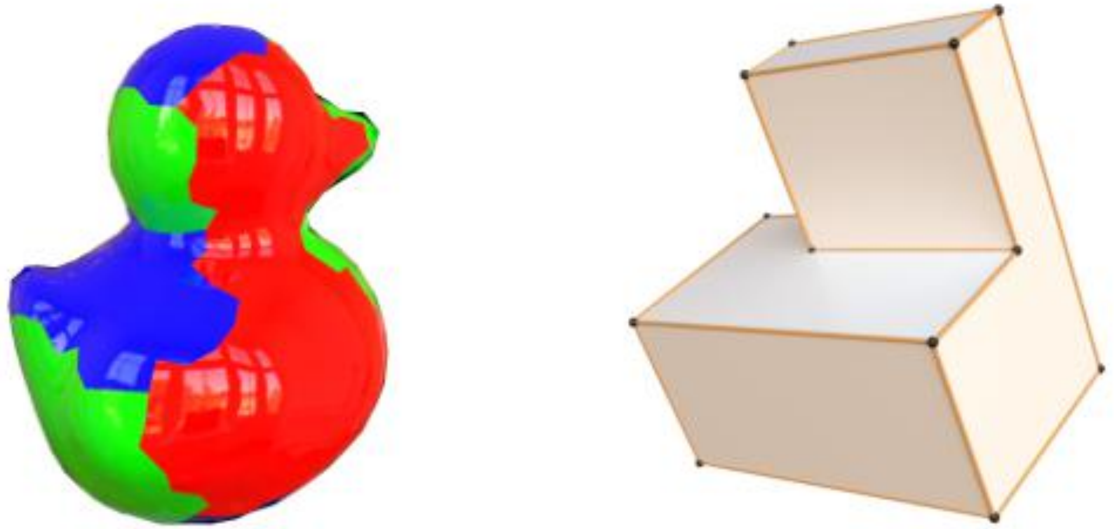


Fig. 15. The "duck" model and its optimal polycube polyhedron.

We look for an optimal embedding whose node positions approximate the input layout graph's node positions.

Theory

Algorithm:

1. a quadratic optimization method to obtain an optimal polycube polyhedron from this space.

2. employ linear inequality constraints to remove the degenerate cases.

3. add constraints on edge lengths to adjust the shape of the polycube polyhedron.

Optimal Embedding from polycube shape space

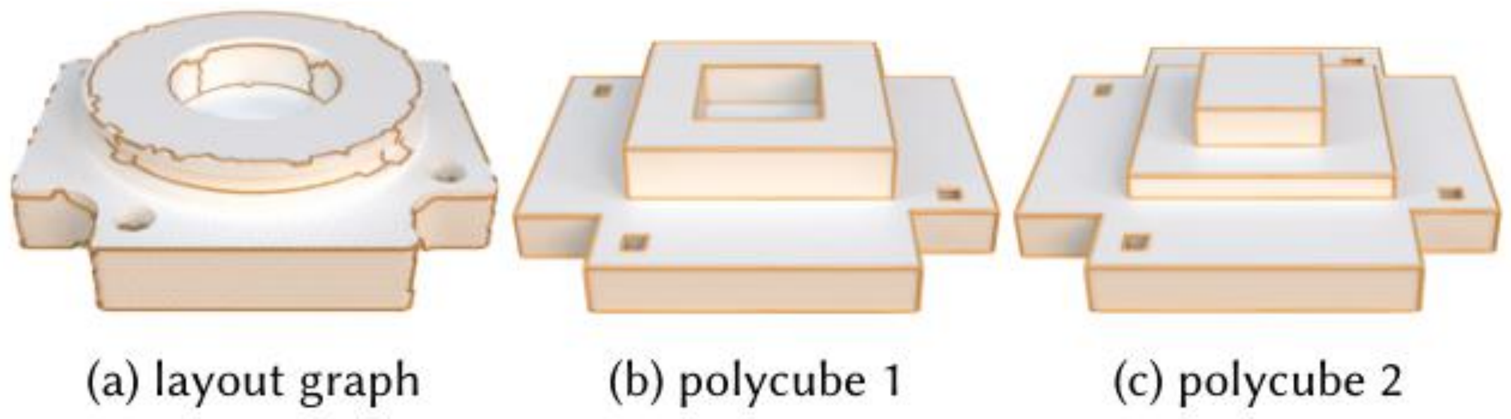


Fig. 6. Two different polycubes (generated by our algorithms) corresponding to the same layout graph.

Embedding of every basis one-form

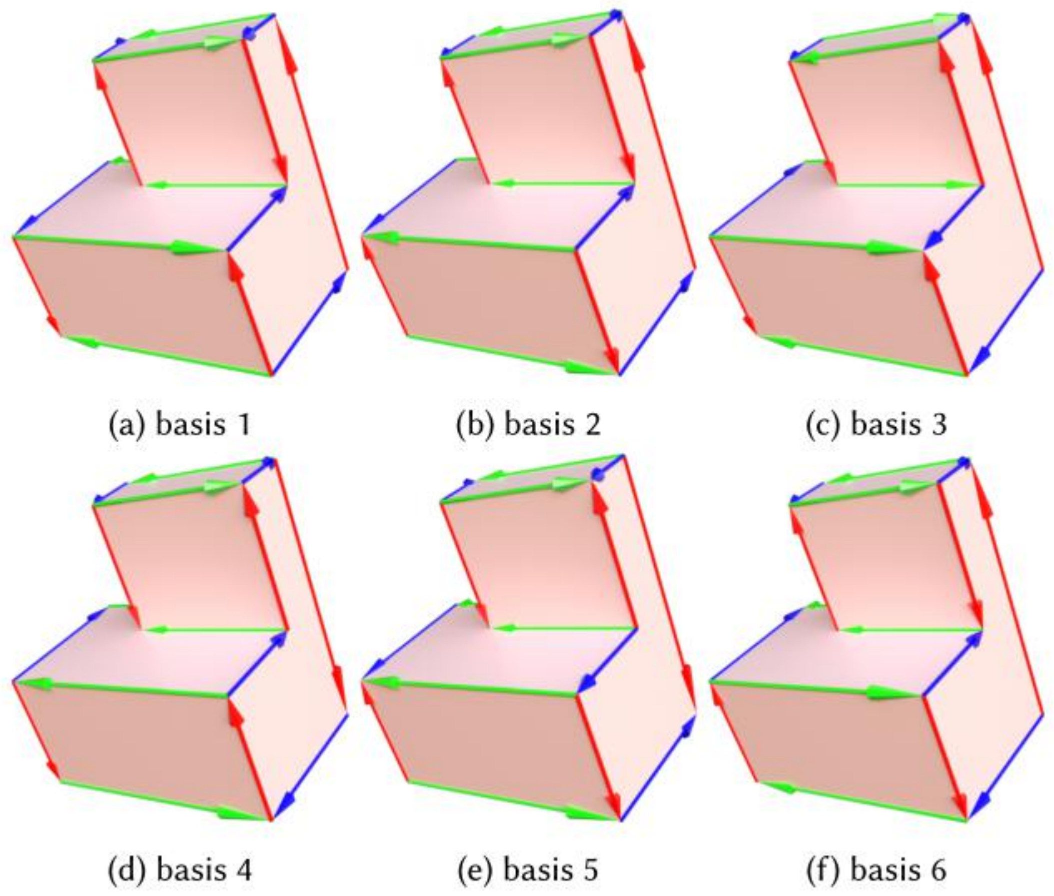


Fig. 14. The basis of the polycube shape space for the "duck" mesh and its layout graph.

Optimal Embedding from polycube shape space

$$\omega = \sum_{\omega^i \in N} c_i \cdot \omega^i,$$

$$S(v) = \sum_{\omega^i \in N} c_i \omega_0^i(v) = \sum_{\omega^i \in N} c_i \int_{v_0}^{v} \omega^i + \mathbf{r}_0,$$

$$(S - \bar{S})^2 = (\sum_{\omega^i \in N} c_i \cdot \int \omega^i + \mathbf{r}_0 - \bar{S})^2$$

Embedding

Well-defined 0-form on vertices: $\omega_0(v) = \int_{\gamma(v_0,v)} \omega + \mathbf{r}_0$

This represents the coordinates of an embedding.

Assume $N = \{\omega^i\}$ is the basis of the polycube space, all possible embeddings is represented by

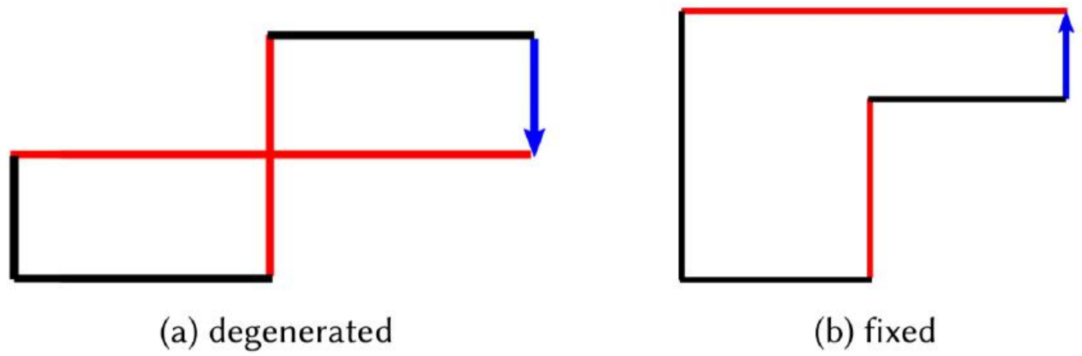
$$S(v) = \sum_{\omega^i \in N} c_i \int_{v_0}^{v} \omega^i + \mathbf{r}_0$$

Quadratic programming

Coordinates on the original mesh

$$(S - \bar{S})^2 = \Big(\sum_{\omega^i \in N} c_i \cdot \int \omega^i + \mathbf{r}_0 - \bar{S} \Big)^2$$

$$\left.\begin{array}{l} l_i \geq \delta, \ or \\ l_i \leq -\delta \ or \\ (\omega_0(v_i) - \omega_0(v_j)) \cdot \mathbf{n}_i \geq \delta, \ or \\ (\omega_0(v_i) - \omega_0(v_j)) \cdot \mathbf{n}_i \leq -\delta \end{array}\right\} \longrightarrow$$


(a) degenerated (b) fixed

Edge Length constraints

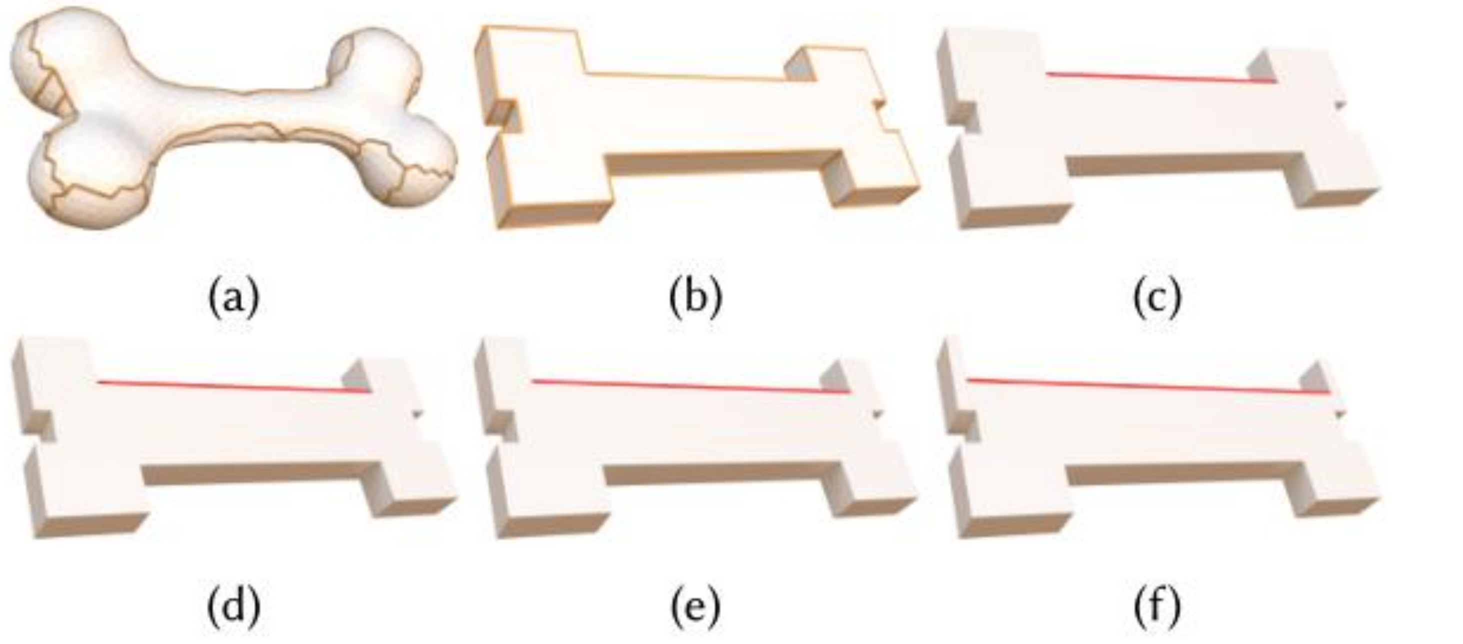


Fig. 16. The polycube embeddings with the different edge length constraints.

Degenerated cases

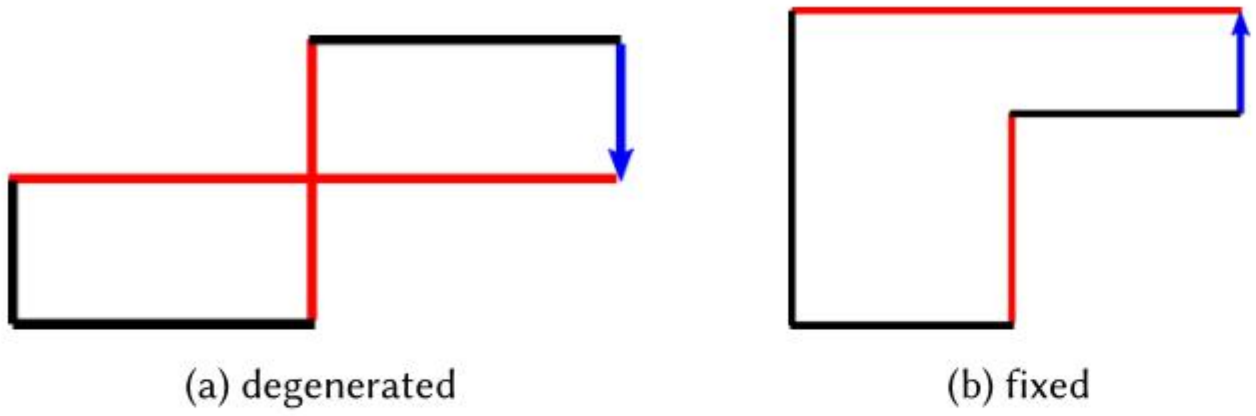


Fig. 17. The degenerated cases.

Quadratic programming

$$(S-\bar{S})^2 = (\sum_{\omega^i \in N} c_i \cdot \int \omega^i + \mathbf{r}_0 - \bar{S})^2$$

$$l_i \geq \delta, \ \ or$$

$$l_i \leq -\delta \ or$$

$$(\omega_0(v_i) - \omega_0(v_j)) \cdot \mathbf{n}_i \geq \delta, \ \ or$$

$$(\omega_0(v_i) - \omega_0(v_j)) \cdot \mathbf{n}_i \leq -\delta$$

where δ is some positive real number.

The inequality symbol is unknown, we use an interactive user interface to setup the inequality value and update the degenerated polycube to a valid one.

Degenerated cases

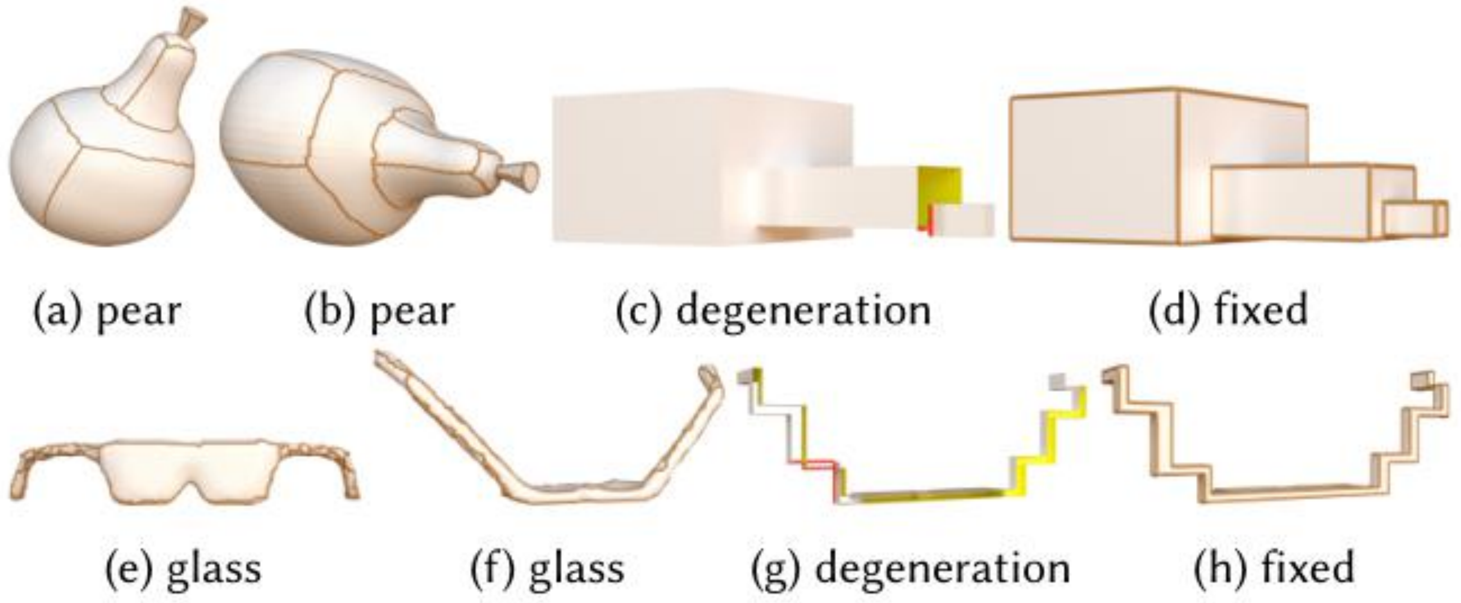


Fig. 18. The "pear" (a), (b) and "glass" (e), (f) models with two different views, and the degenerated polycubes (c), (g) and the fixed polycubes (d),(h).