University of Cagliari
Ph.D. Program in Mathematics and Computer Science
Computer Science Track

Dept. of Mathematics
and Computer Science

Polycubes Optimization and Applications

From the Digital World to Manufacturing

Candidate
Gianmarco Cherchi

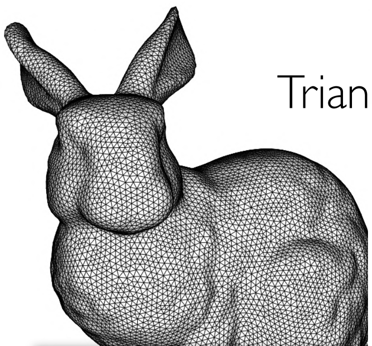
Supervisor
Riccardo Scateni

February 27th, 2019

Meshes and Polycubes

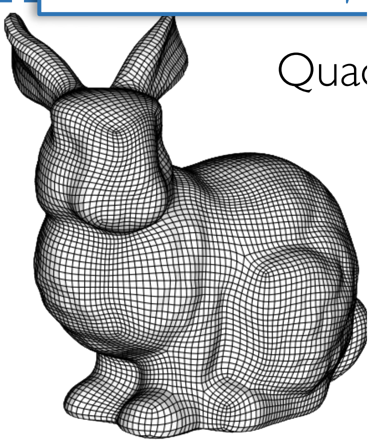
Meshes

Surface mes

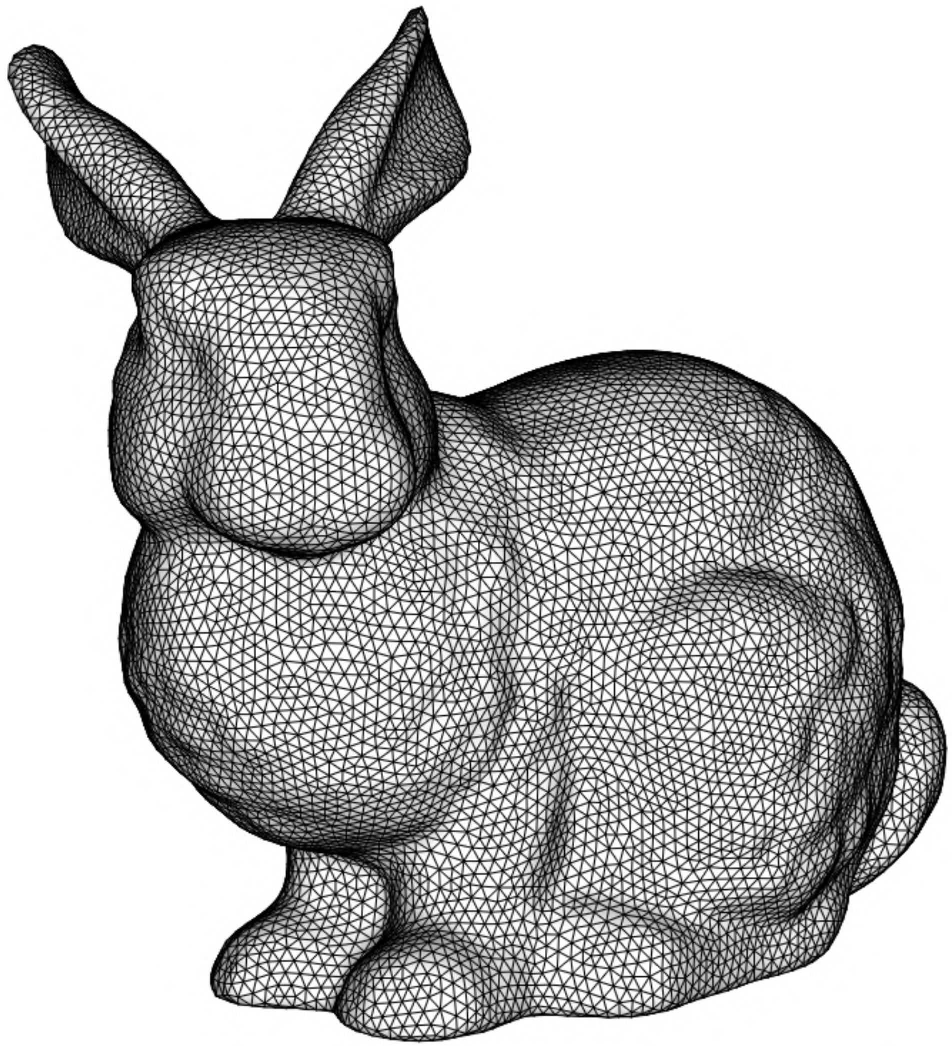


Trian

animation
game develop

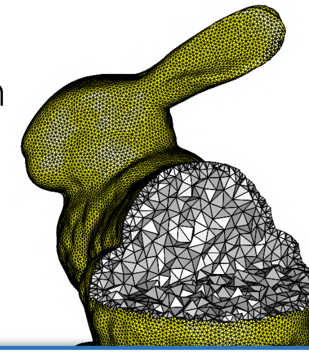


Quad



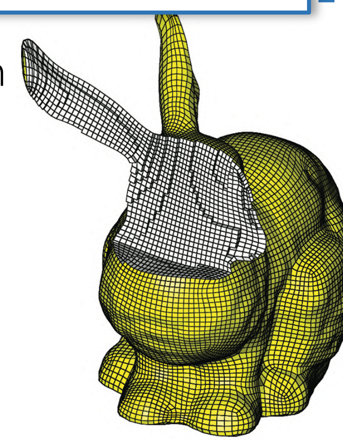
ne meshes

sh



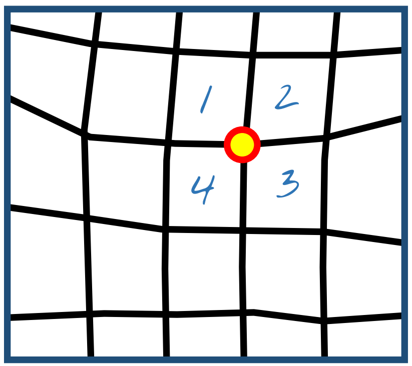
sics and
al simulation

sh

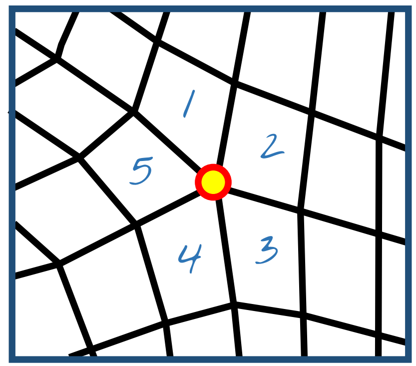


Mesh singularities

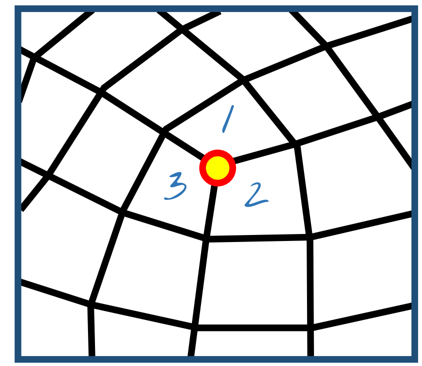
Quad-mesh example



regular vertex (val. 4)



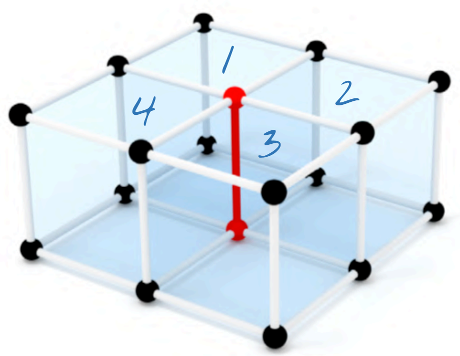
singular vertex (val. 5)



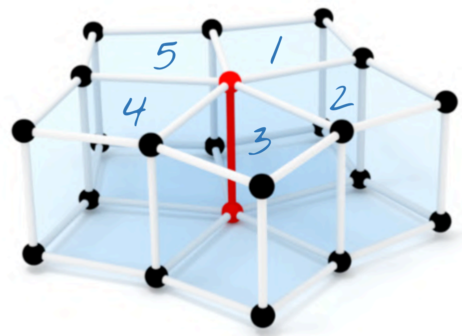
singular vertex (val. 3)

Hex-mesh example

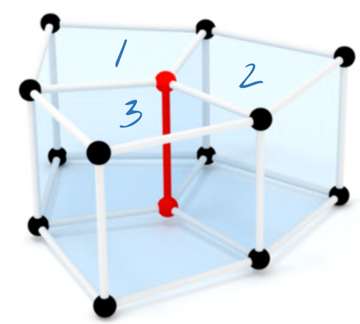
[Liu et al, 2018]



regular edge (val. 4)



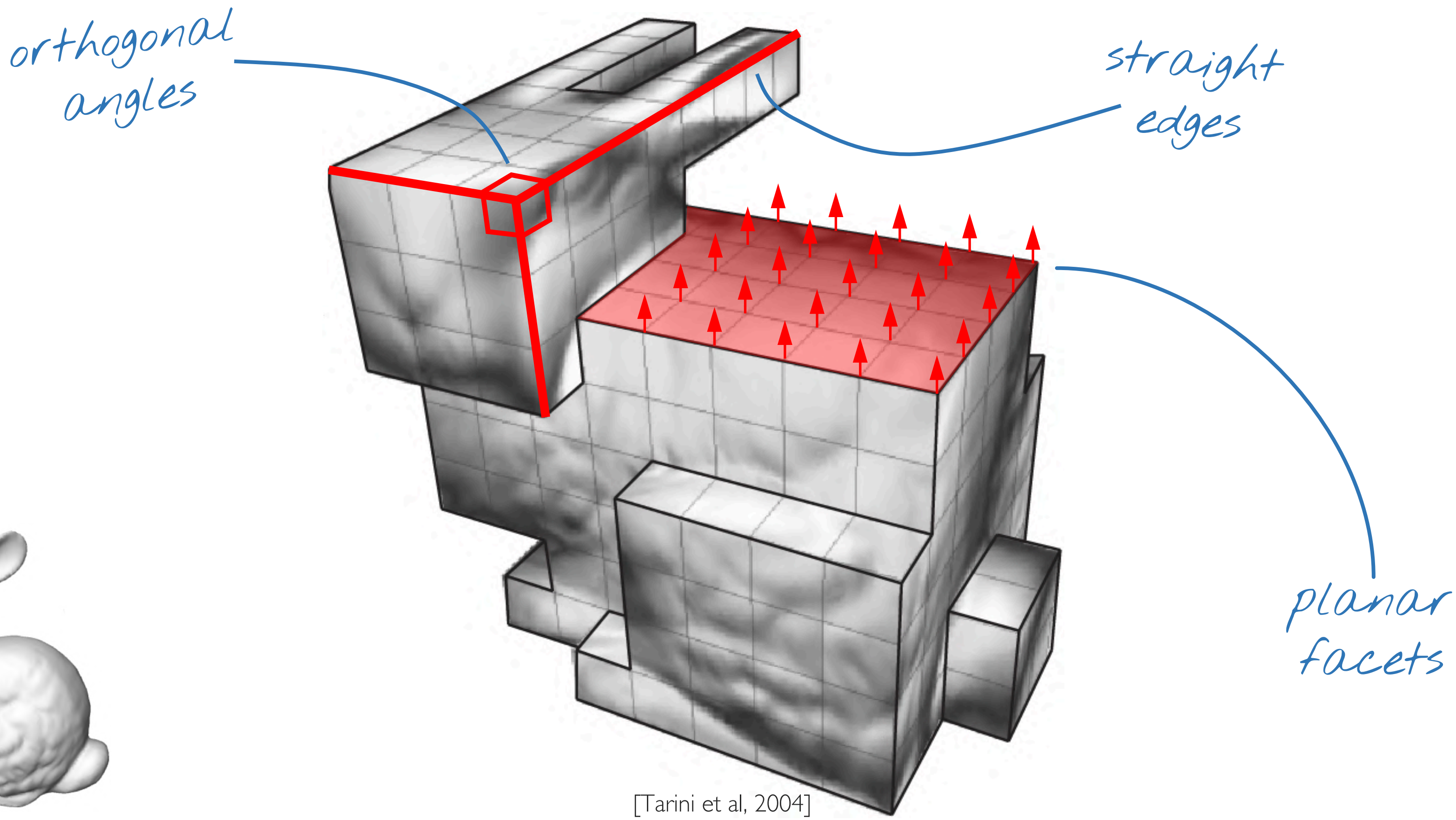
singular edge (val. 5)



singular edge (val. 3)

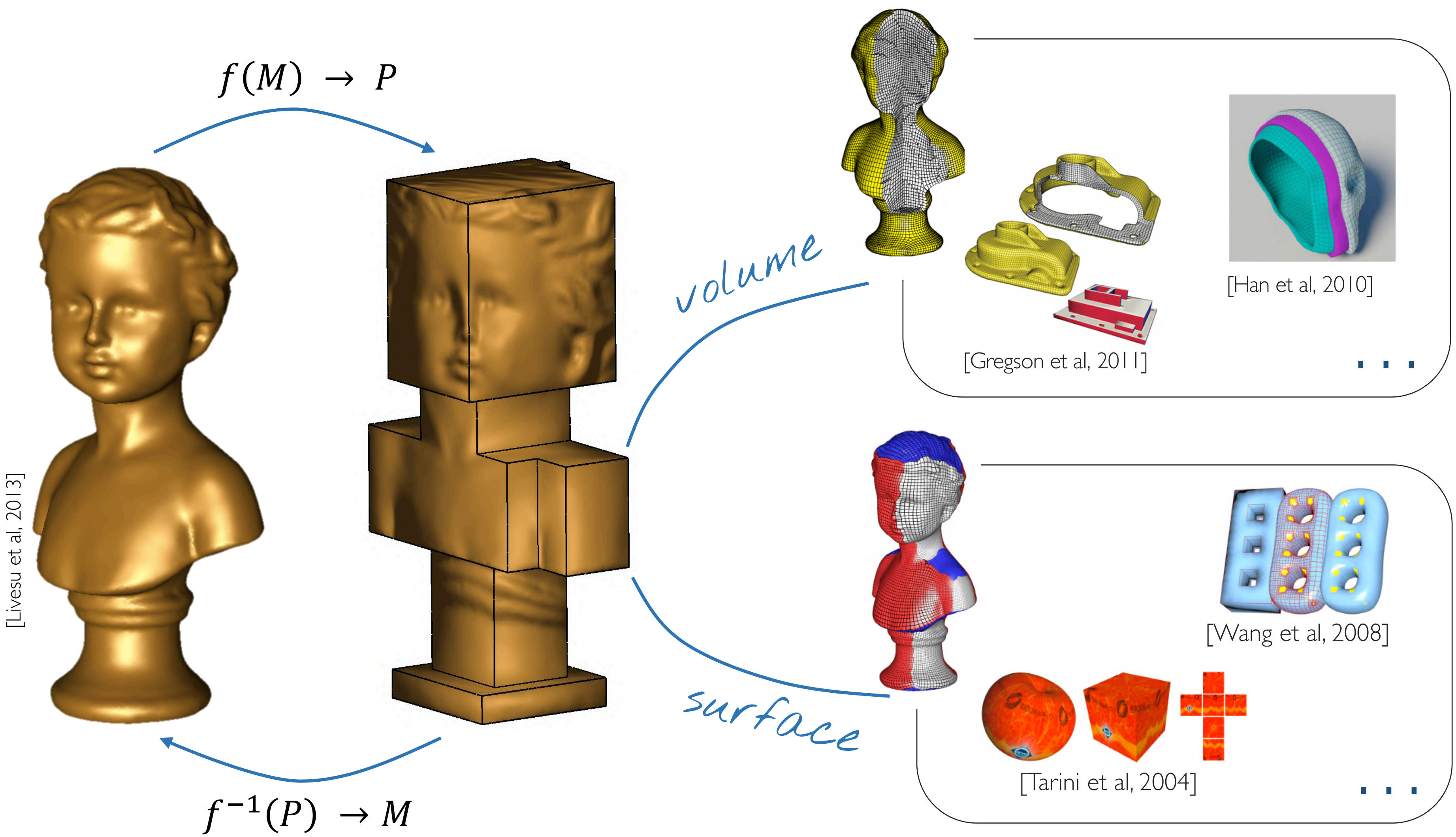
Polycubes

A polycube is a very simple representation (orthogonal polyhedra) of a tridimensional model, made up of a set of connected cuboids.

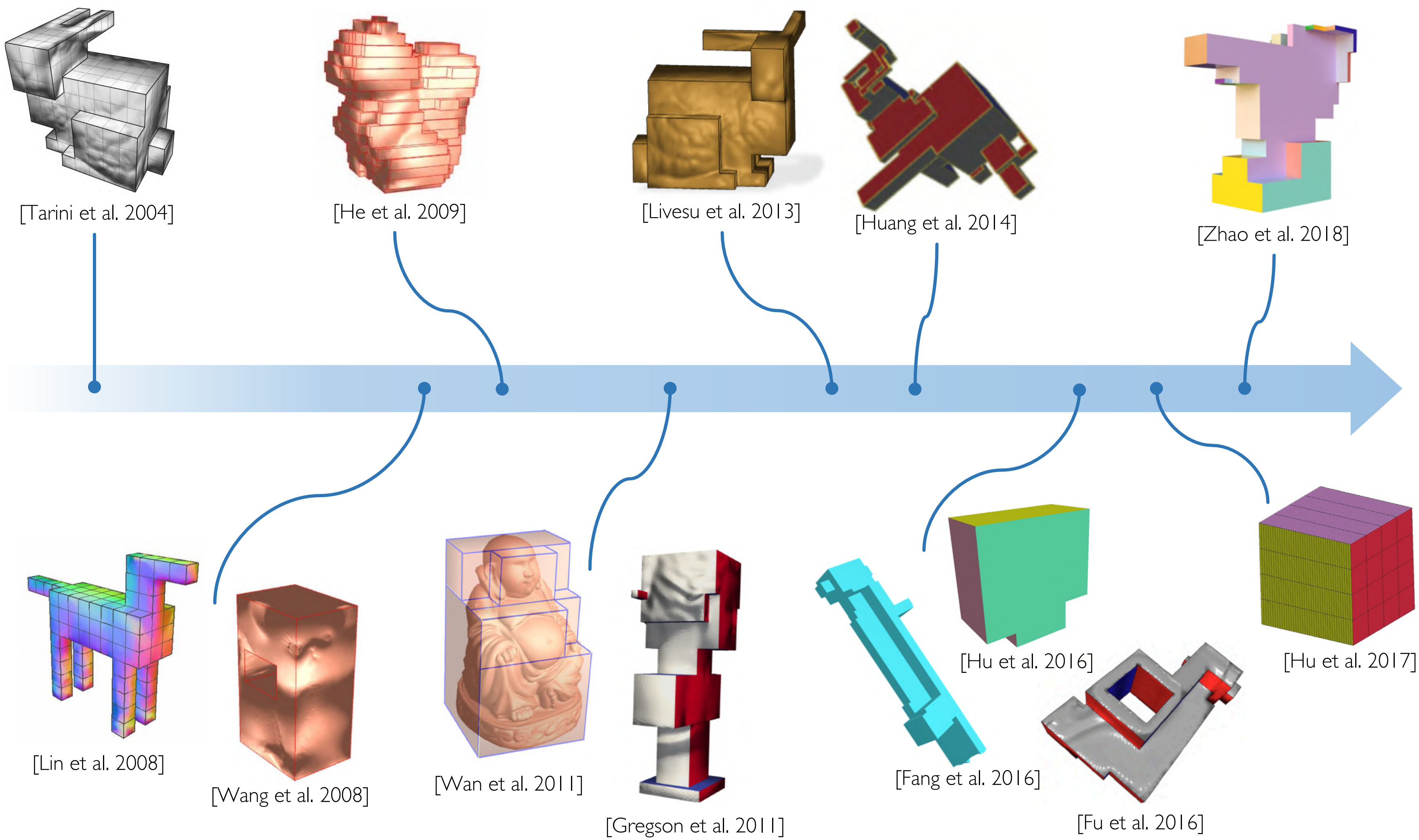


[Tarini et al, 2004]

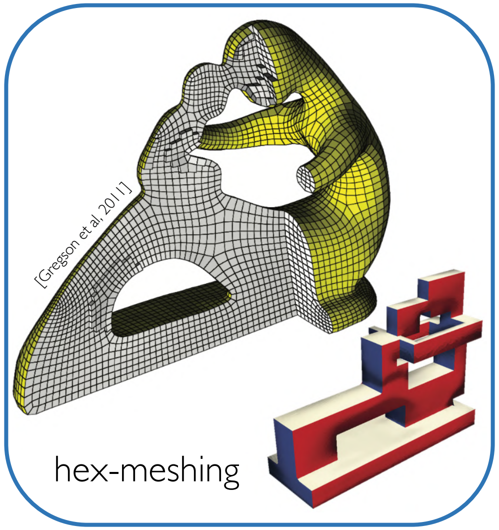
Polycubes

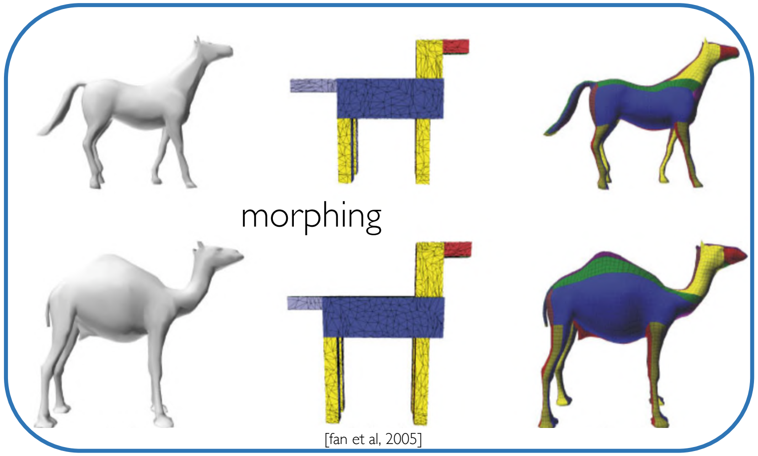

$f(M) \rightarrow P$
$f^{-1}(P) \rightarrow M$
[Livesu et al, 2013]
volume
surface
[Gregson et al, 2011]
[Han et al, 2010]
. . .
[Wang et al, 2008]
[Tarini et al, 2004]
. . .

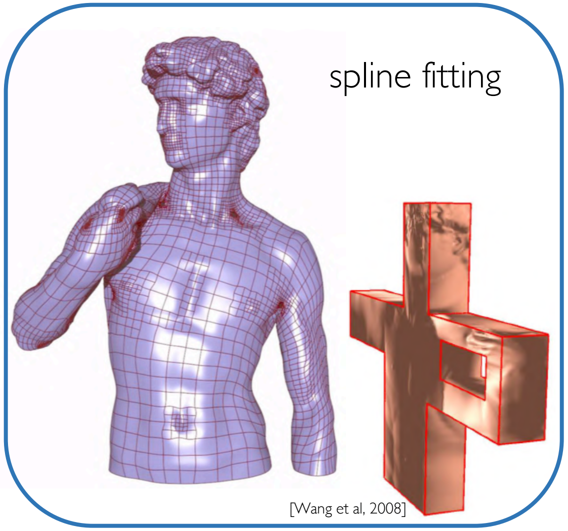
Polycubes – State of the art

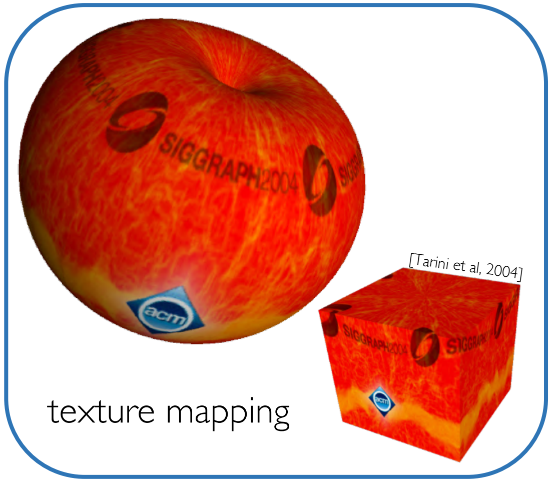


Why polycubes?

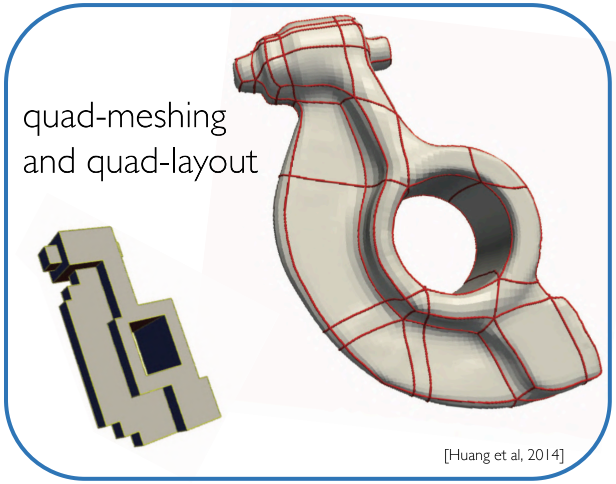








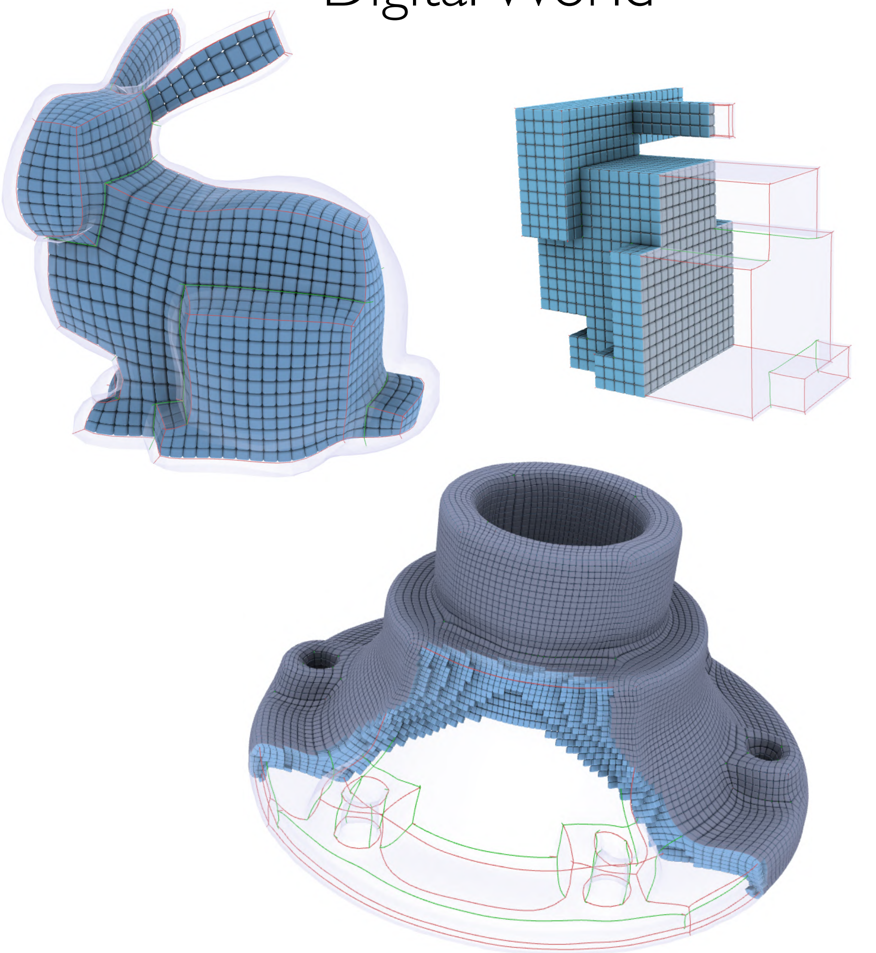
and others...



Polycubes Optimization and Applications

From the digital world to manufacturing

Digital World



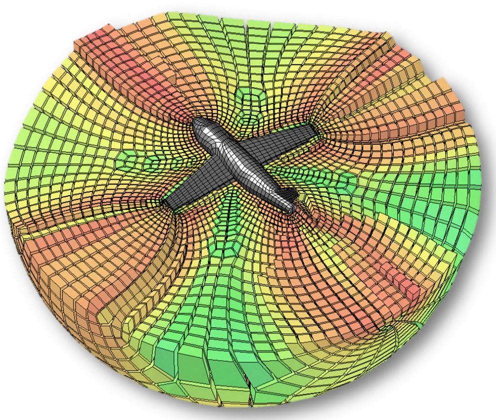
Fabrication

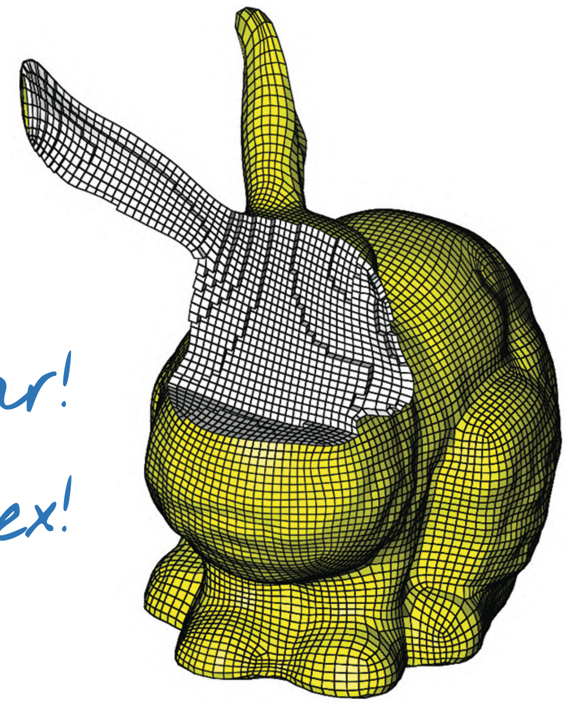
Polycubes for Hex-meshing

What is a hex-mesh?

- A hexahedral mesh is a volumetric mesh where each element is a **hexahedron**
- The union of all elements is the desired volume

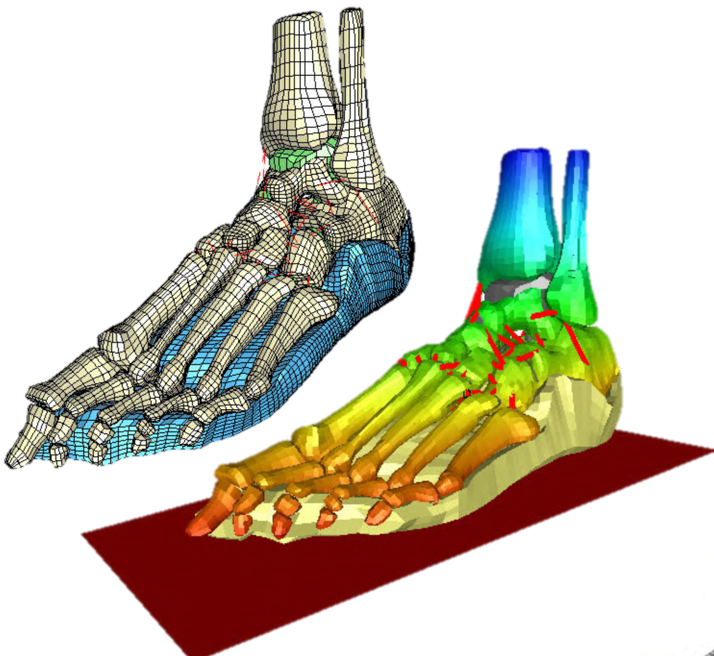
may not be planar!
may not be convex!



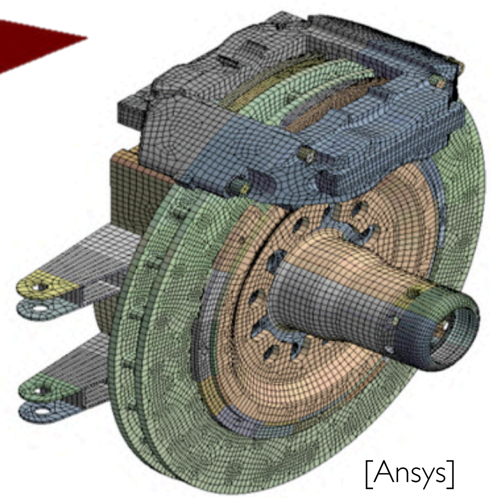


Why hex-meshes?

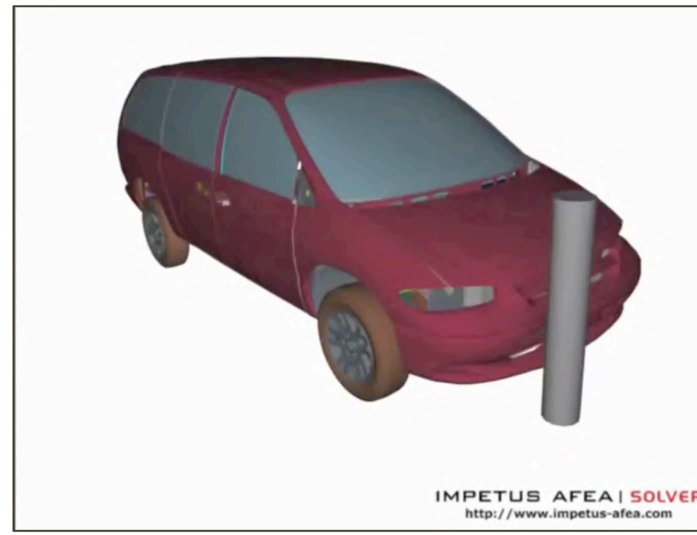
Simulations via Finite Elements Methods



[TrueGrid]



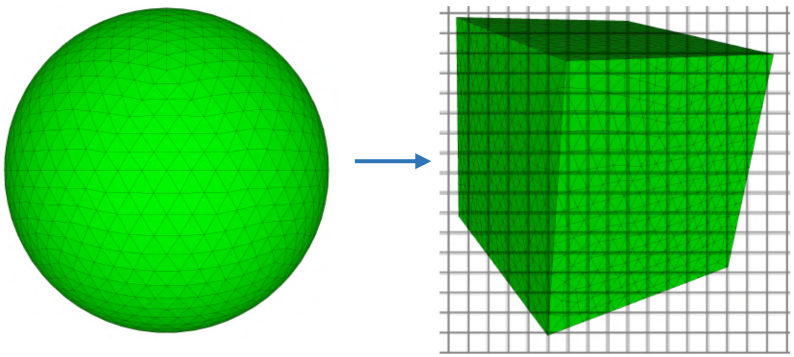
[Ansys]



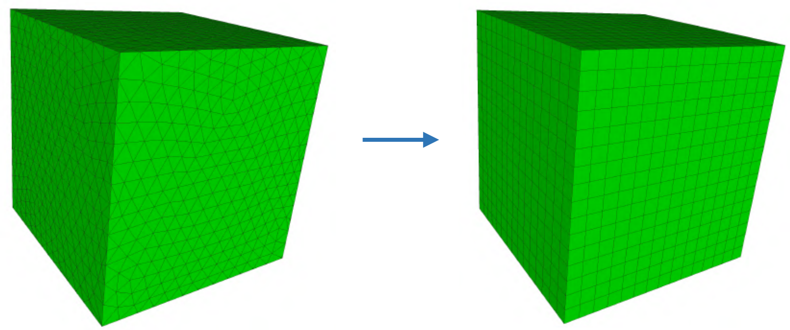
animation↑

Polycube-based meshing pipeline

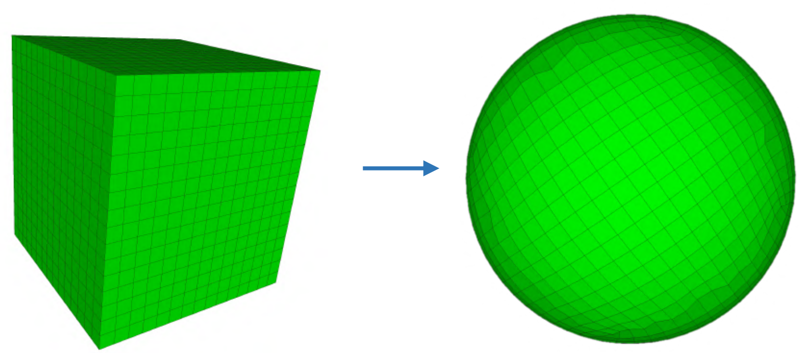
- We **map** the volume of the shape to the **polycube space**, where the generation of the hex-mesh is easier



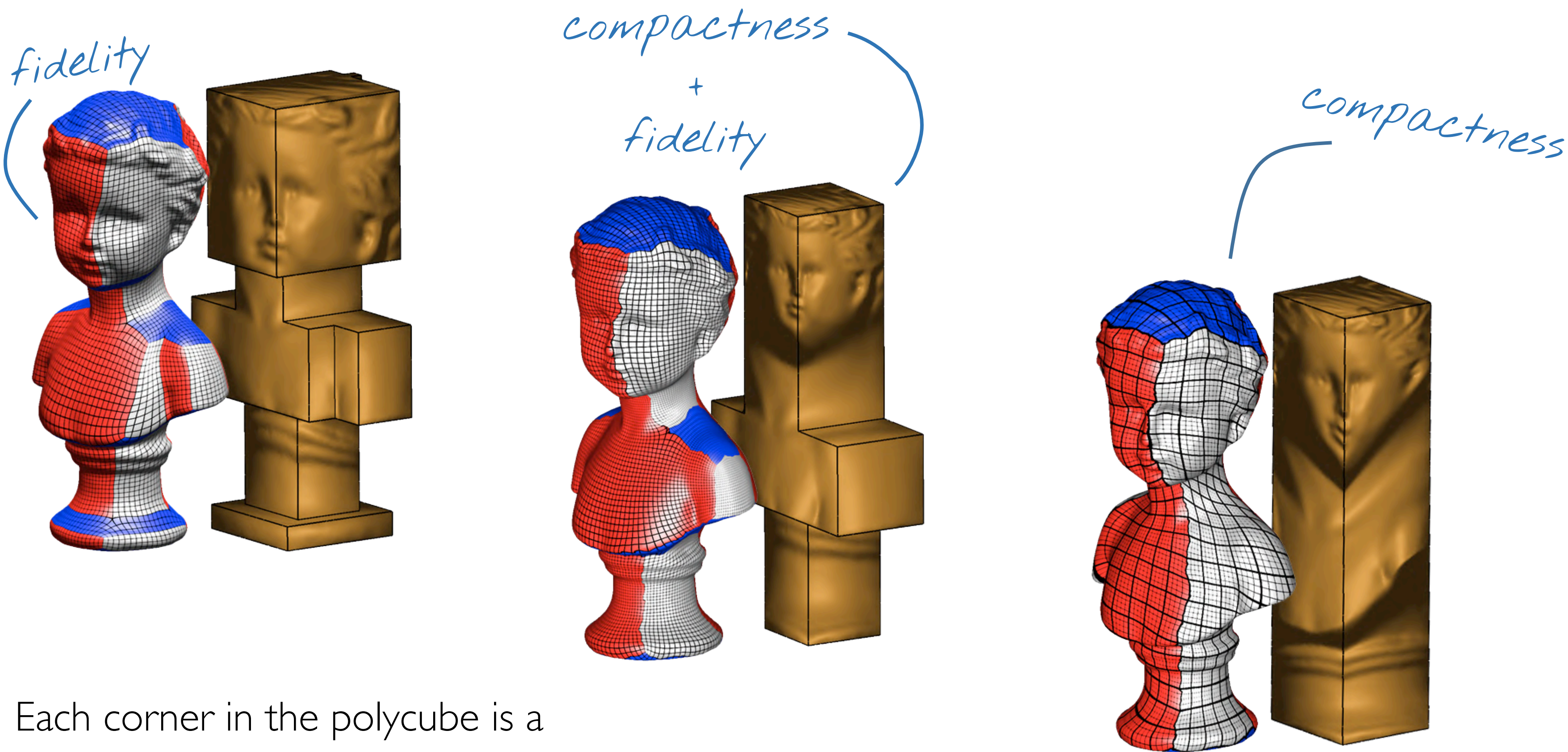
- We define the **mesh structure** in such space



- We use the **inverse mapping** to bring the hex-mesh back to $\mathbb{R}^3$



Polycube structure → hex-mesh structure



- Each corner in the polycube is a singularity in the final mesh

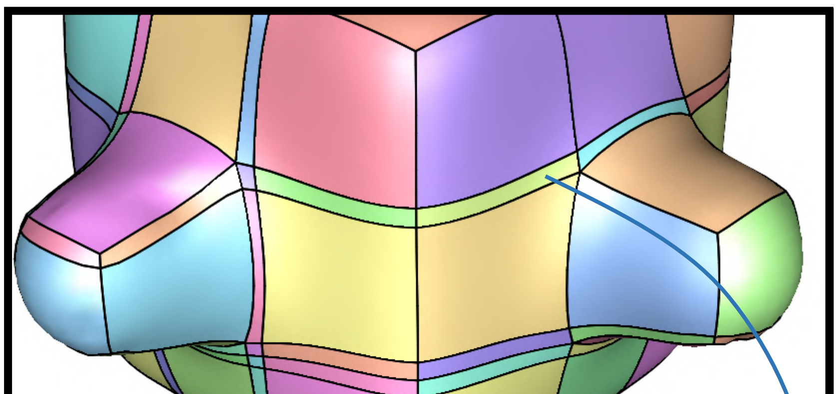
- The polycube compactness influences the mapping distortion

Contribution #1: Polycube Optimization and Corner Alignment

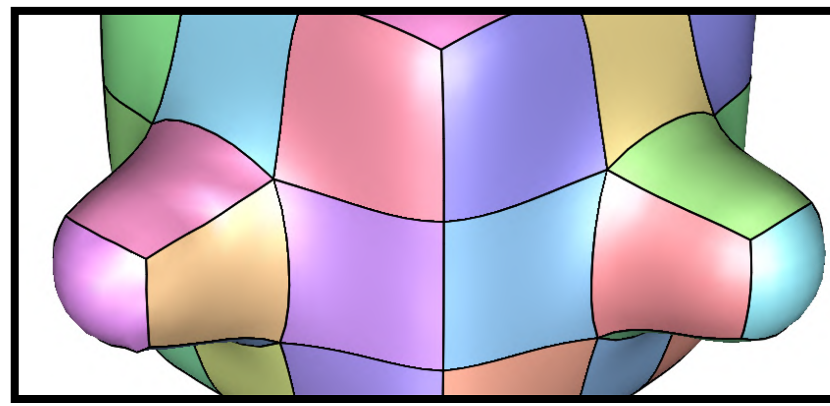
The singularity misalignment problem

Meshes **with** singularity misalignments → "Poor" structure

Meshes **without** singularity misalignments → "Good" structure



the "nearly miss" problem

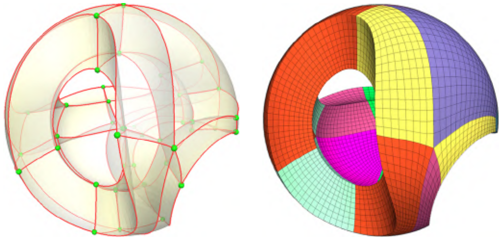


aligned singularities

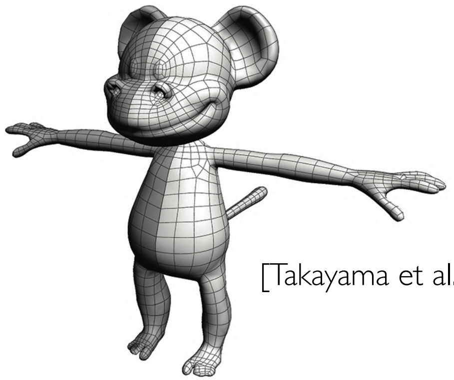
separatrices

Singularity alignment

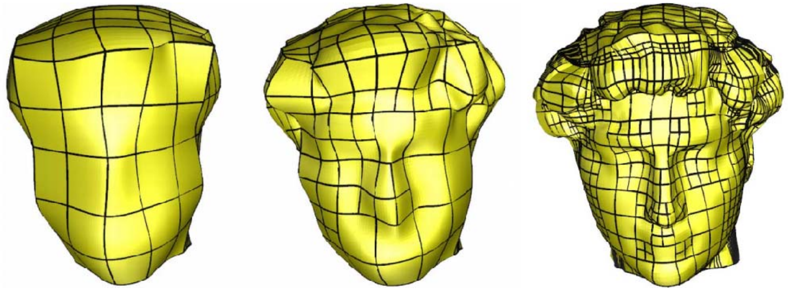
- Having a good singularity alignment is important in a number of applications:
  - High quality hex-meshes for simulation
  - High quality quad-meshes for animation
  - Higher order-meshing
  - Benefits for memory requirements
  - Benefits for performance speedup



[Gao et al. 2015]



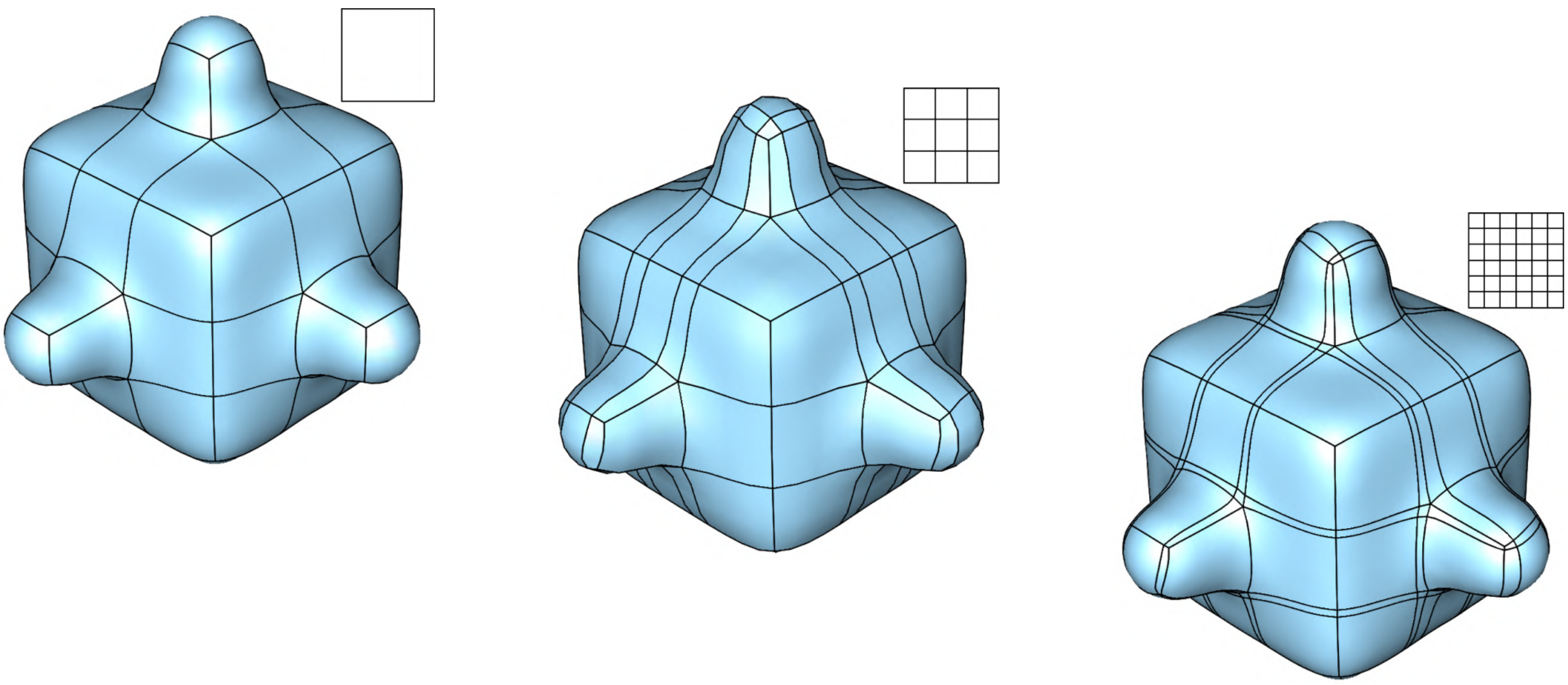
[Takayama et al. 2013]



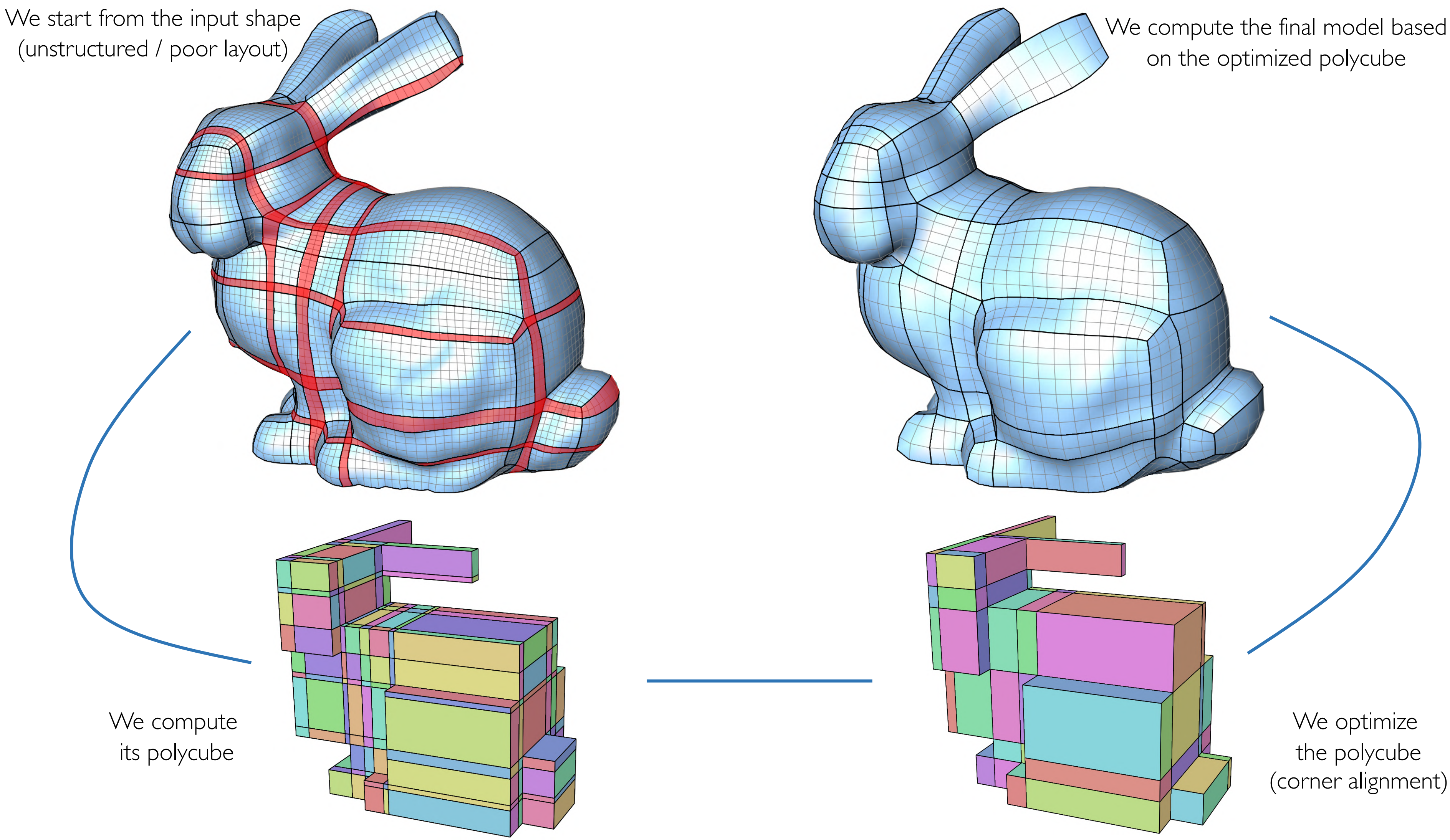
[Li et al. 2013]

Another problem to solve

- Polycube in an integer lattice
- Different lattice densities generate different mesh structures



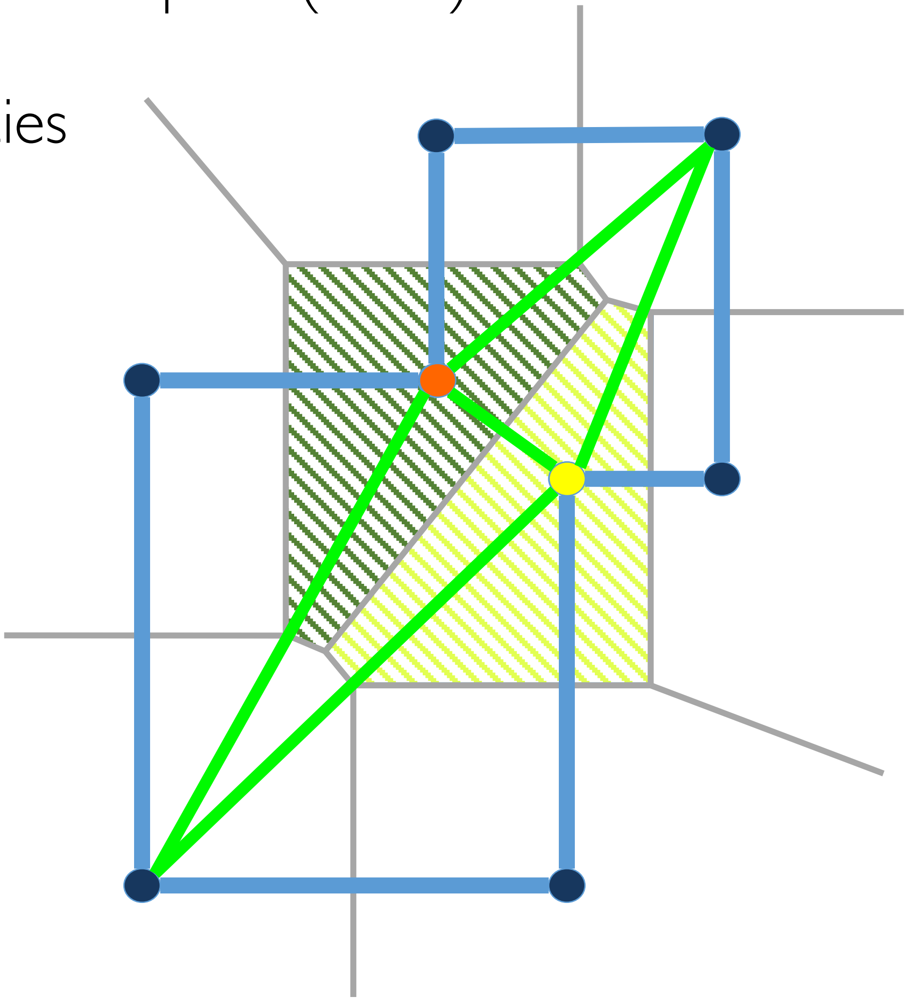
Idea: Alignment in the polycube space



Corner pairing

Voronoi based heuristic to find corner pairs ($\mathbf{A}$ set)

Pruning of the graph of adjacencies



Corner alignment

A mathematical model with integer variables

$$\min E = \boxed{E_{align}(A)} + \boxed{\lambda} \cdot \boxed{E_{shape}}$$

$$s.t.$$

$$\boxed{structural\ constraints}$$

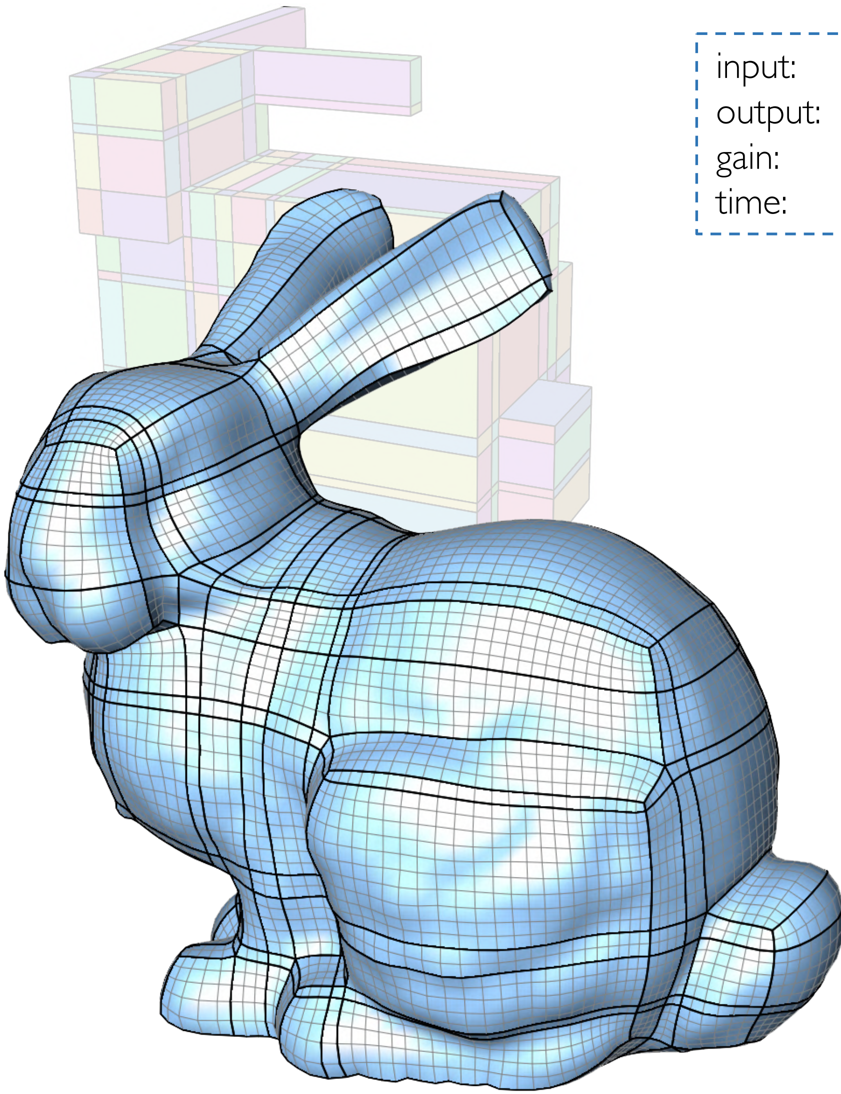
$$\boxed{E_{align} = \sum_{(c,c')\in A_x} (c_x - c'_x)^2 + \sum_{(c,c')\in A_y} (c_y - c'_y)^2 + \sum_{(c,c')\in A_z} (c_z - c'_z)^2}$$

$$\boxed{E_{shape} = \sum_{c} \|c - \tilde{c}\|^2}$$

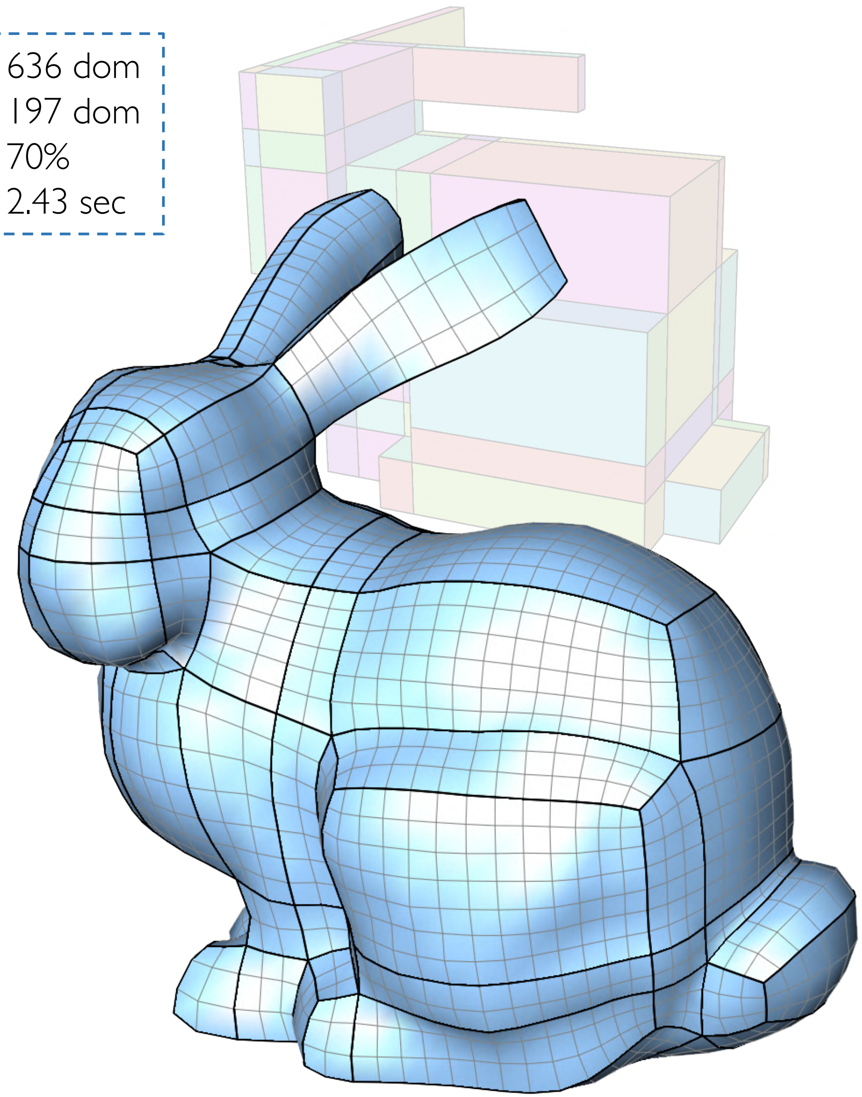
$\boxed{\lambda = trade\text{-}off\ factor}$

- Collinearity of end-points
- Avoid edge collapse (length >= 1)
- Avoid corner collapse
- Dummy vertices and edges

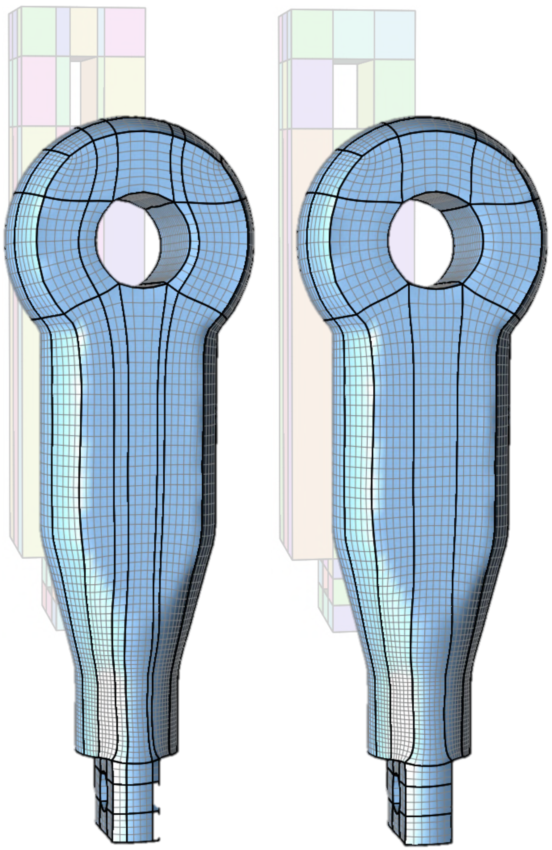
Results



input:	636 dom
output:	197 dom
gain:	70%
time:	2.43 sec

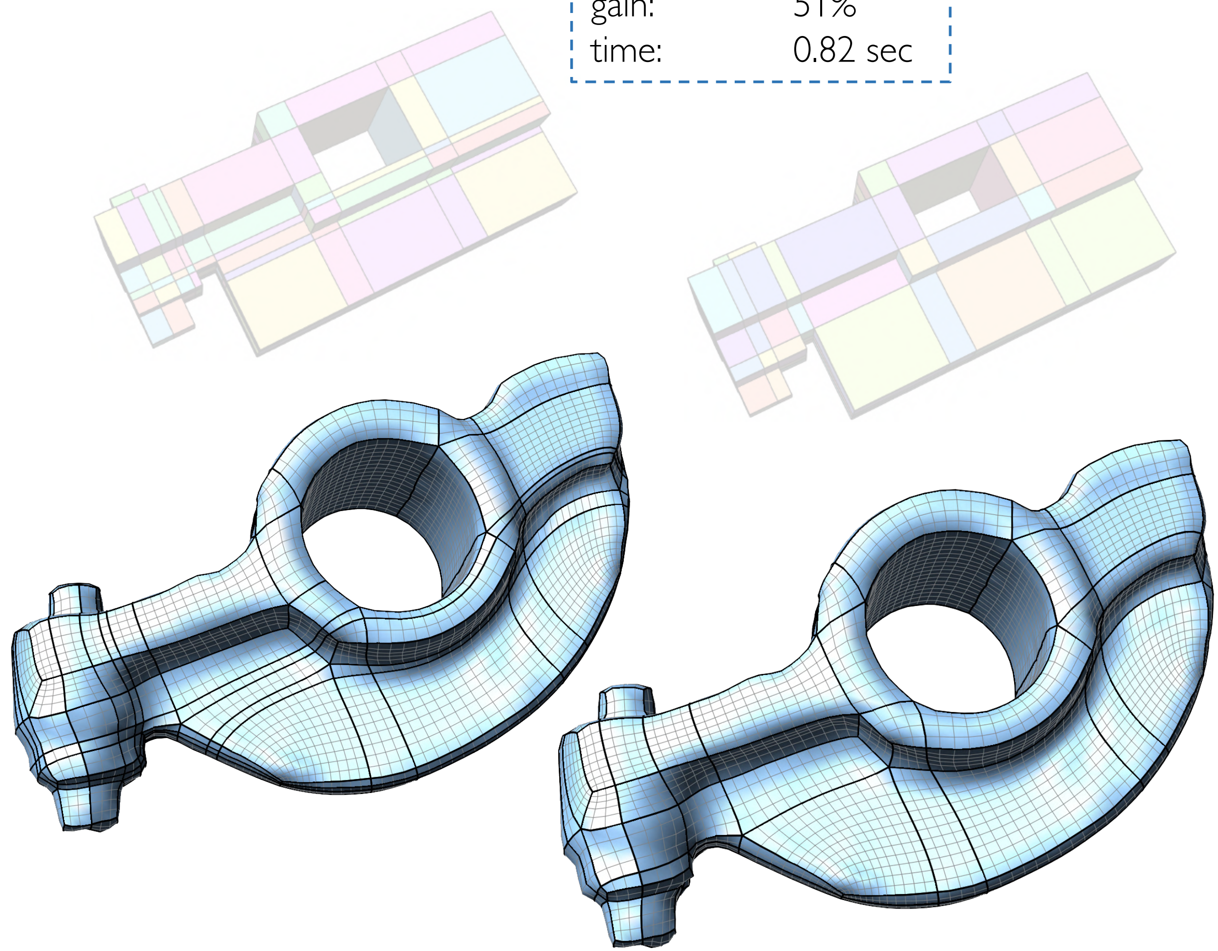


Results

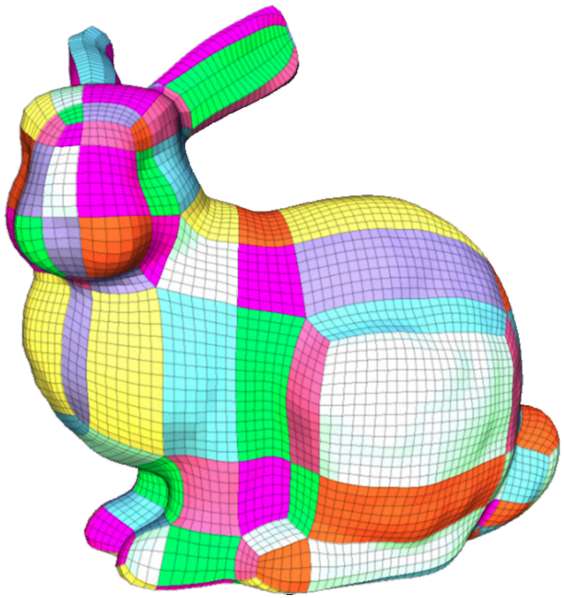


input:	184 dom
output:	114 dom
gain:	38%
time:	1.33 sec

input:	684 dom
output:	332 dom
gain:	51%
time:	0.82 sec

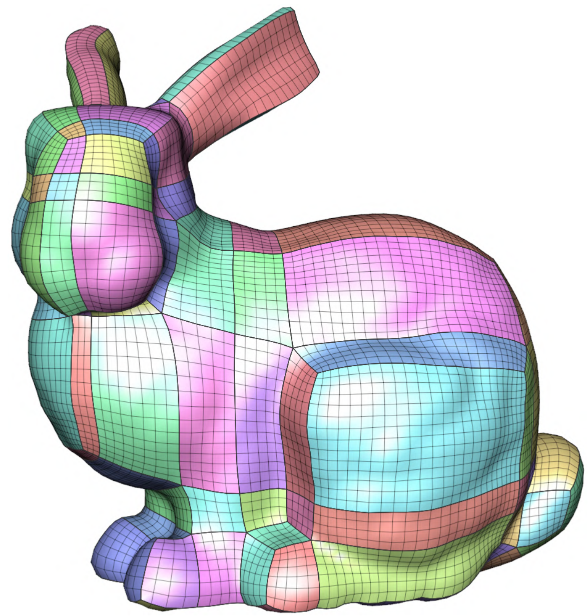


Comparison



[Gao et al. 2015]

input:	580 dom
output:	194 dom
gain:	67%
time:	from 1 m to ½ h



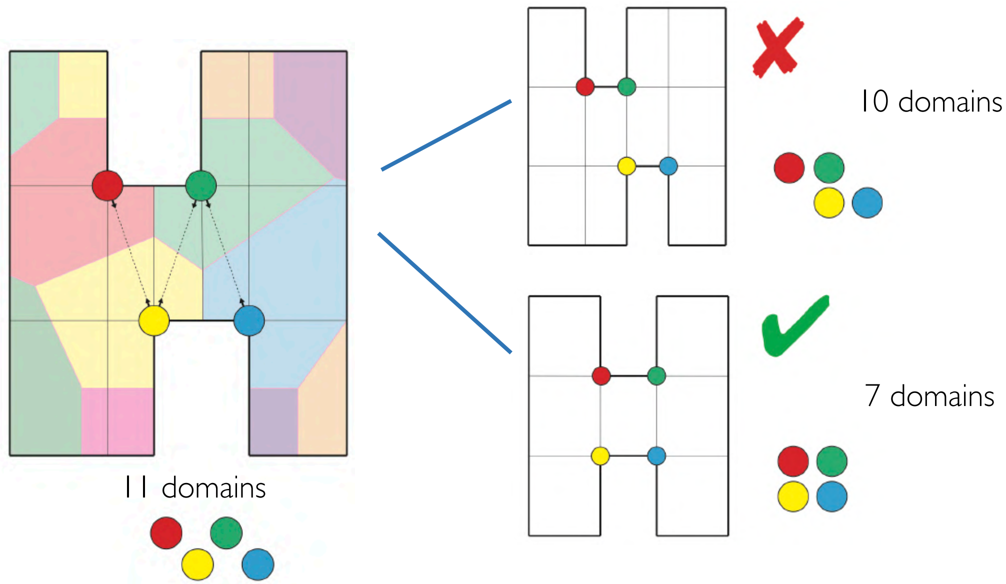
[Ours]

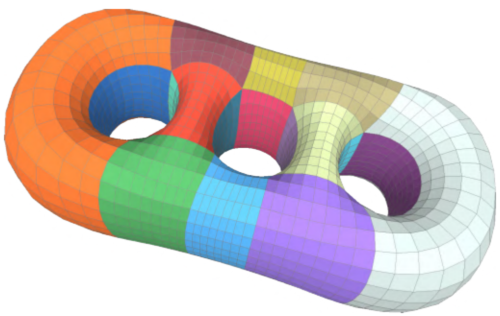
input:	636 dom
output:	197 dom
gain:	70%
time:	2.43 secs

- Comparable results
- Time: two orders of magnitude lower

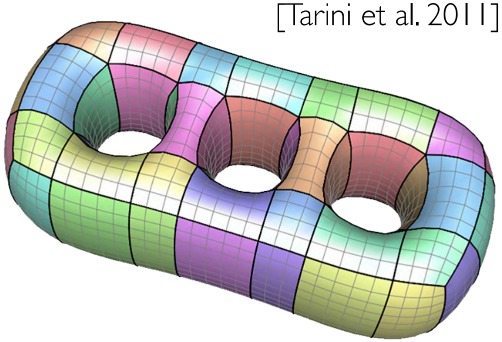
Limitations

- Corner pairing
- Domain number
- Mapping distortion

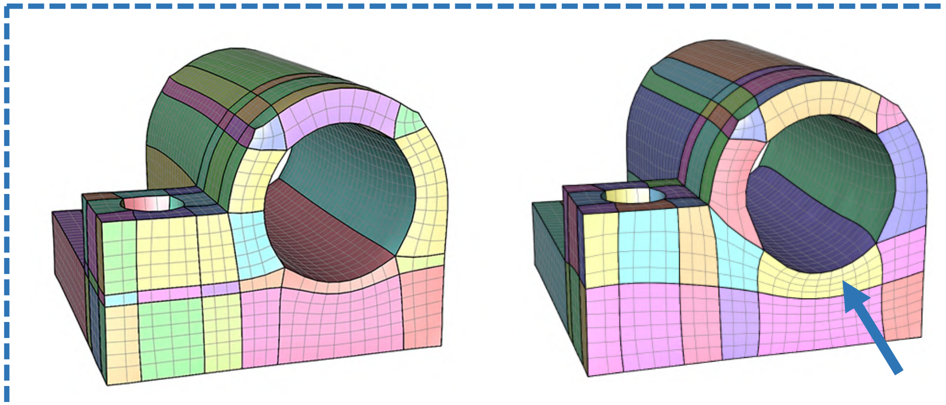




[Tarini et al. 2011]



[Ours]



About this work

DOI: 10.1111/cgf.12959
Eurographics Symposium on Geometry Processing 2016
Maks Ovsjanikov and Daniele Panozzo
(Guest Editors)

Volume 35 (2016), Number 5

Polycube Simplification for Coarse Layouts of Surfaces and Volumes

Gianmarco Cherchi
Università degli Studi di Cagliari, Italy

Marco Livesu
CNR IMATI, Genoa, Italy

Riccardo Scateni
Università degli Studi di Cagliari, Italy

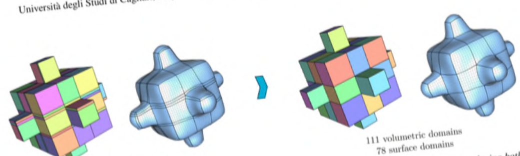


Figure 1: *State-of-the-art methods for polycube-based meshing do not consider the corner alignment problem, thus producing both surface and volumetric meshes with a poor structure (left). We propose an additional step in the pipeline: given a polycube we optimize for the position of its corners, maximizing the singularity alignment and producing well structured meshes with remarkably less domains (right). In this example we show how, with a few tiny adjustments on the position of the polycube corners, our method has been able to reduce the number of volumetric domains from 276 to 111 and the number of surface domains from 111 to 78.*

Abstract
Representing digital objects with structured meshes that embed a coarse block decomposition is a relevant problem in applications like computer animation, physically-based simulation and Computer Aided Design (CAD). One of the key ingredients to produce coarse block structures is to achieve a good alignment between the mesh singularities (i.e., the corners of each block). In this paper we improve on the polycube-based meshing pipeline to produce both surface and volumetric coarse block-structured meshes of general shapes. To this aim we add a new step in the pipeline. Our goal is to optimize the positions of the polycube corners to produce as coarse as possible base complexes. We rely on re-mapping the positions of the corners on an integer grid and then using integer numerical programming to reach the optimal. To the best of our knowledge this is the first attempt to solve the singularity misalignment problem directly in polycube space. Previous methods for polycube generation did not specifically address this issue. Our corner optimization strategy is efficient and requires a negligible extra running time for the meshing pipeline. In the paper we show that our optimized polycubes produce coarser block structured surface and volumetric meshes if compared with previous approaches. They also induce higher quality hexahedral meshes and are better suited for spline fitting because they reduce the number of splines necessary to cover the domain, thus improving both the efficiency and the overall level of smoothness throughout the volume.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Introduction

A cuboid can be trivially turned into a structured mesh by gridding it. This simple statement has triggered the development of techniques that aim to transform a shape into a conformal collection of connected cuboids, also known as *polycubes* [THCM04].

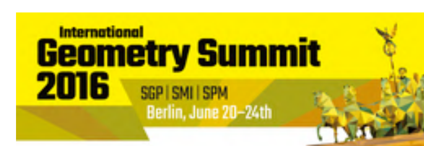
The complex problem of generating a structured mesh out of a general shape becomes straighforward with a polycube at hand. The common pipeline starts with a morph that transforms the input shape into a polycube, generating a volumetric map; the cuboids of the polycube are then subdivided so as to achieve the desired scale and, as a last step, the inverse map is used to transfer the

Polycube Simplification for Coarse Layouts of Surfaces and Volumes

Computer Graphics Forum
Wiley 2016

G. Cherchi, R. Scateni
University of Cagliari (IT)

M. Livesu
CNR-IMATI, Genoa (IT)



Work presented at the IGS 2016
SGP Chapter – June 2016
Berlin (GE)

models available on
hexalab.net

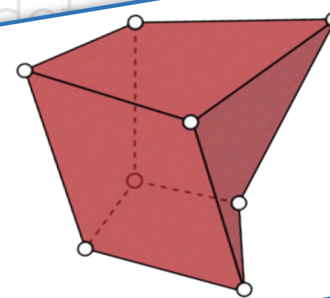
Contribution #2:
Selective Padding for Polycube-based Hex-meshes

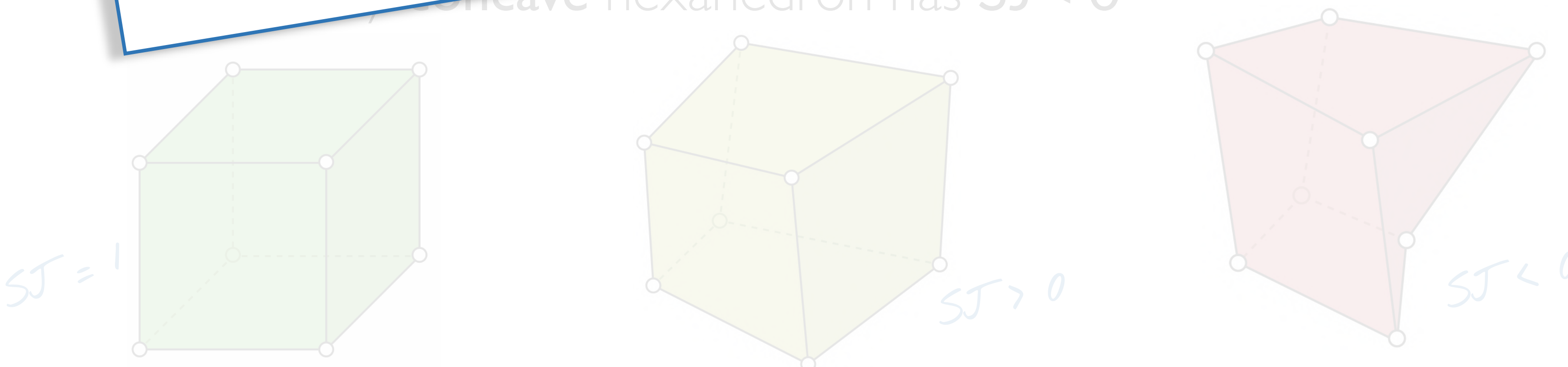
Scaled Jacobian (SJ)

- It is the most popular **quality metric** for hex-meshes
- It measures the quality of a hexahedron def… within the range [-1,1] :

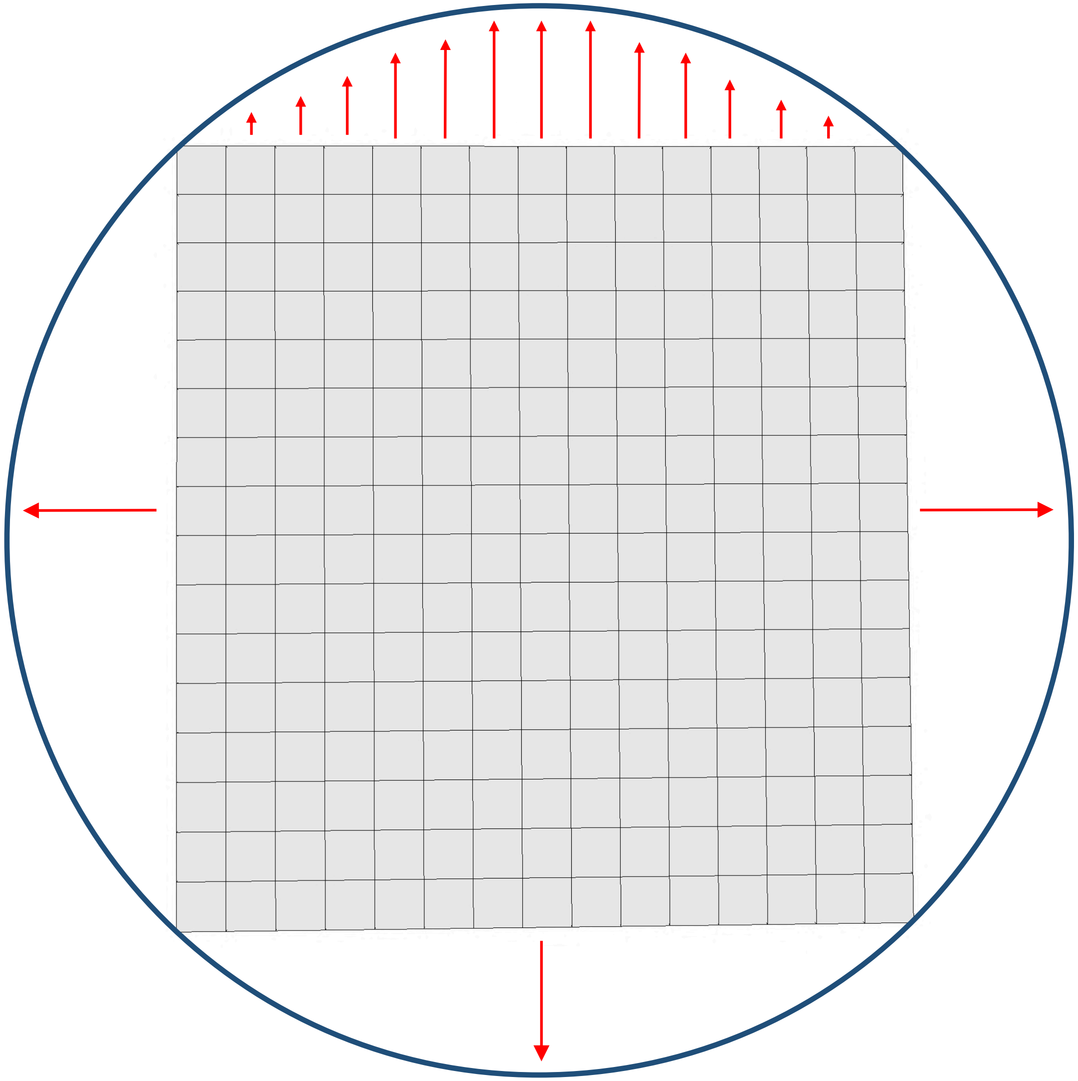
…concave hexahedron has SJ < 0

even ONE non-convex (inverted) element makes meshes unusable!

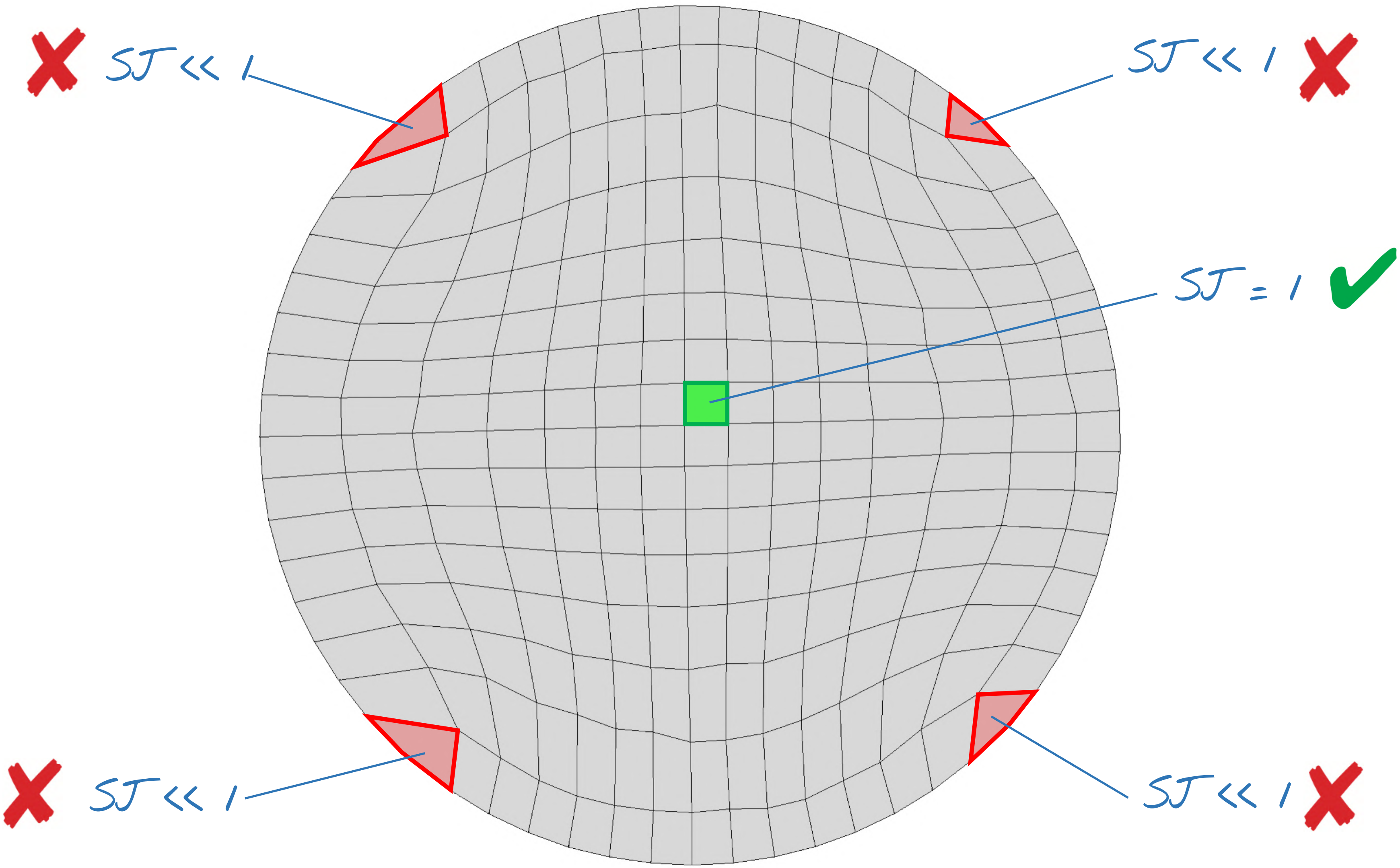




Polycube-based hex-meshes

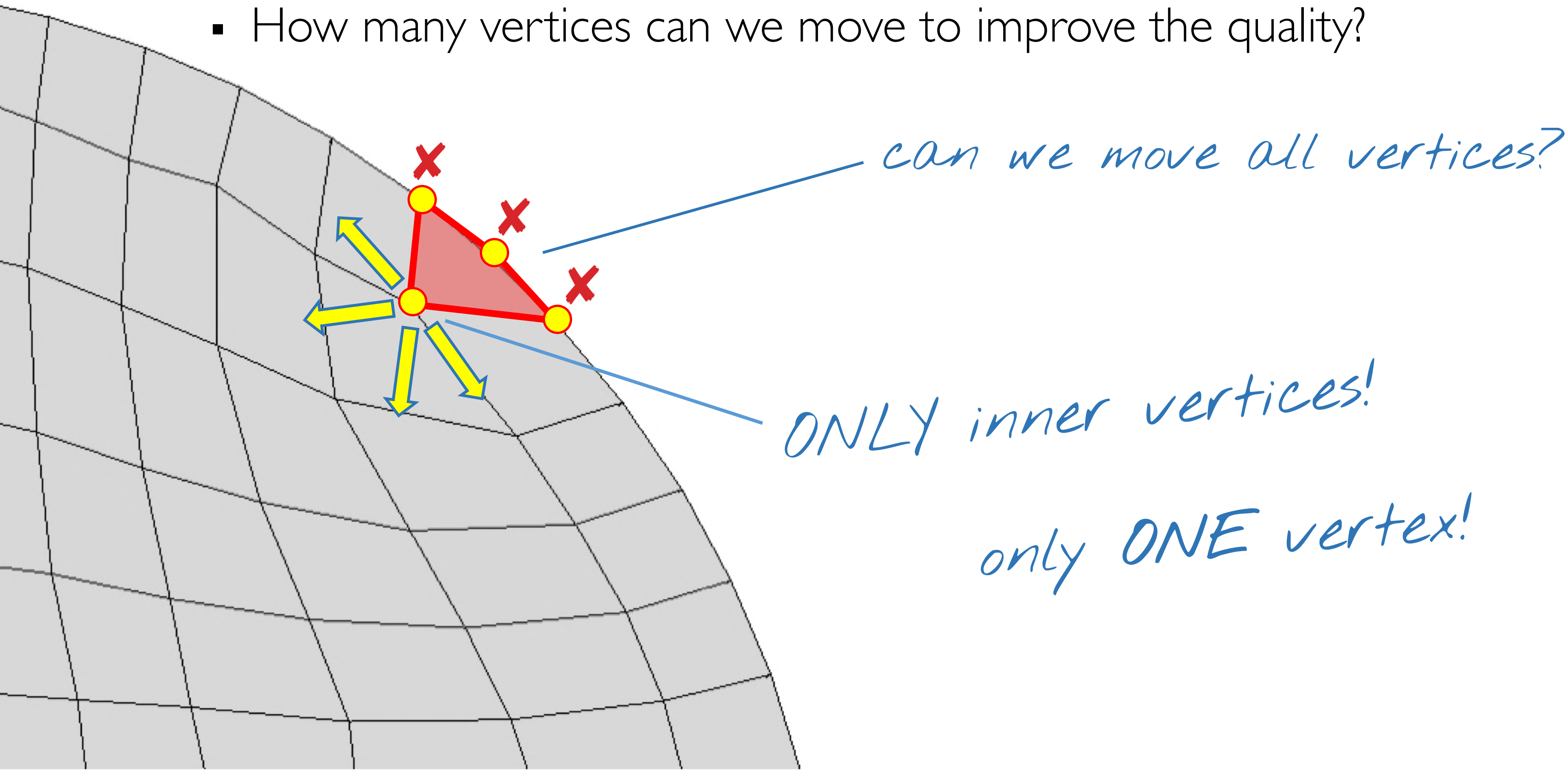


Quality analysis



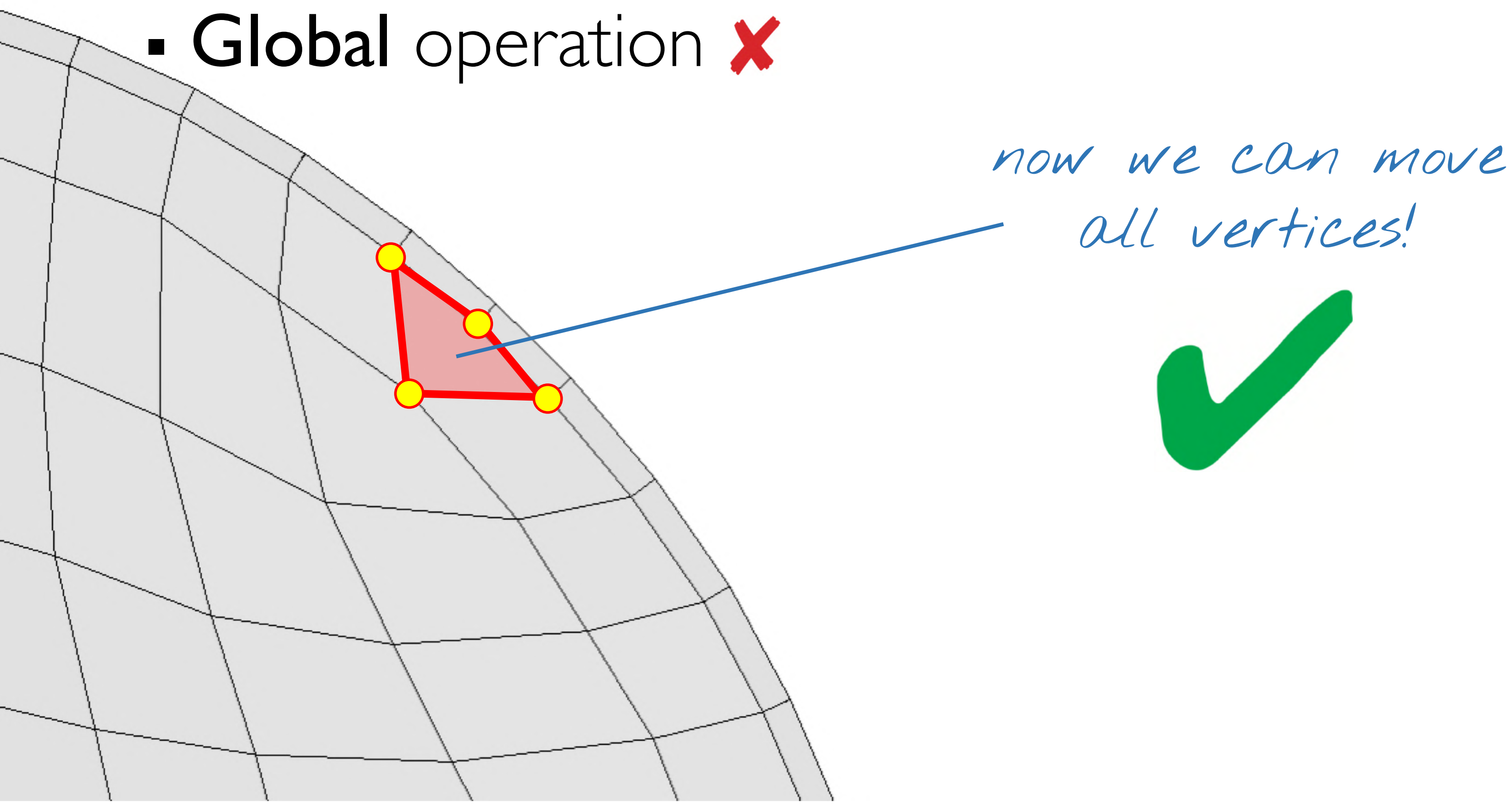
Quality improvement

- We have a problem with the degrees of freedom
  - How many vertices can we move to improve the quality?



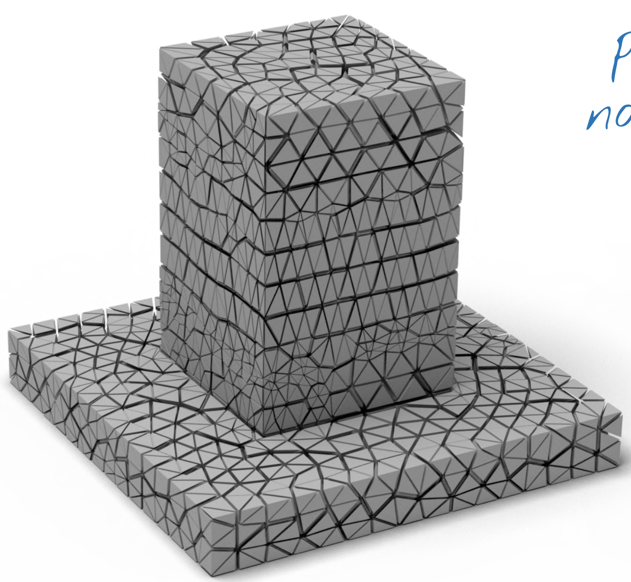
The solution is "padding" the mesh

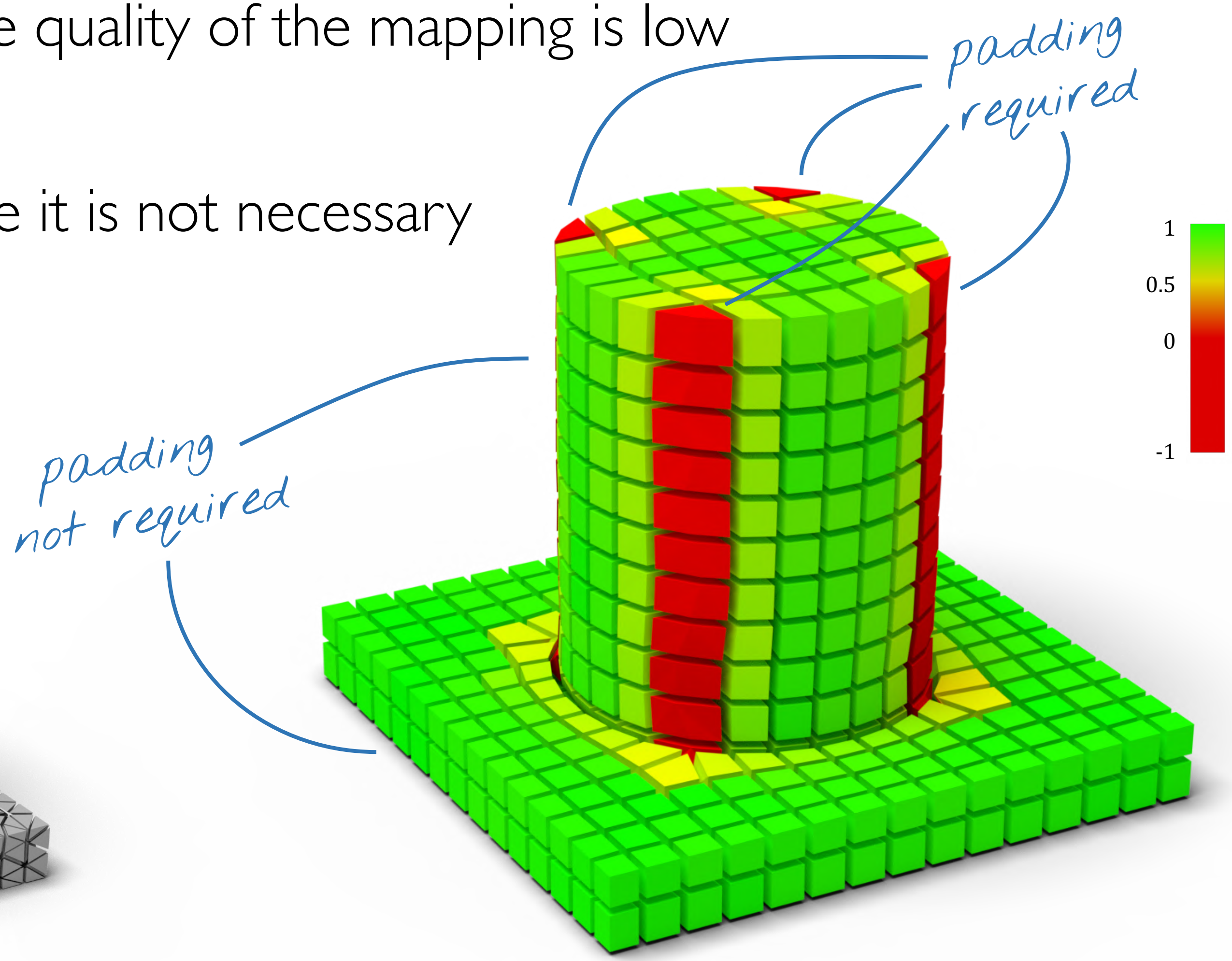
- We add a layer of new hexahedra in **all the surface** to add degrees of freedom
- **Global** operation ✘



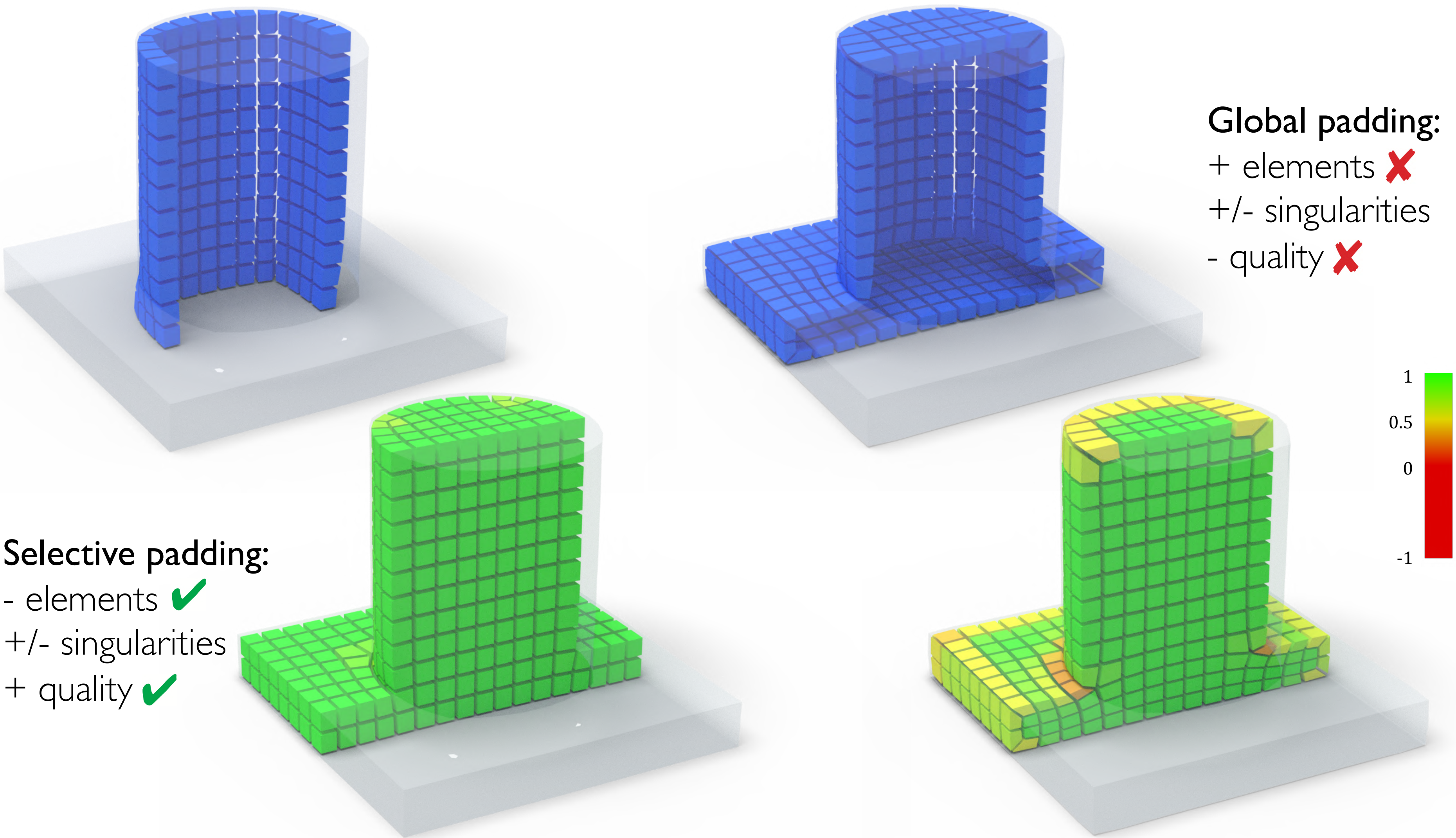
Idea: Selective padding

- Padding where the quality of the mapping is low
- No padding where it is not necessary





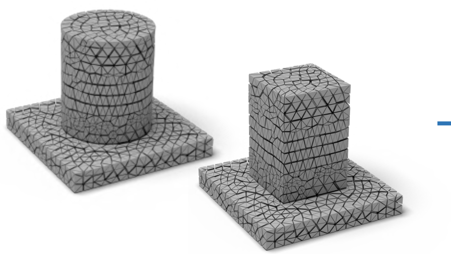
Selective padding vs global padding



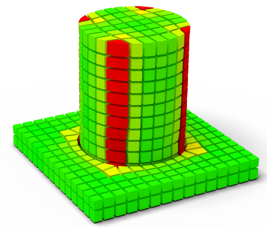
Global padding:
+ elements ✘
+/- singularities
- quality ✘

Selective padding:
- elements ✔
+/- singularities
+ quality ✔

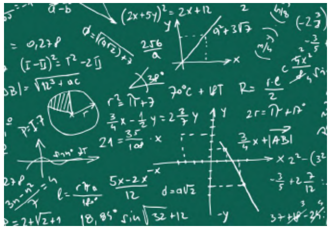
Selective padding pipeline



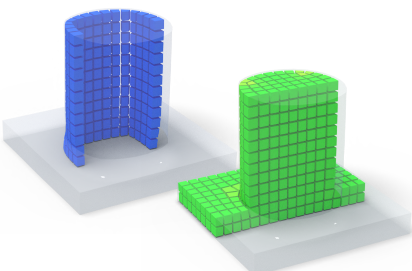
We start from the **model**
and its **polycube**



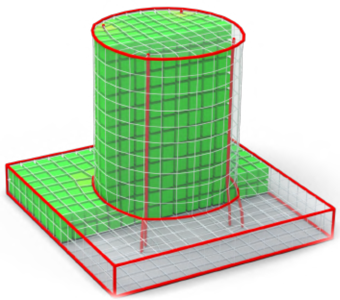
We compute the **hex-mesh** and
we **analyse the quality** of the mapping
to decide where the padding is required



We set and solve a **mathematical model**
that compute the position
of the padding layer



We perform the padding as a
sheet insertion in the polycube
and compute the **new hex-mesh**



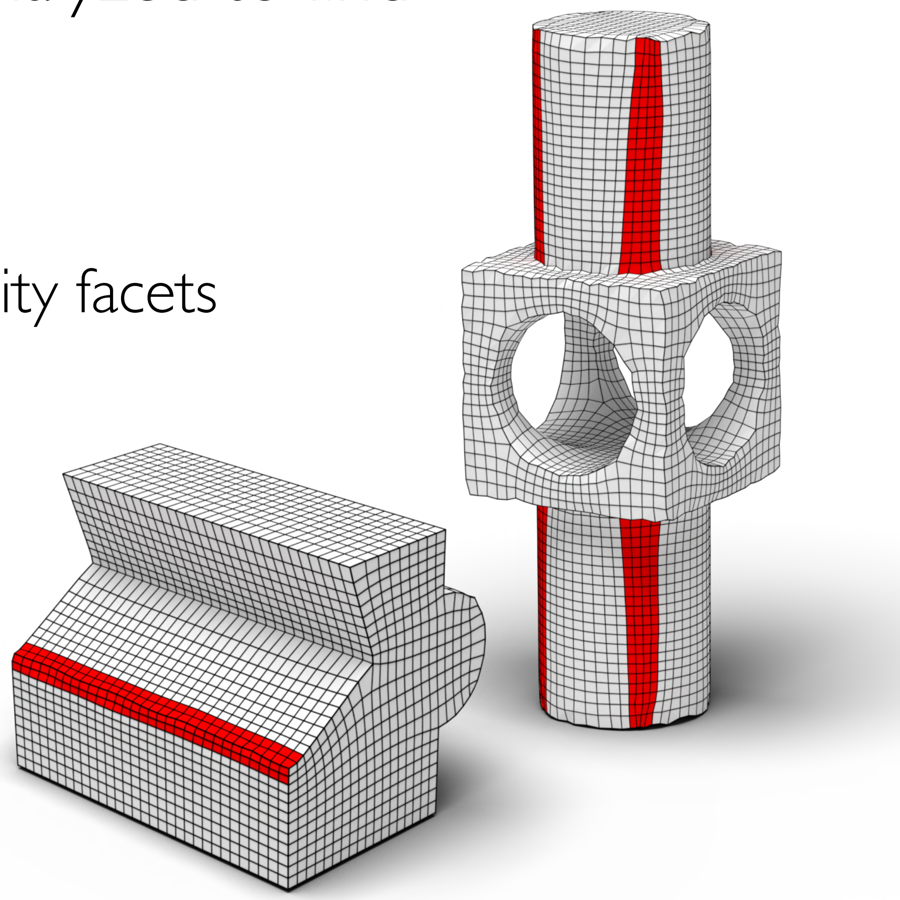
Now we can **optimize** the hex-mesh
with the new degrees of freedom
(better quality)

Distortion analysis

The surface of the model is analyzed to find the elements with low quality

We obtain the final set of low quality facets (HF set)

HF is the set of hard constraints of the mathematical model



Mathematical model

We want to set a mathematical model as follow:

- **INPUT**: *set of faces that require a padding (HF set)*

- **OUTPUT**: *set of faces to pad*
  - Taking care about the number of new hexahedra
  - Taking care about the number of new singularities
  - Taking care about topological consistency

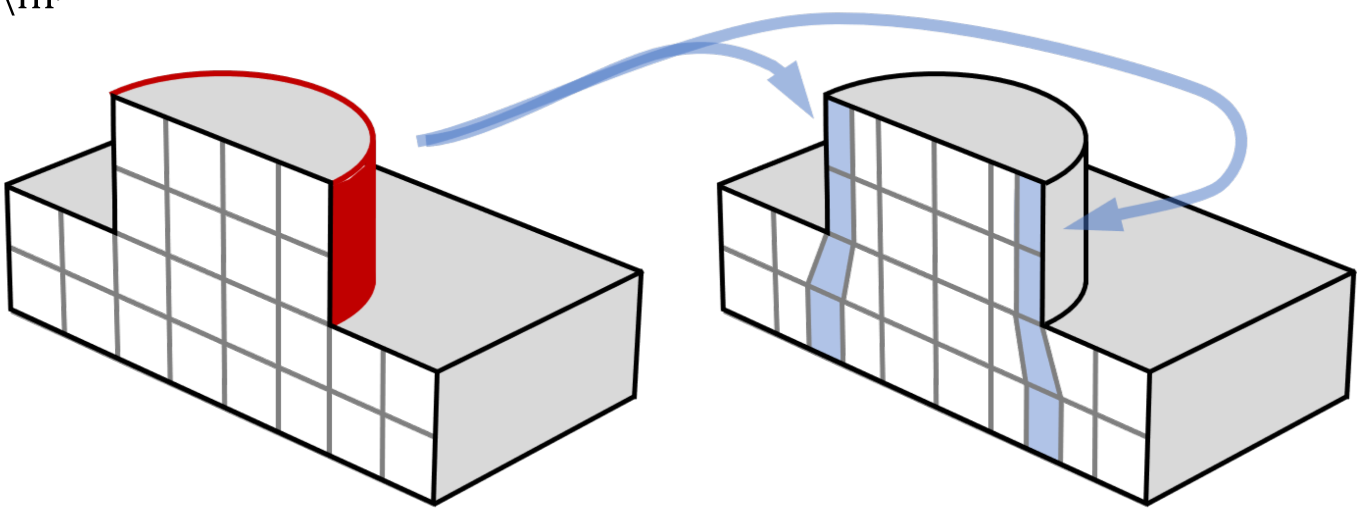
Find padding facets

A mathematical model with binary and integer variables

$$\underset{s.t.}{\min E} = \boxed{E_{padding}} + \lambda \cdot E_{complexity}$$

$$structural\ constraints$$

$$E_{padding} = \sum_{f_i \in F \setminus \mathrm{HF}} x f_i$$



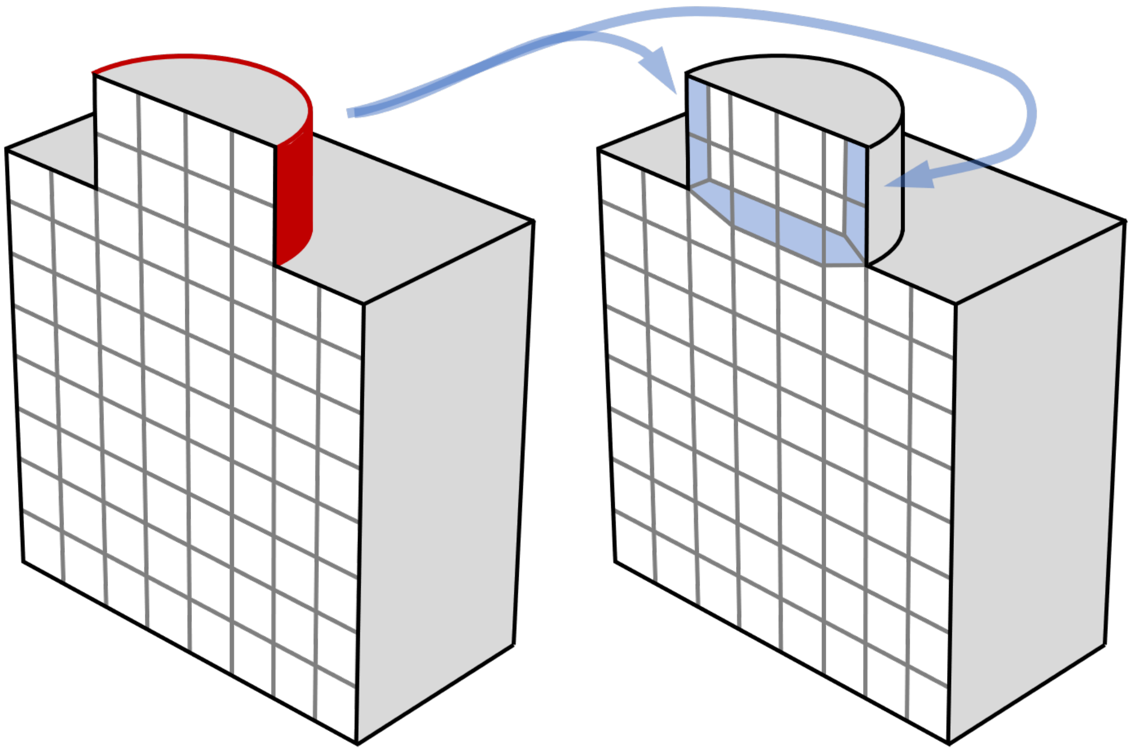
NB: simplified formulas. Extended version of formulas in the thesis.

Find padding facets

A mathematical model with binary and integer variables

$$\min E = E_{padding} + \lambda \cdot \boxed{E_{complexity}}$$
$$s.t.$$
$$structural\ constraints$$

$$E_{complexity} = \sum_{e_j \in E^*} te_j + \sum_{v_l \in V^*} tv_l$$



NB: simplified formulas. Extended version of formulas in the thesis.

Find padding facets

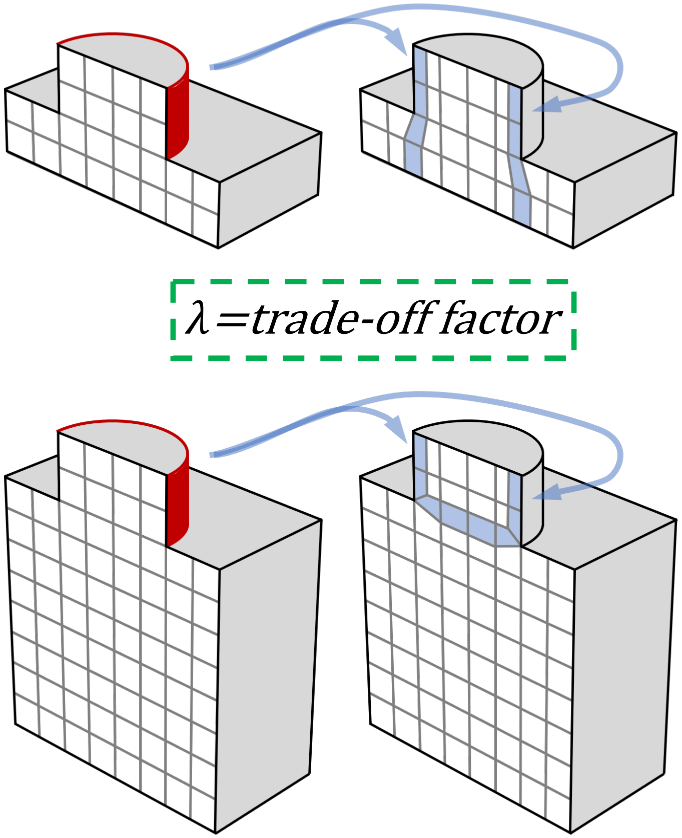
A mathematical model with binary and integer variables

$$\min E = E_{padding} + \boxed{\lambda} \cdot E_{complexity}$$

$$\text{s.t.}$$
$$\text{structural constraints}$$

$$E_{padding} = \sum_{f_i \in F \setminus \mathrm{HF}} xf_i$$

$$E_{complexity} = \sum_{e_j \in E^*} te_j + \sum_{v_l \in V^*} tv_l$$



NB: simplified formulas. Extended version of formulas in the thesis.

Find padding facets

A mathematical model with binary and integer variables

$$min\ E = E_{padding} + \lambda \cdot E_{complexity}$$

$$s.t.$$

$$\boxed{structural\ constraints}$$

$$E_{padding} = \sum_{f_i \in F \setminus \mathrm{HF}} xf_i$$

$$E_{complexity} = \sum_{e_j \in E^*} te_j + \sum_{v_l \in V^*} tv_l$$

λ=*trade-off factor*

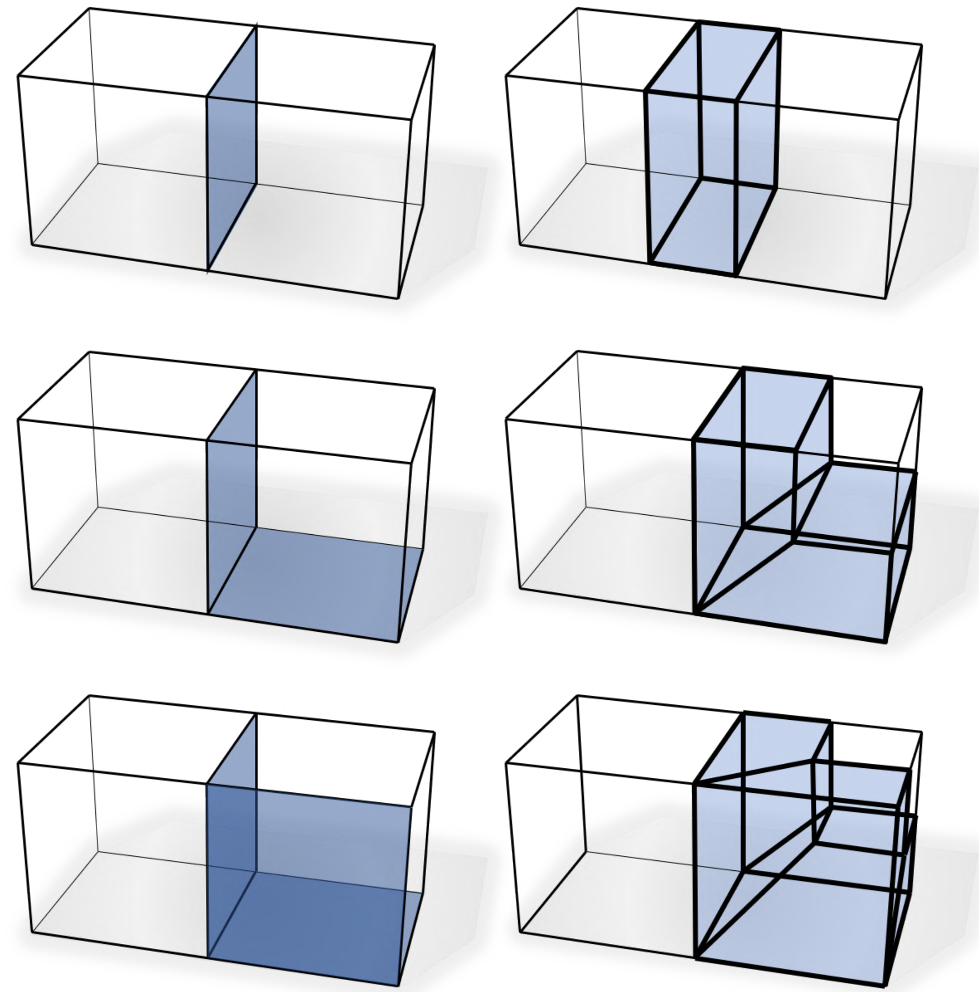
- Correct propagation of the new layer
- Counting vertex turns
- Counting edge turns

NB: simplified formulas. Extended version of formulas in the thesis.

The sheet insertion

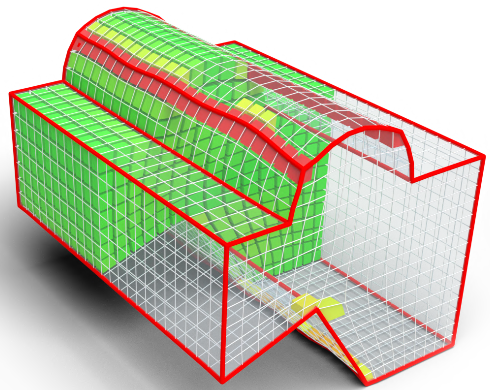
Now we know the set of facets to “pad” to create the padding layer

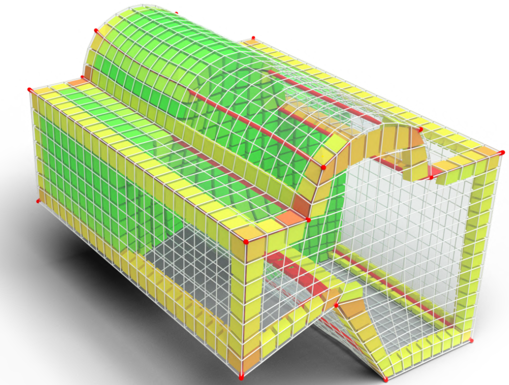
Padding == facet extrusion



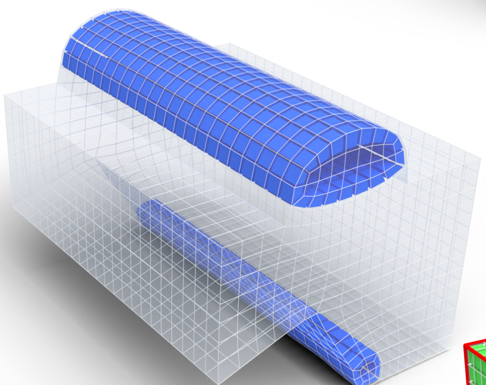
Results

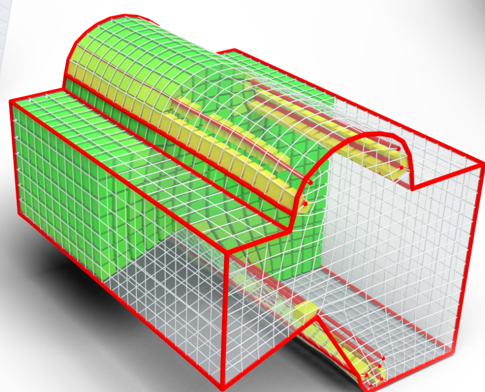
Original model
min SJ: 0.10
avg SJ: 0.96
#Hex: 4347



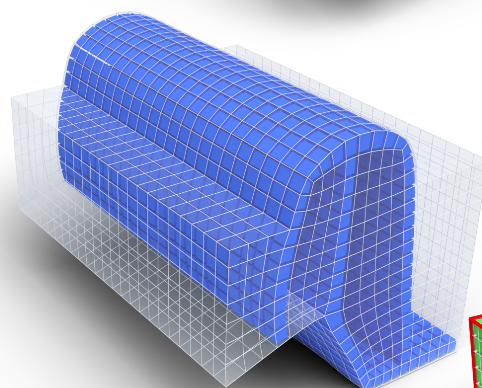


Global padding
min SJ: 0.10
avg SJ: 0.94
#Hex: 6197

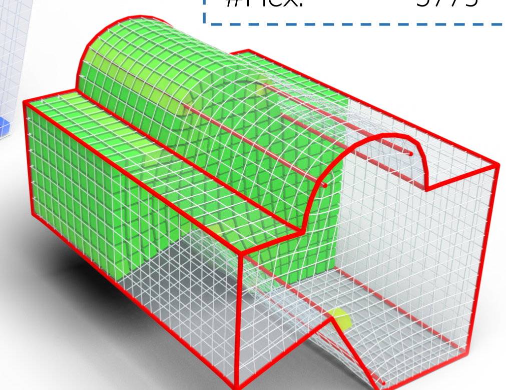




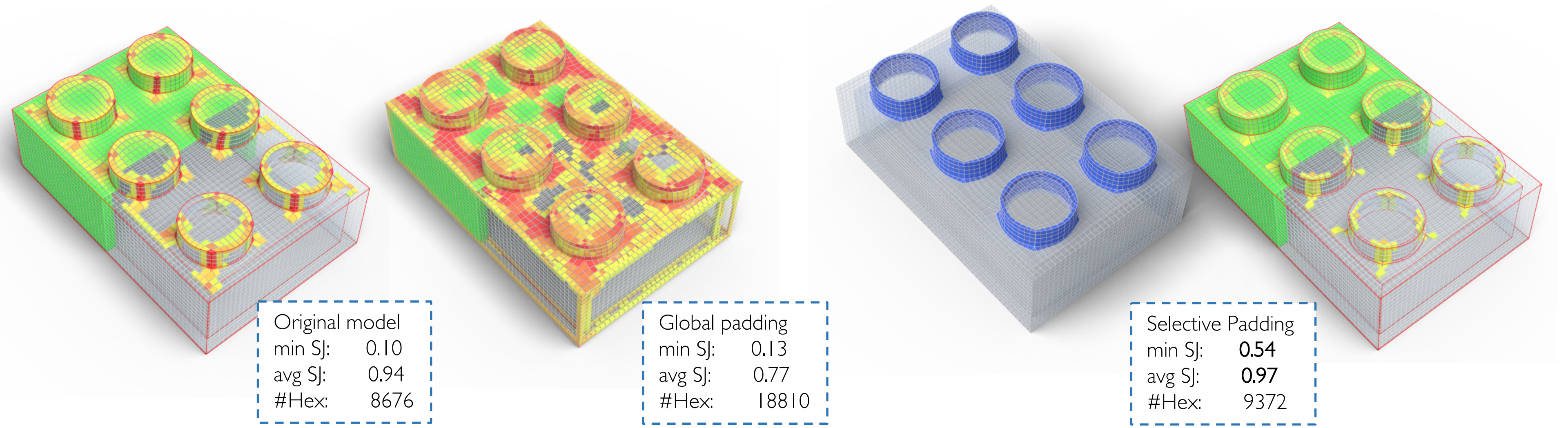
Select. padding (**λ=0**)
min SJ: **0.60**
avg SJ: **0.95**
#Hex: 4945



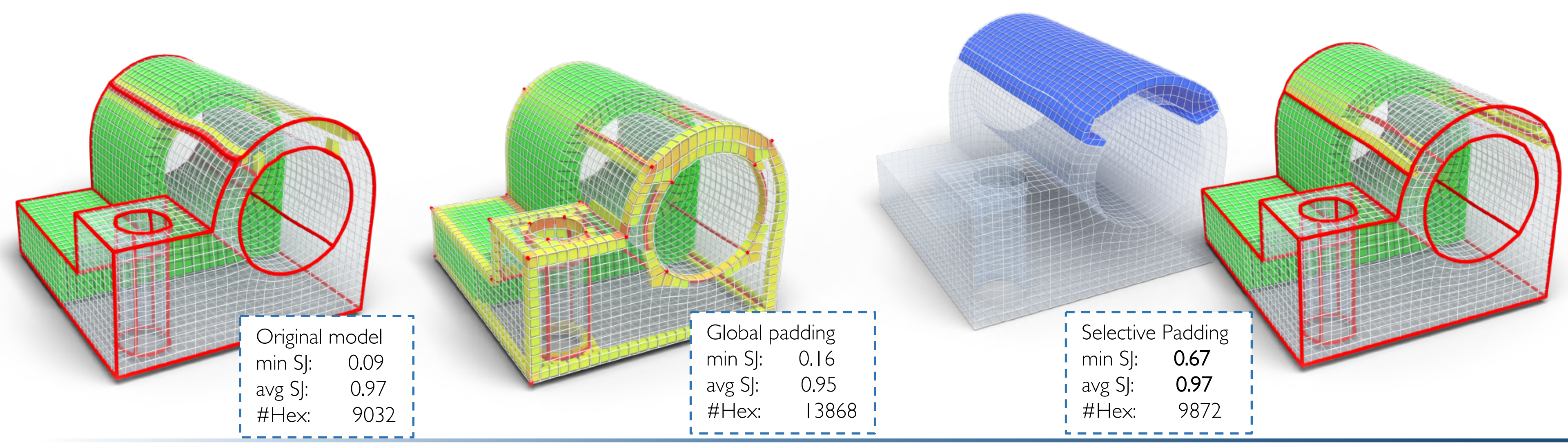
Select. padding (**λ=4**)
min SJ: **0.74**
avg SJ: **0.98**
#Hex: 5773



Results

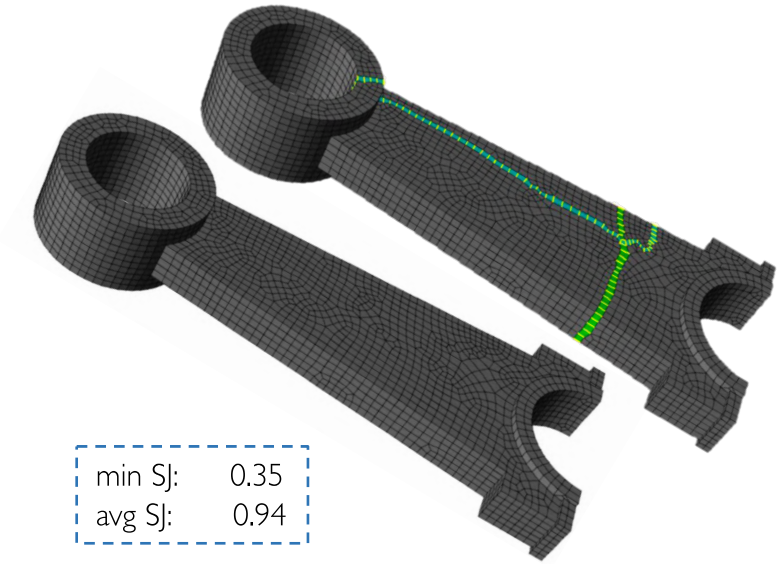


Original model
min SJ: 0.10
avg SJ: 0.94
#Hex: 8676
Global padding
min SJ: 0.13
avg SJ: 0.77
#Hex: 18810
Selective Padding
min SJ: **0.54**
avg SJ: **0.97**
#Hex: 9372

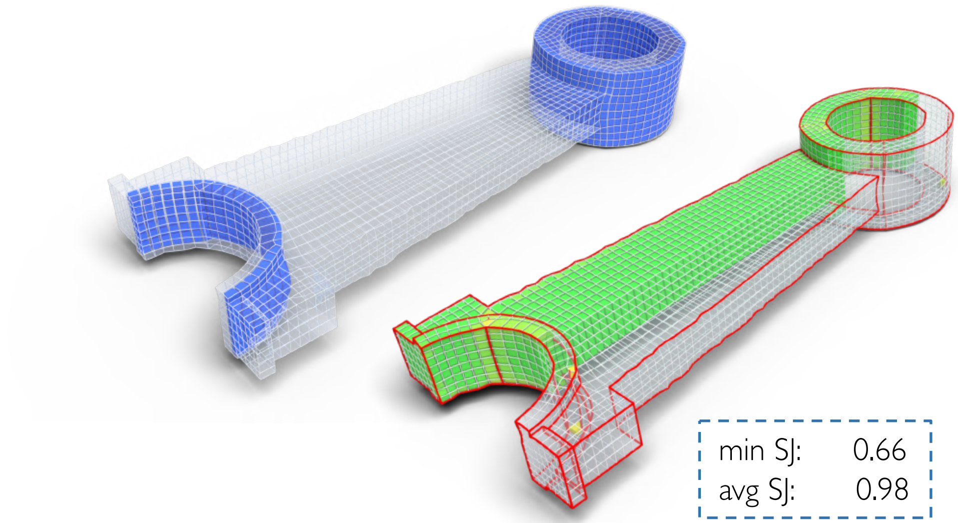


Original model
min SJ: 0.09
avg SJ: 0.97
#Hex: 9032
Global padding
min SJ: 0.16
avg SJ: 0.95
#Hex: 13868
Selective Padding
min SJ: **0.67**
avg SJ: **0.97**
#Hex: 9872

Comparison



[Wang et al. 2018]



[Ours]

- Comparable results (usually better)
- More regular inner structure (and less singularities)
- We can focus the analysis only on the model surface

Limitations

- How to choose the **λ** parameter?

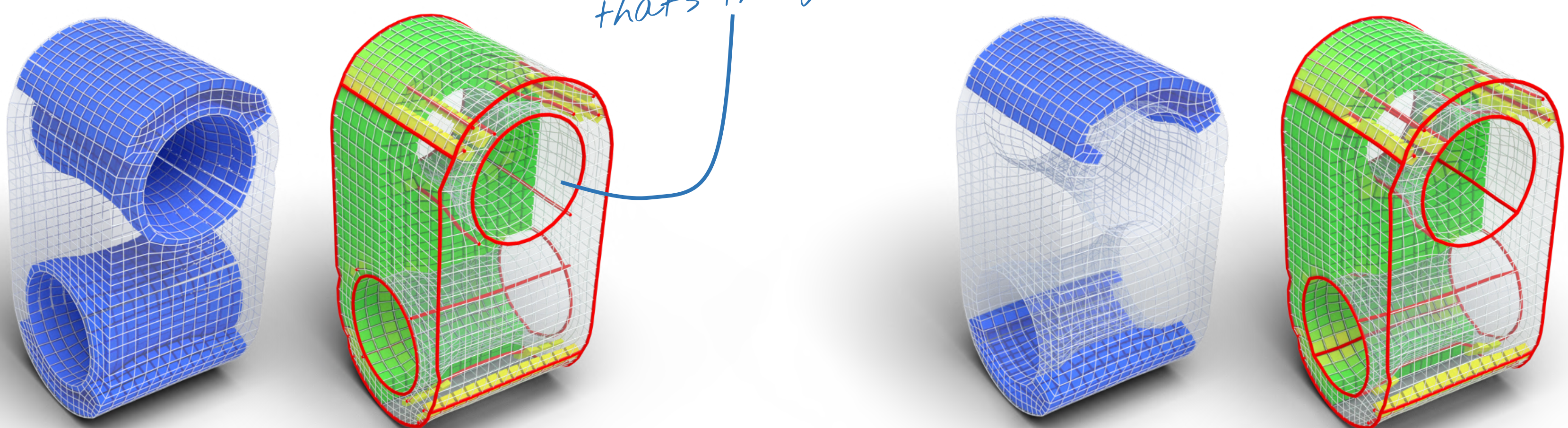
object shape
object final use
object resolution

- Padding "holes"

Selec. Padding **(holes)**
min SJ: 0.63
avg SJ: 0.95
#Hex: 4550

to pad, or not to pad?
that's the question

Selec. Padding **(NO holes)**
min SJ: **0.65**
avg SJ: 0.95
#Hex: **3770**



About this work

COMPUTER GRAPHICS *forum*

Volume 0 (1981), Number 0 pp. 1–12

Selective Padding for Polycube-based Hexahedral Meshing

G. Cherchi[1], P. Alliez[2], R. Scateni[1], M. Lyon[3] and D. Bommes[4]

[1]University of Cagliari, Italy.
[2]Inria Sophia Antipolis - Mediterranée and Université Côte d'Azur, France.
[3]RWTH Aachen University, Germany.
[4]University of Bern, Switzerland.

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

Figure 1: *Overview. Our pipeline takes as input a model and its polycube mapping (a); we compute the relative hex-mesh and locate the surface areas in need of padding analyzing the mapping quality (b); we set and solve a binary problem to find a set of facets to extrude in order to create a selective padding layer (c); we compute and analyze the mapping with the new hex-mesh structure (d).*

Abstract
Hexahedral meshes generated from polycube mapping often exhibit a low number of singularities but also poor quality elements located near the surface. It is thus necessary to improve the overall mesh quality, in terms of the minimum Scaled Jacobian (MSJ) or average Scaled Jacobian (ASJ). Improving the quality may be obtained via global padding (or pillowing), which pushes the singularities inside by adding an extra layer of hexahedra on the entire domain boundary. Such a global padding operation suffers from a large increase of complexity, with unnecessary hexahedra added. In addition, the quality of elements near the boundary may decrease. We propose a novel optimization method which inserts sheets of hexahedra so as to perform selective padding, where it is most needed for improving the mesh quality. A sheet can pad part of the domain boundary, traverse the domain and form singularities. Our global formulation, based on solving a binary problem, enables us to control the balance between quality improvement, increase of complexity and number of singularities. We show in a series of experiments that our approach increases the MSJ value and preserves (or even improves) the ASJ, while adding fewer hexahedra than global padding.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

1. Introduction

Volumetric meshes are an ideal tool for physically-based simulations, especially in the fields of mechanical simulation and computational fluid dynamics. While tetrahedral meshes are very simple to obtain and flexible in adapting to the computational domain, hexahedral meshes have better numerical properties and lower complexity. One straightforward way to generate a hexahedral mesh on a given domain passes through polycube mapping.

Polycubes are three-dimensional domains formed by connecting axis-aligned cuboids face to face. The simplicity of their structure explains their popularity for several computer graphics problems such as quadrilateral and hexahedral meshing (e.g., [HXH10, GSZ11]), efficient texture mapping (e.g., [THCM04, CL*10]), morphing (e.g., [FJFS05]), spline fitting (e.g., [WHL*07]), etc.

When the volumetric domains to partition in hexahedra are regular enough (e.g., with limited range of levels of details like mechanical and CAD models), polycubes are an ideal tool for generating hexahedral meshes, by computing volumetric mappings with their corresponding domains. However, the quality of the hexahedra directly depends on the mapping distortion. While the distortion is

© 2018 The Author(s)
Computer Graphics Forum © 2018 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

Selective Padding for Polycube-based Hexahedral Meshing

Computer Graphics Forum
Wiley 2019

G. Cherchi, R. Scateni
University of Cagliari (IT)

P. Alliez
INRIA, Sophia Antipolis (FR)

M. Lyon, D. Bommes
Aachen University (GE)

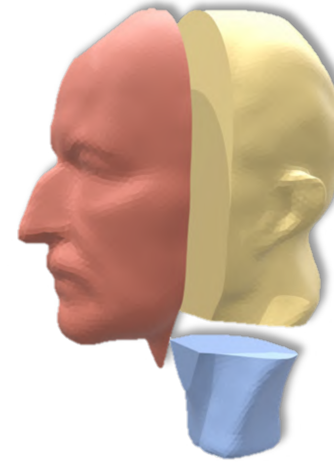
This work will be presented at
EUROGRAPHICS 2019
in May 2019, Genoa (IT)

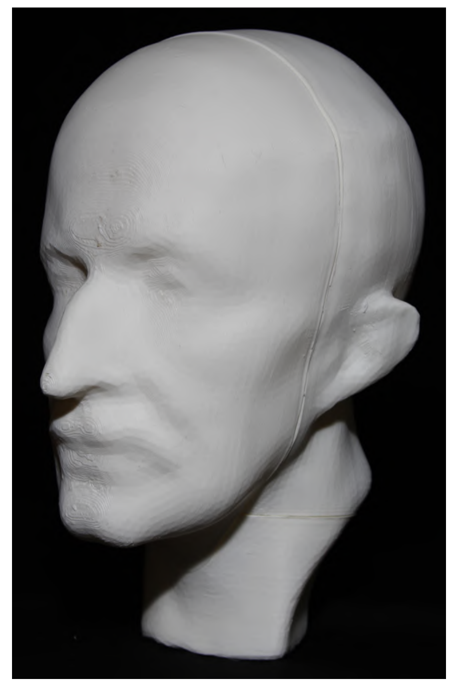
models available on
hexalab.net

From the digital world to manufacturing

Digital World

Fabrication



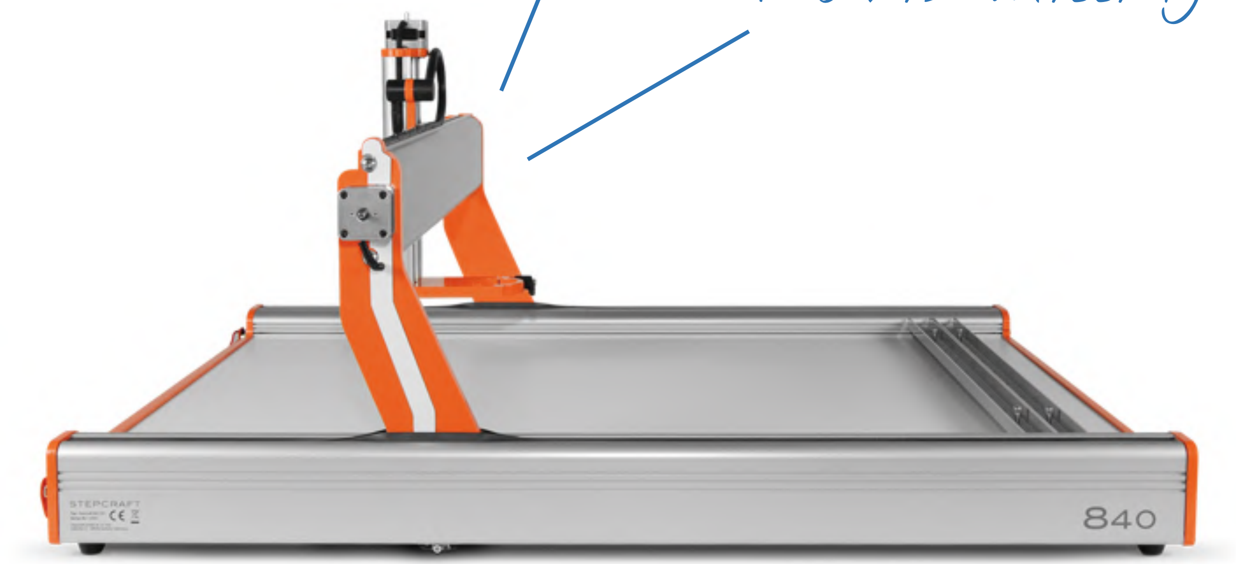


Contribution #3:
Polycube-based Decomposition for Fabrication

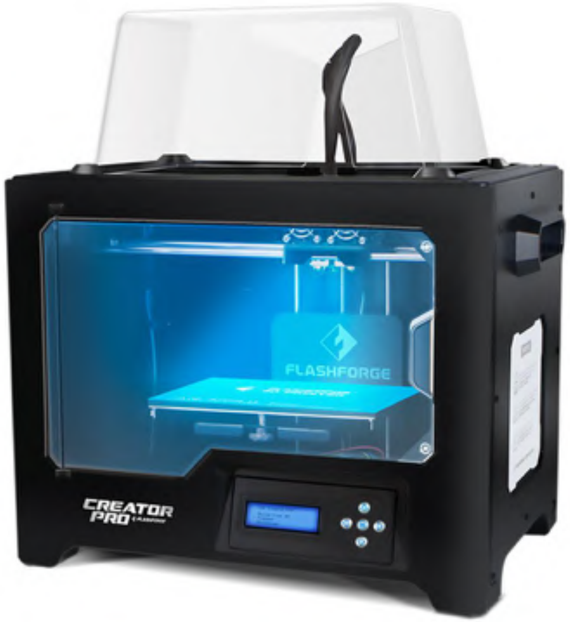
Digital fabrication

3-axis milling

4-axis milling



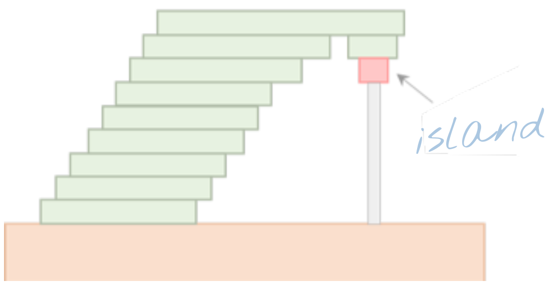
Subtractive manufacturing

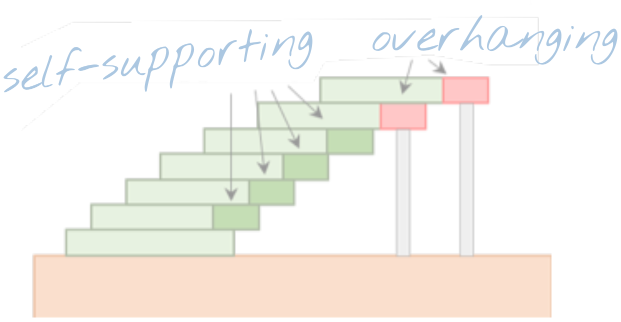


Additive manufacturing

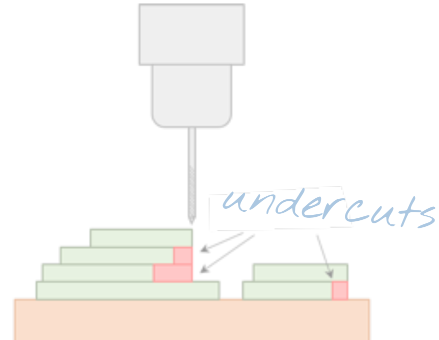
Digital fabrication: limitations

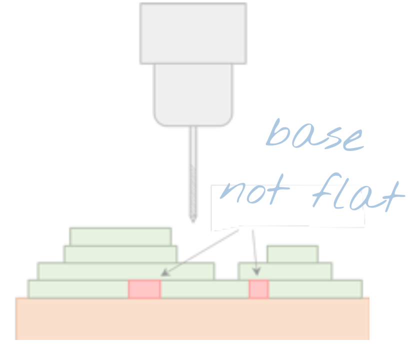
Additive manufacturing





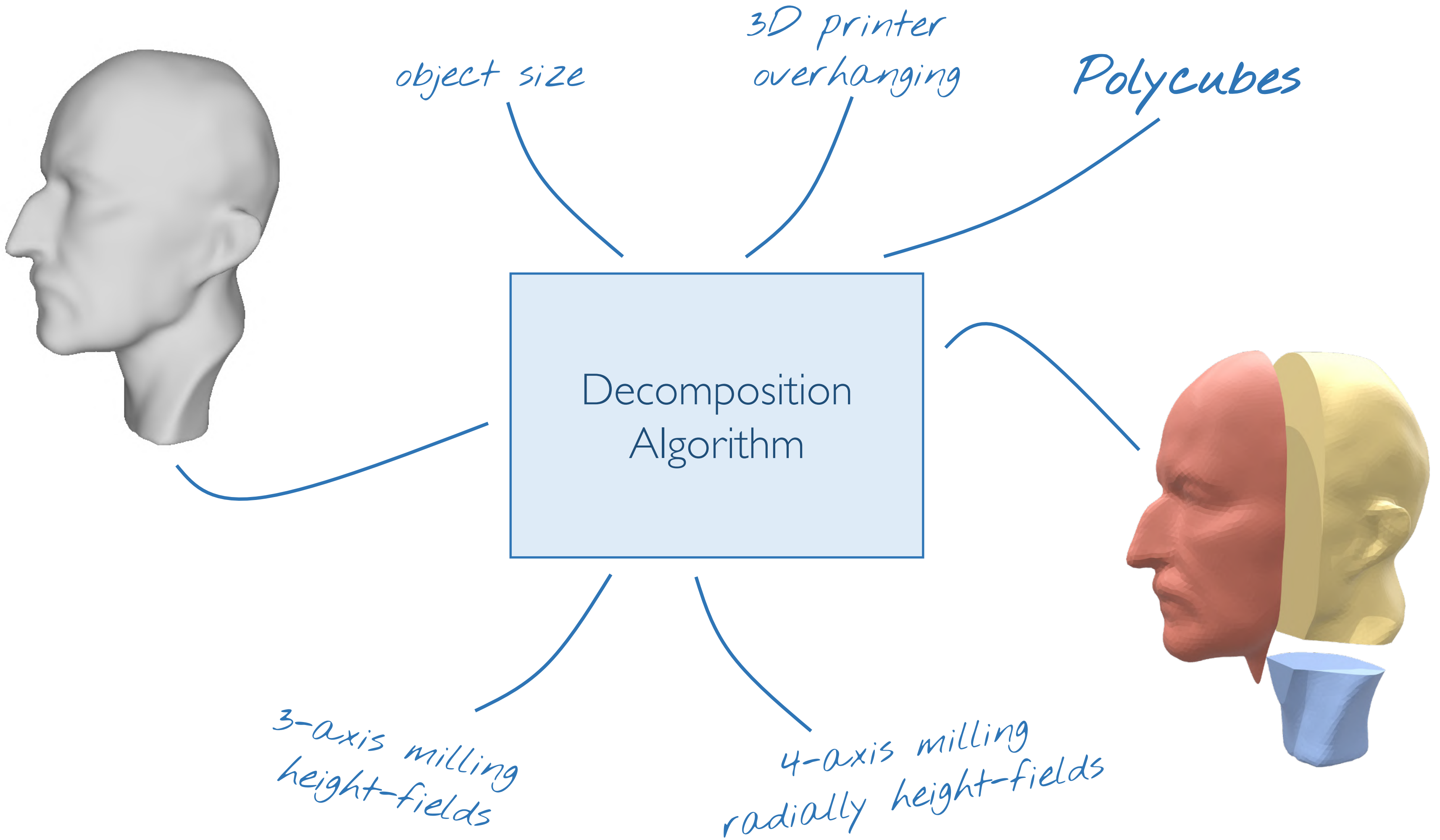
Subtracting manufacturing



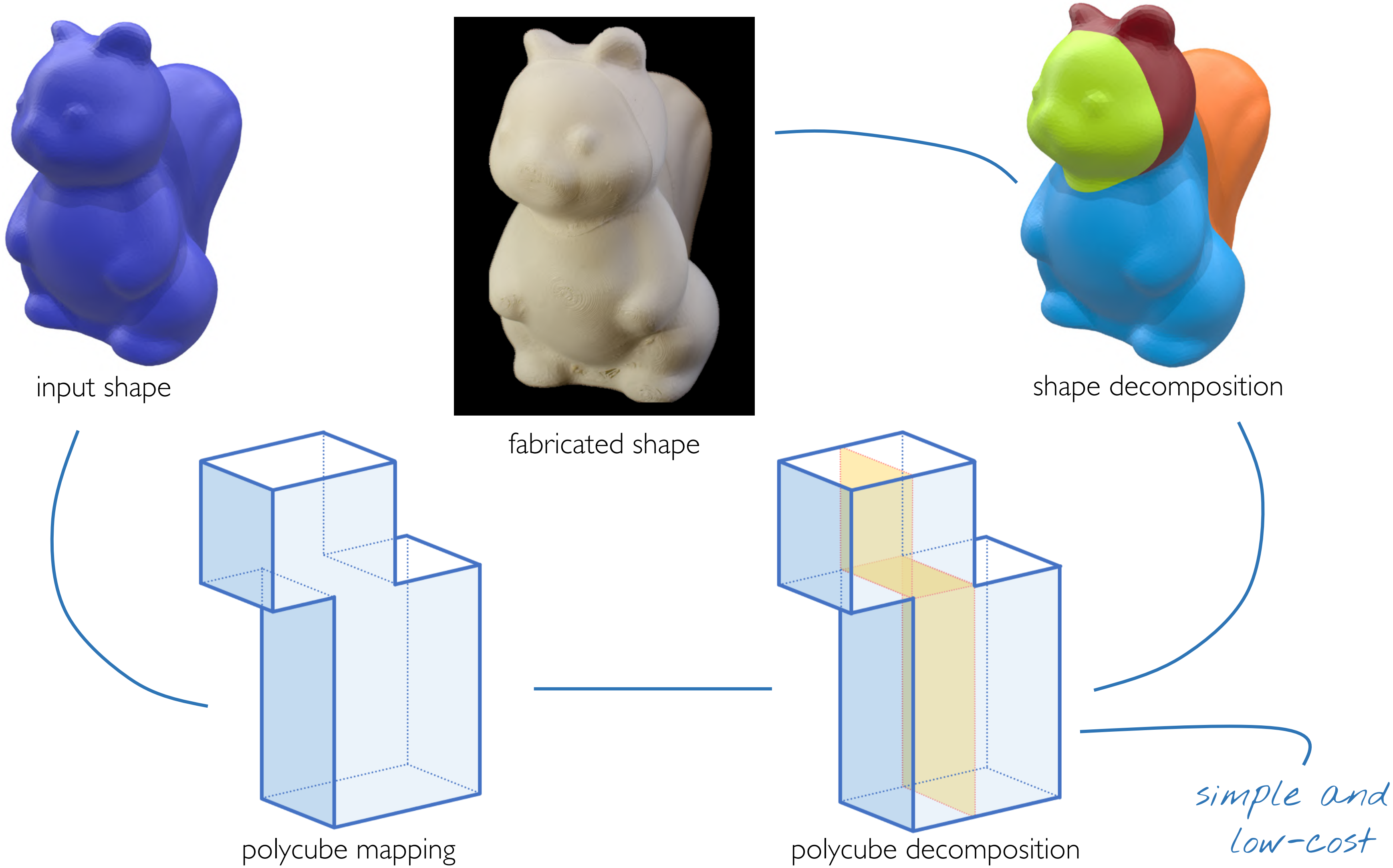


object size

Shape decomposition

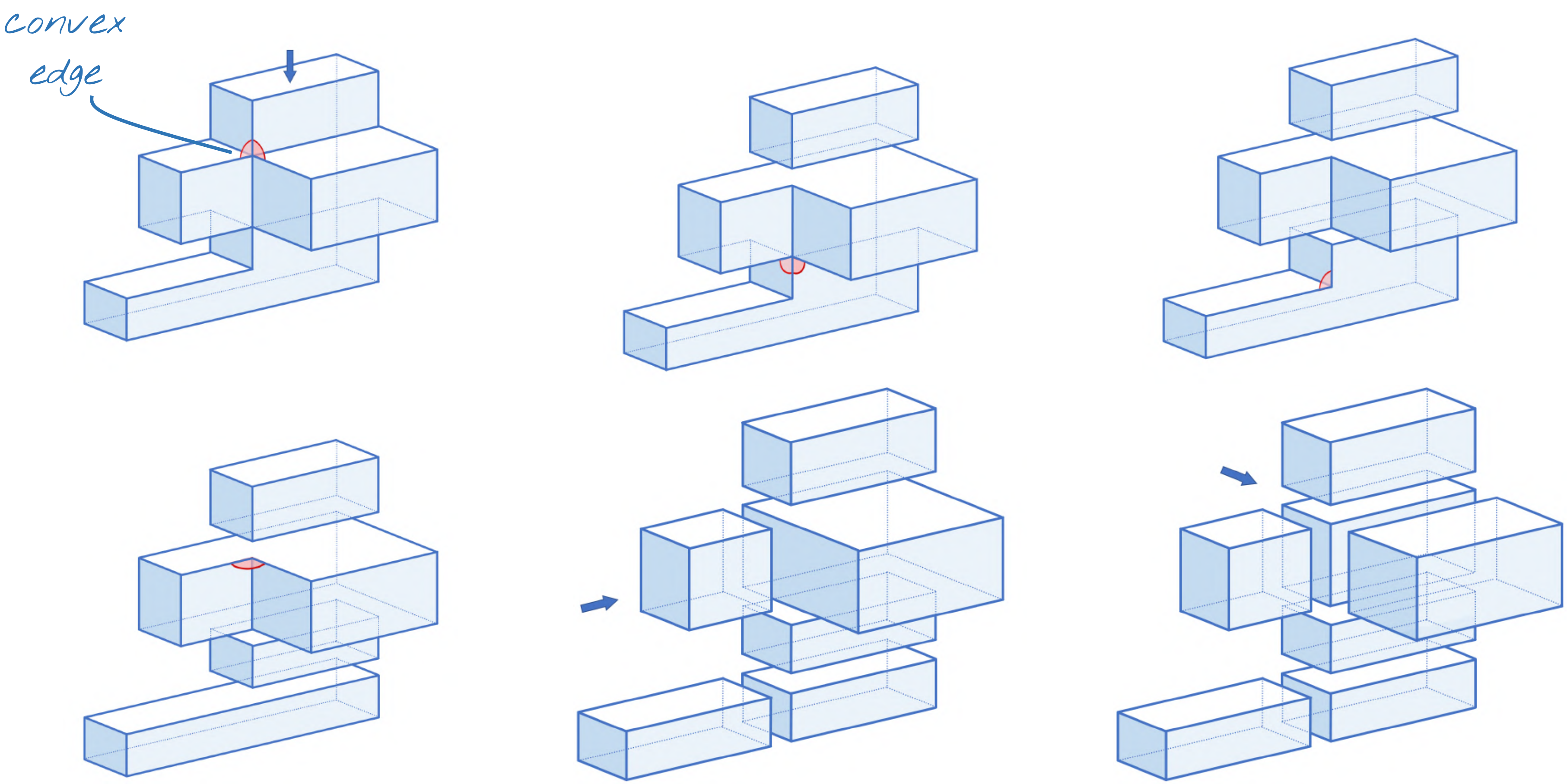


Idea: Polycube-based decomposition



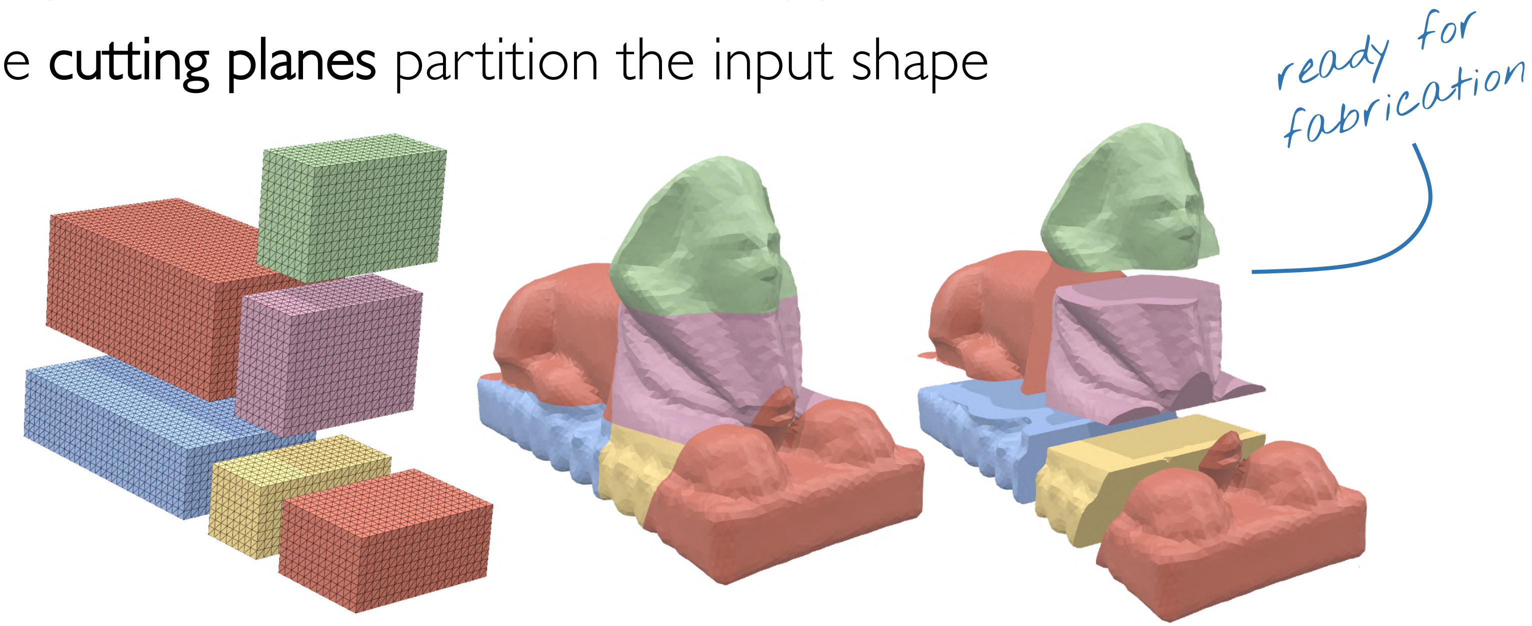
Polycube partition

A sweep line algorithm applied in each direction



Flattening and shape decomposition

- We use **barycentric coordinates** to map the corners of the found parallelepiped to their position in the input shape
- We use a **queue-based algorithm** to flatten touching facets between adjacent portions
- Every **flat facet** determines a cutting plane
- The **cutting planes** partition the input shape



NB: simplified steps. Extended version of steps (with formulas) in the thesis.

Fabricability check and analysis

3D printing:

Every piece is fabricable with a 3D printer, using one of the flat sides as the base

We chose the base that generates the lowest number of supports

3-axis milling:

A piece is fabricable only if it is a height-field for one of its flat sides

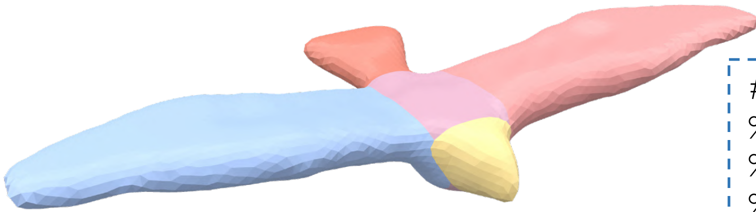
Empirically, we can obtain a height-field piece by splitting a no height-field one with an appropriate cutting plane

4-axis milling:

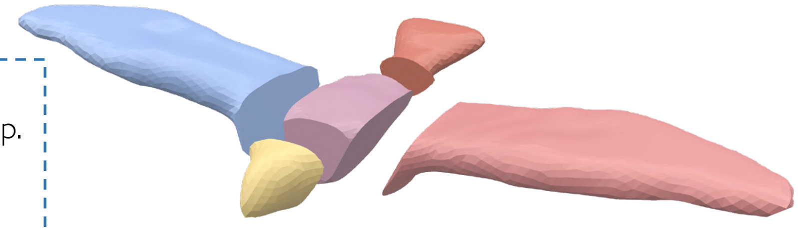
A piece is fabricable only if it is a height-field for a selected rotation axis

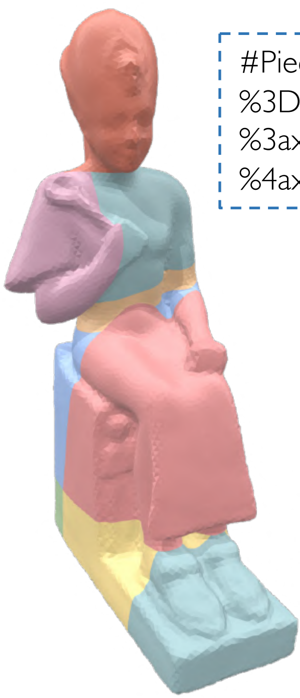
We give the user the possibility to chose the best axis for each piece and check if it is fabricable

Results

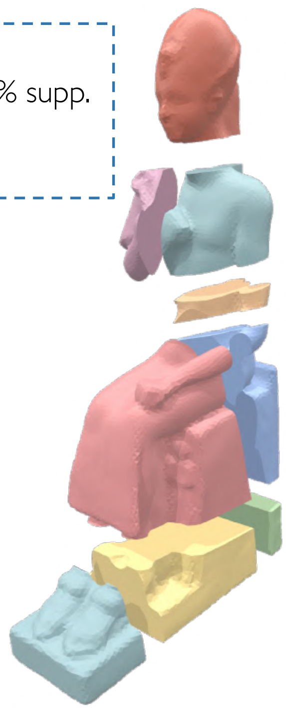


#Pieces: 5
%3DP: 32% → 0% supp.
%3axis: no
%4axis; 100%

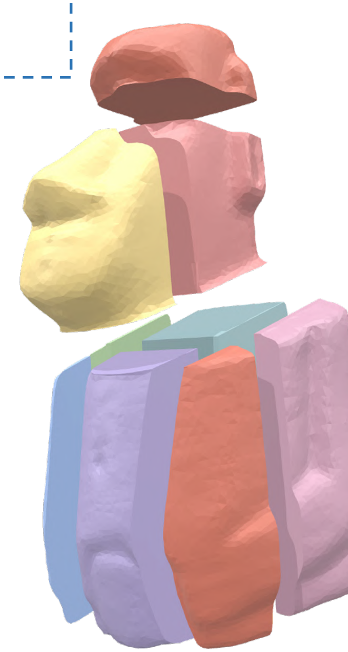




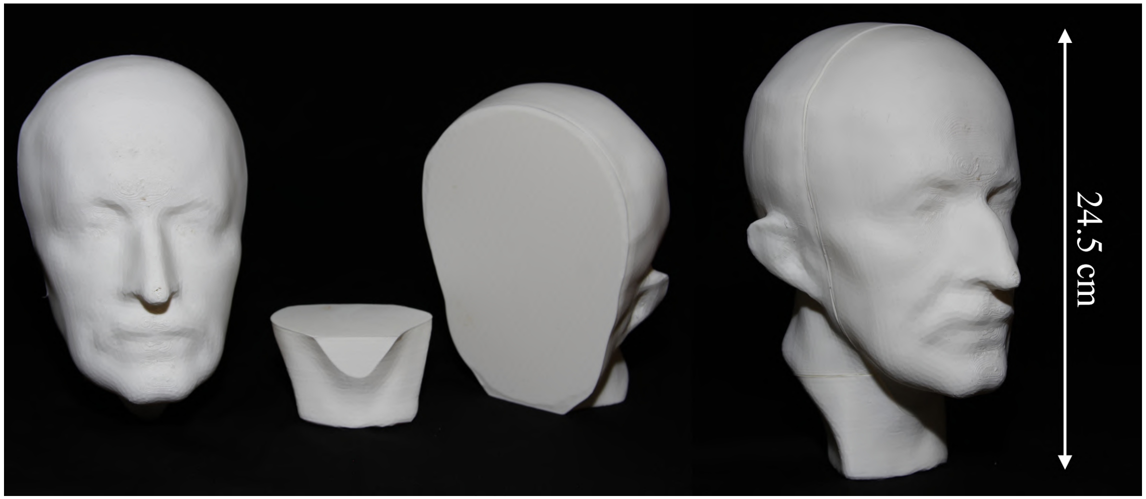
#Pieces: 9
%3DP: 3% → 1.7% supp.
%3axis: no
%4axis; 100%



#Pieces: 9
%3DP: 6% → 0% supp.
%3axis: no
%4axis; 97.7%

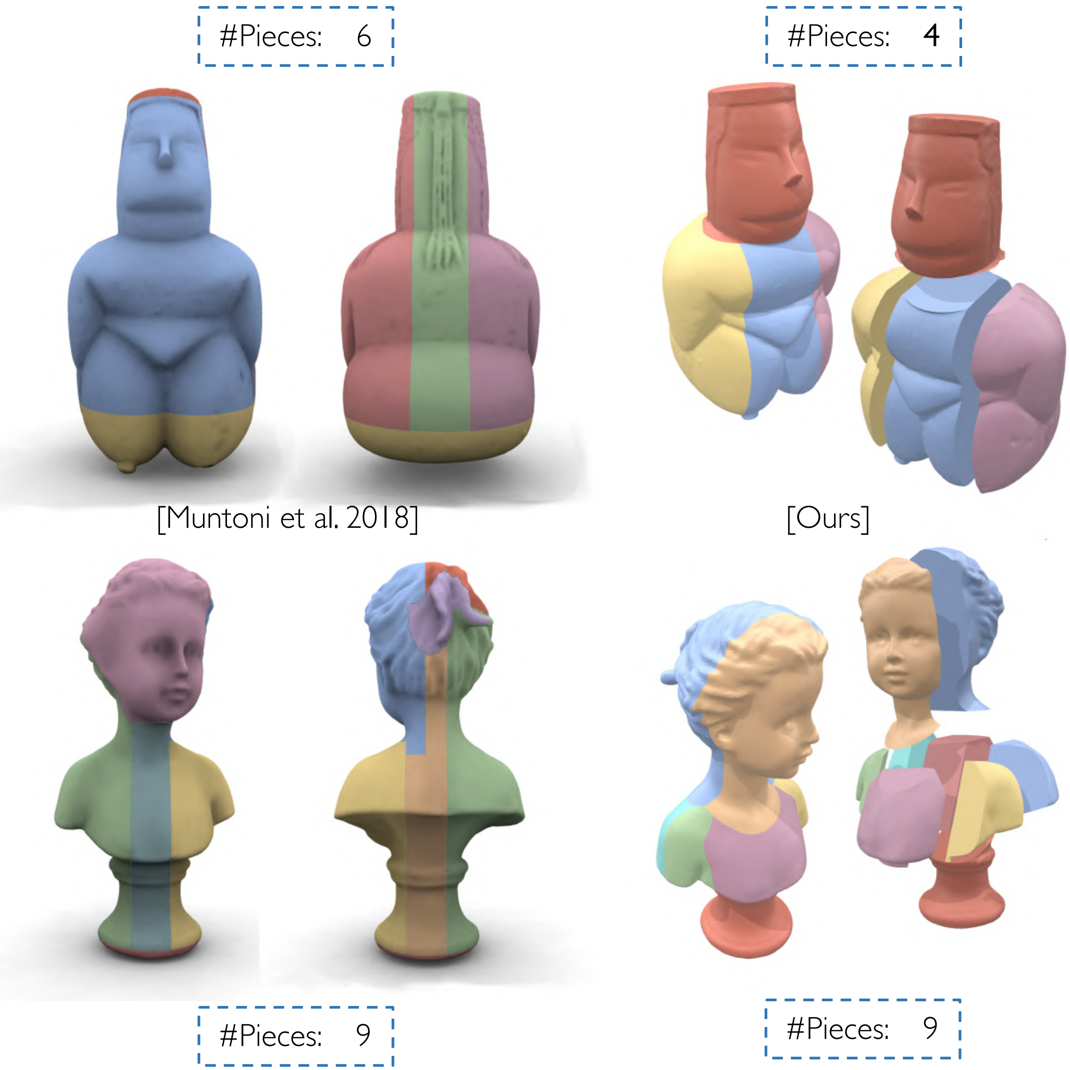


Fabricated results



bigger than the printing chamber

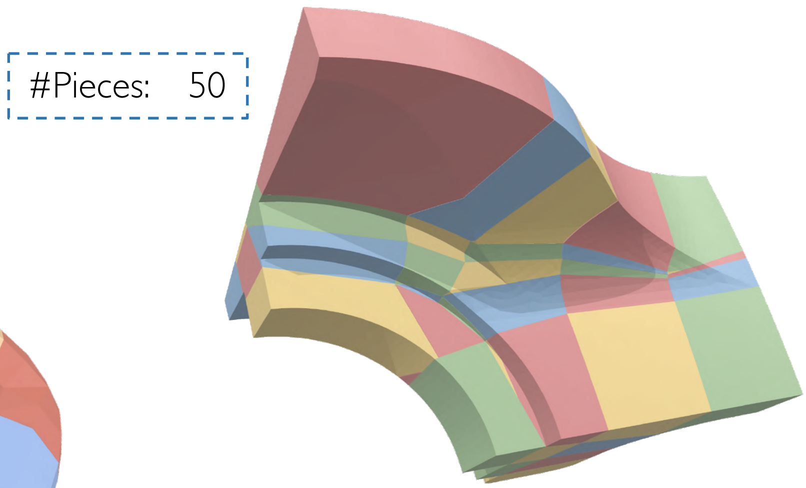
Comparison



- Comparable results
- Simpler subdivision algorithm
- Not suitable for 3-axis machining

Limitations

- 3-axis machining
- High number of portions



About this work

Polycube-based Decomposition for Fabrication

F. Fanni, G. Cherchi, R. Scateni
University of Cagliari (IT)

Work presented at STAG 2017
Catania (IT)

Fabrication Oriented Shape Decomposition Using Polycube Mapping

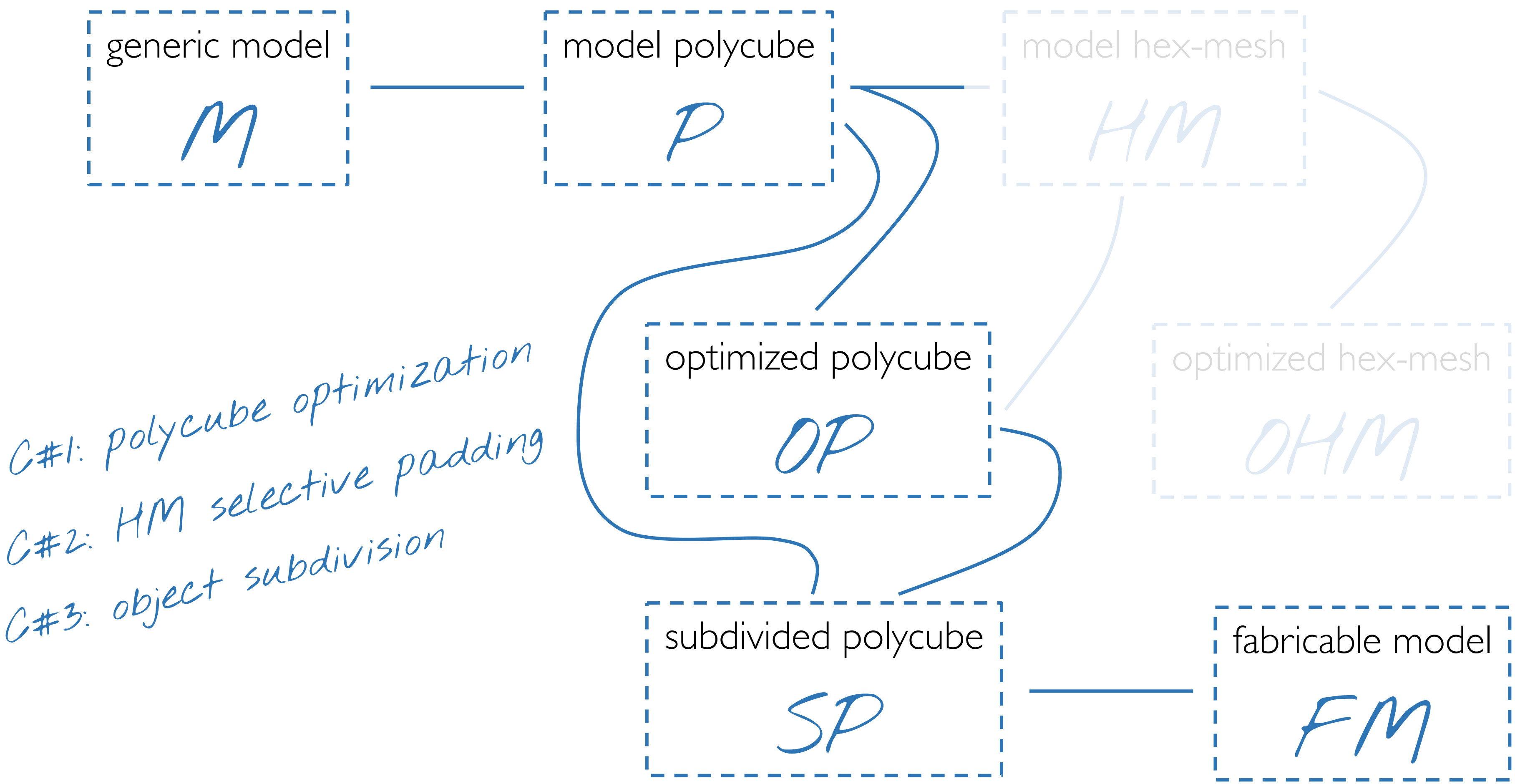
Computer & Graphics
Elsevier 2018

F. Fanni, G. Cherchi, R. Scateni, A. Tola
University of Cagliari (IT)

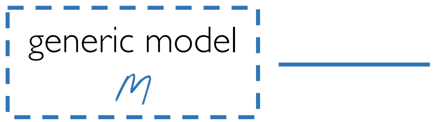
A. Muntoni
CNR-ISTI, Pisa (IT)

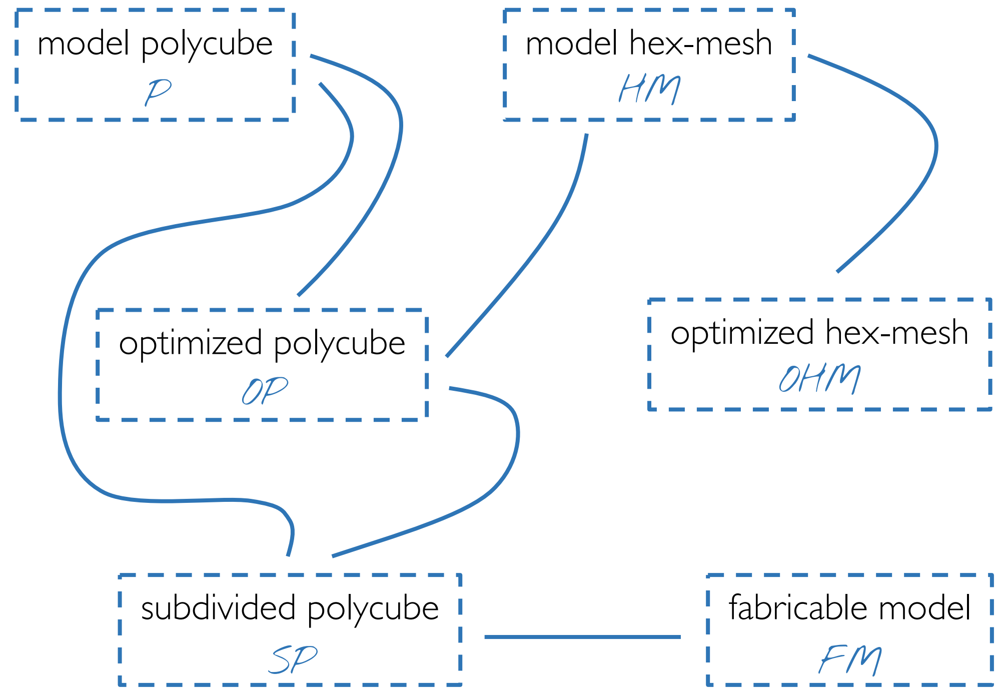
Conclusions and Future Works

Conclusions

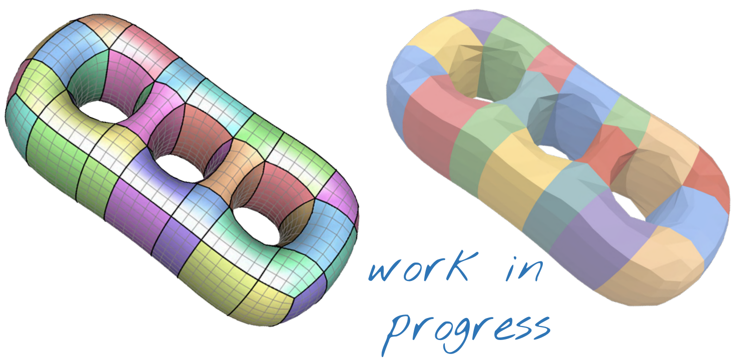


Future work





- Reduce the number of domains



- Improve the corner pairing step
- Automatic λ selection
- Reach the highest possible quality

- 3-axis milling constraints
- Reduce the number of pieces (clusters)

Other Ph.D. activities

Other Ph.D. activities

Publications:

- Polycube Simplification for Coarse Layouts of Surfaces and Volumes (CGF 2016)
- Polycube-based Decomposition for Fabrication (STAG proc. 2017)
- A Seamless Pipeline for the Acquisition of the Body Shape: the Virtuoso Case Study (STAG proc. 2017)
- ChIP: teaching coding in primary schools (CHI proc. 2017)
- Fabrication Oriented Shape Decomposition Using Polycube Mapping (C&G 2018)
- Selective Padding for Polycube-based Hexahedral Meshing (CGF 2019)

International mobility:

- Visiting Ph.D. student at the INRIA Sophia Antipolis Mediterranée, France (sept - dec 2017)

Talks in conference and seminars:

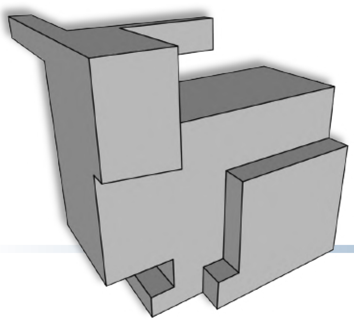
- Polycube Simplification for Coarse Layouts of Surfaces and Volumes (SGP 2016)
- Polycube Optimization - Generating coarse quad-layout via smart polycube quantization (STAG 2016)
- Polycube Simplification, Optimization and Remeshing (EG 2017)
- Polycube-based Decomposition for Fabrication (STAG 2017)
- Polycubes and related operations (INRIA 2017)

Posters:

- Polycube Optimization for Hex-meshing (DENIS 2016)
- Polycube Simplification, Optimization and Remeshing (EG 2017)

Teaching activities:

- Algorithms and Data Structure Lab. Course 2016 – 2018
- Various seminars and mini-courses



Thanks!