Progressive Embedding

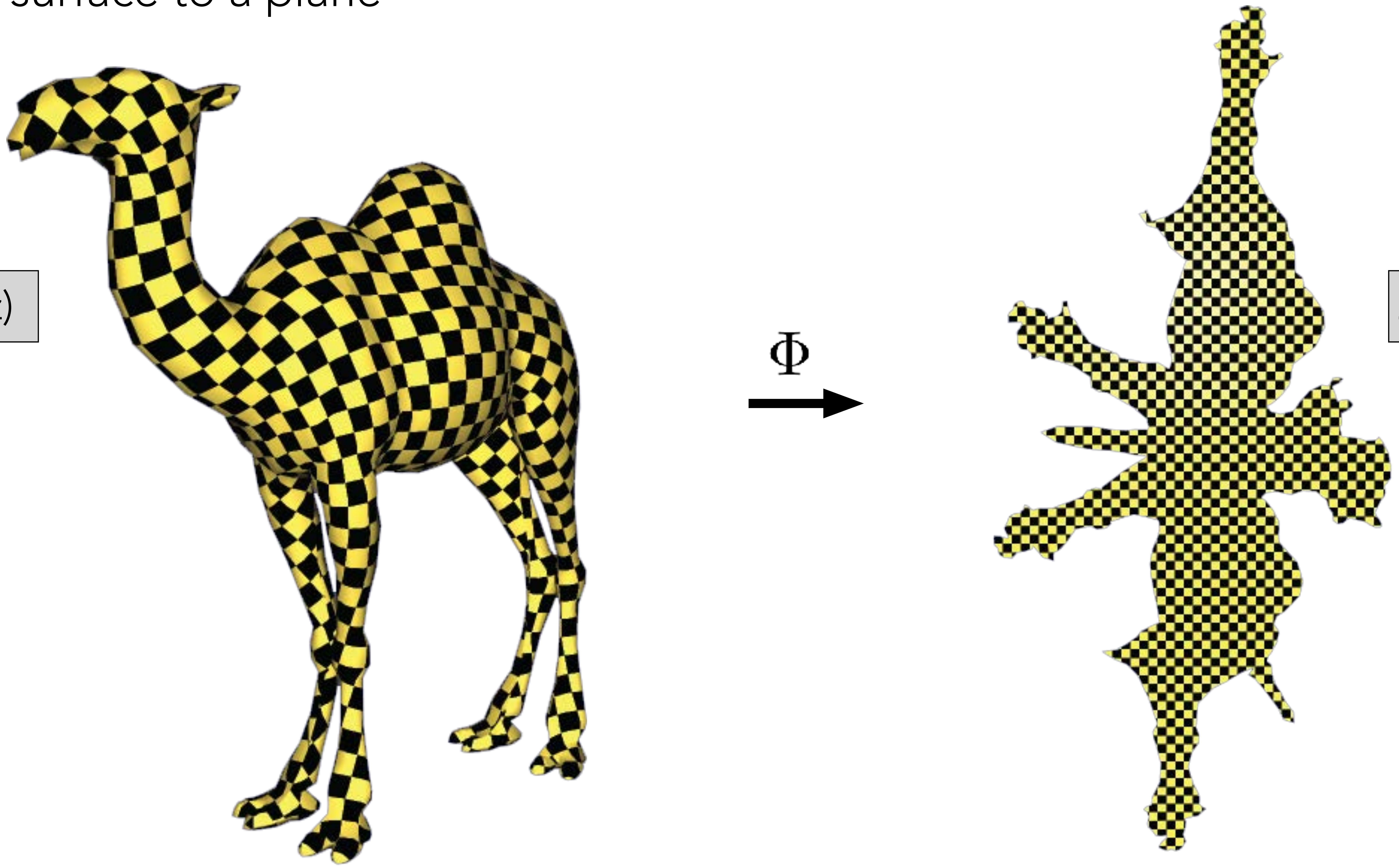
Hanxiao Shen, Zhongshi Jiang, Denis Zorin, Daniele Panozzo

Geometric Computing Lab, New York University

Surface Parametrization

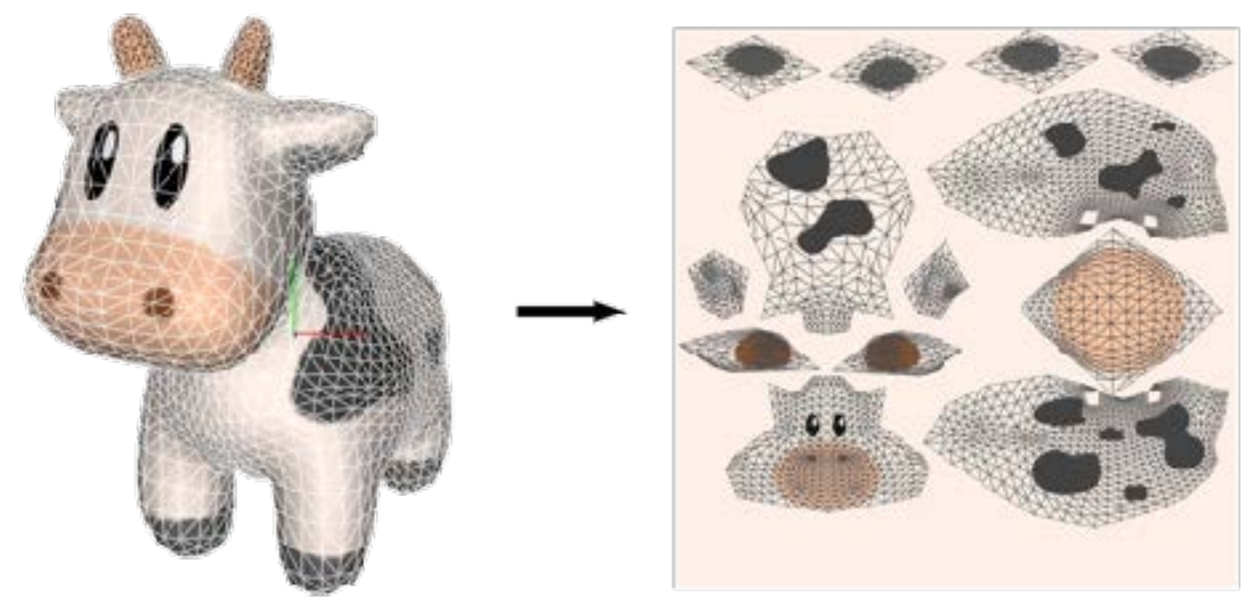
Flatten a surface to a plane

3D Mesh (x,y,z)

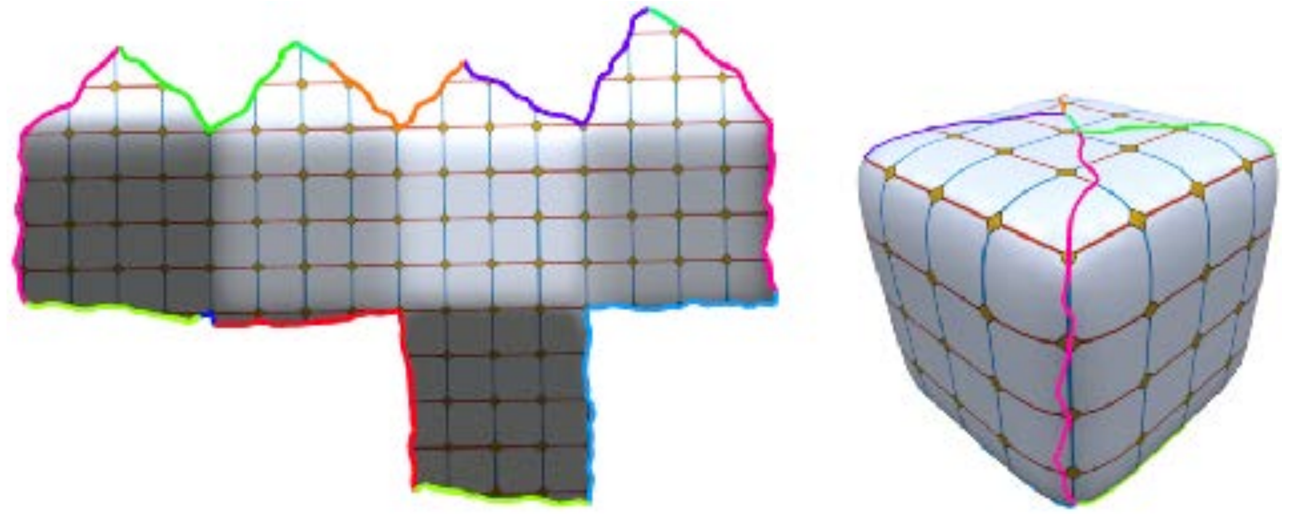


2D Parametric domain (u,v)

Applications

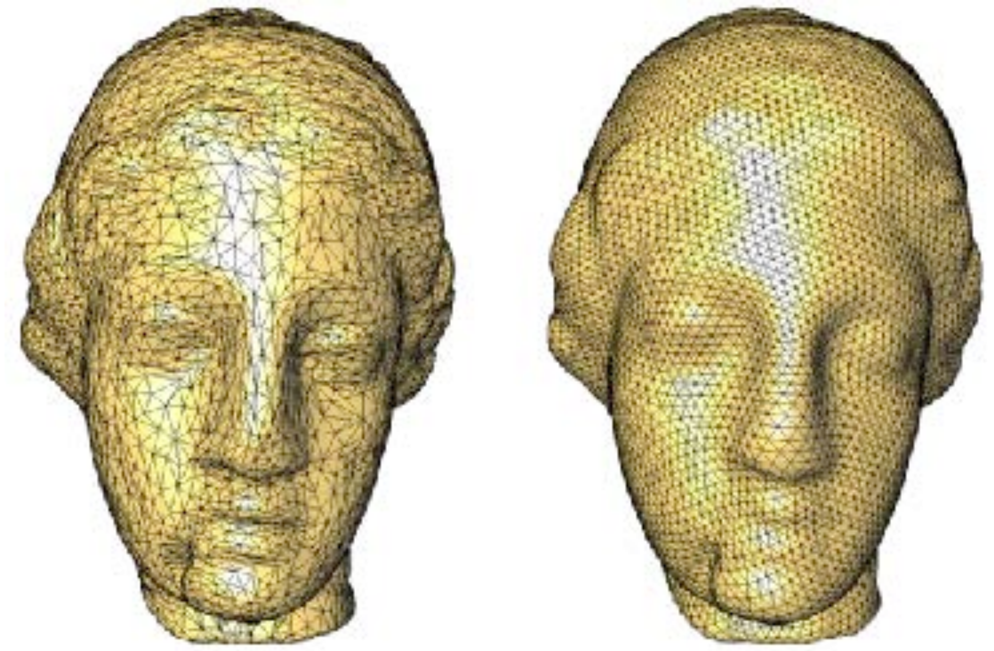


Texture mapping



[Bommes et al. 2012]

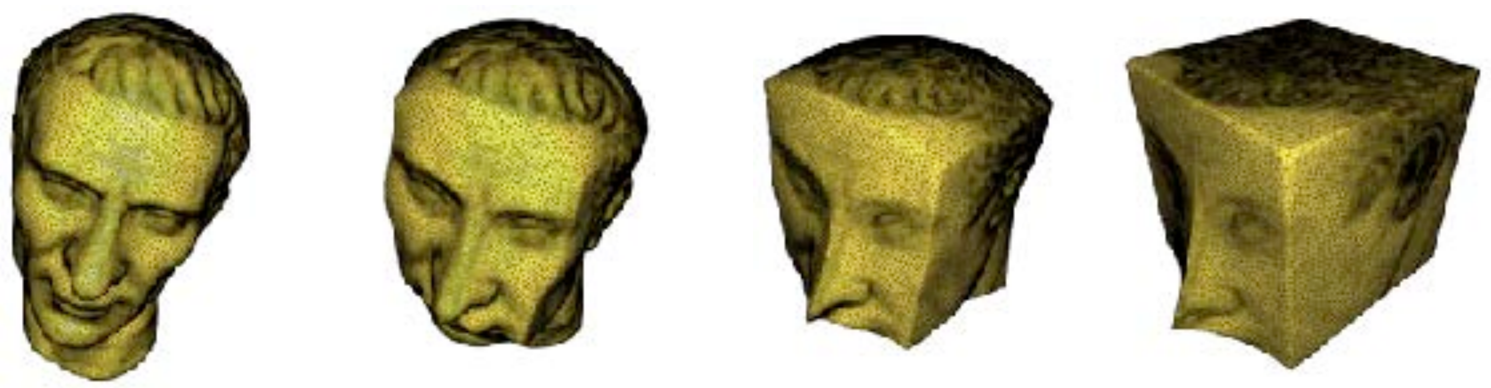
Quadrangulation



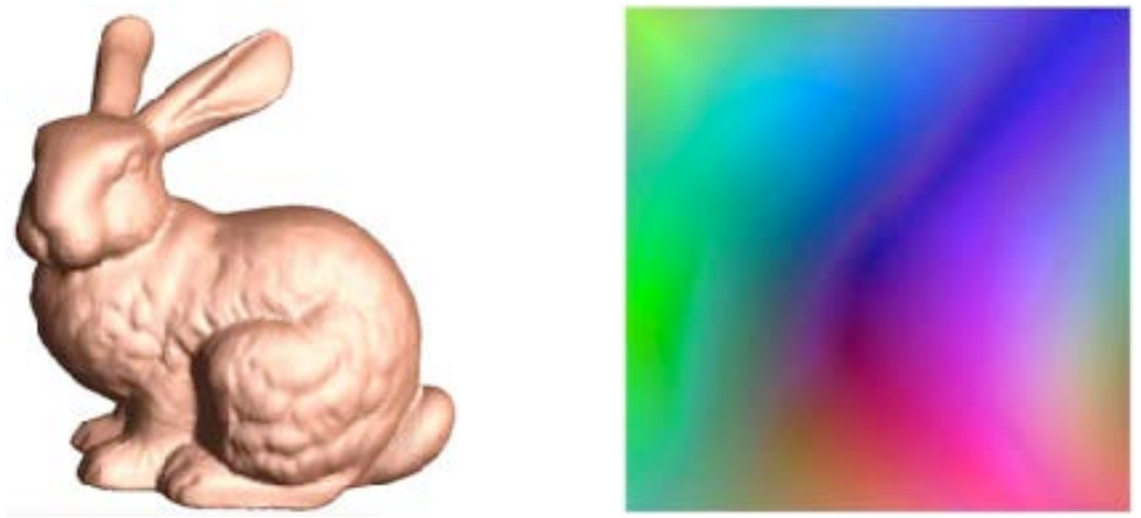
[Botsch et.al 2010]

Remeshing

Cartography



Shape Interpolation

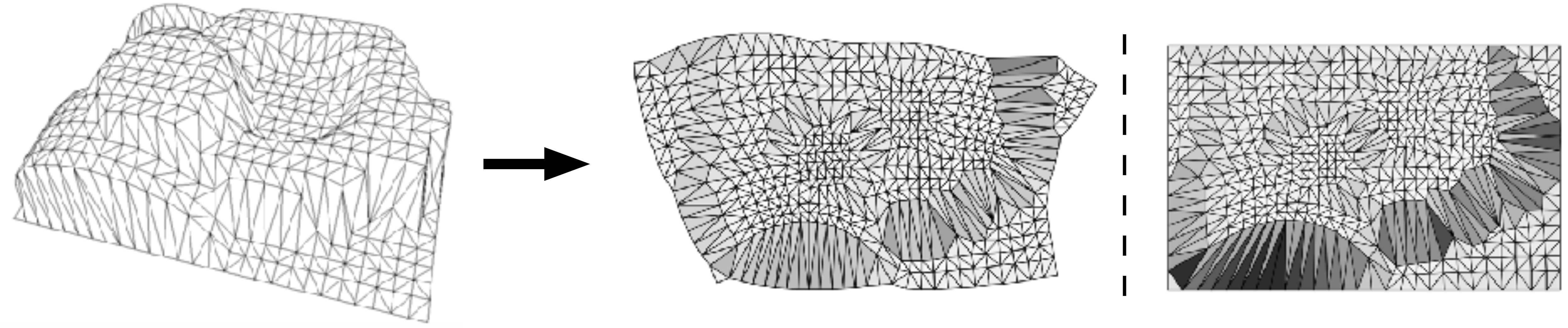


[Gu et al. 2002]

Compression

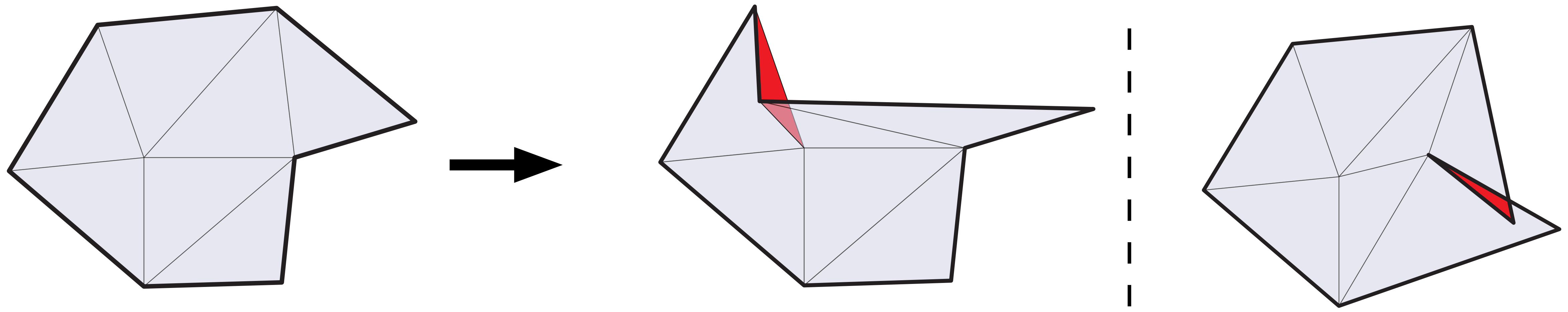
Desirable Properties

Low Geometric Distortion

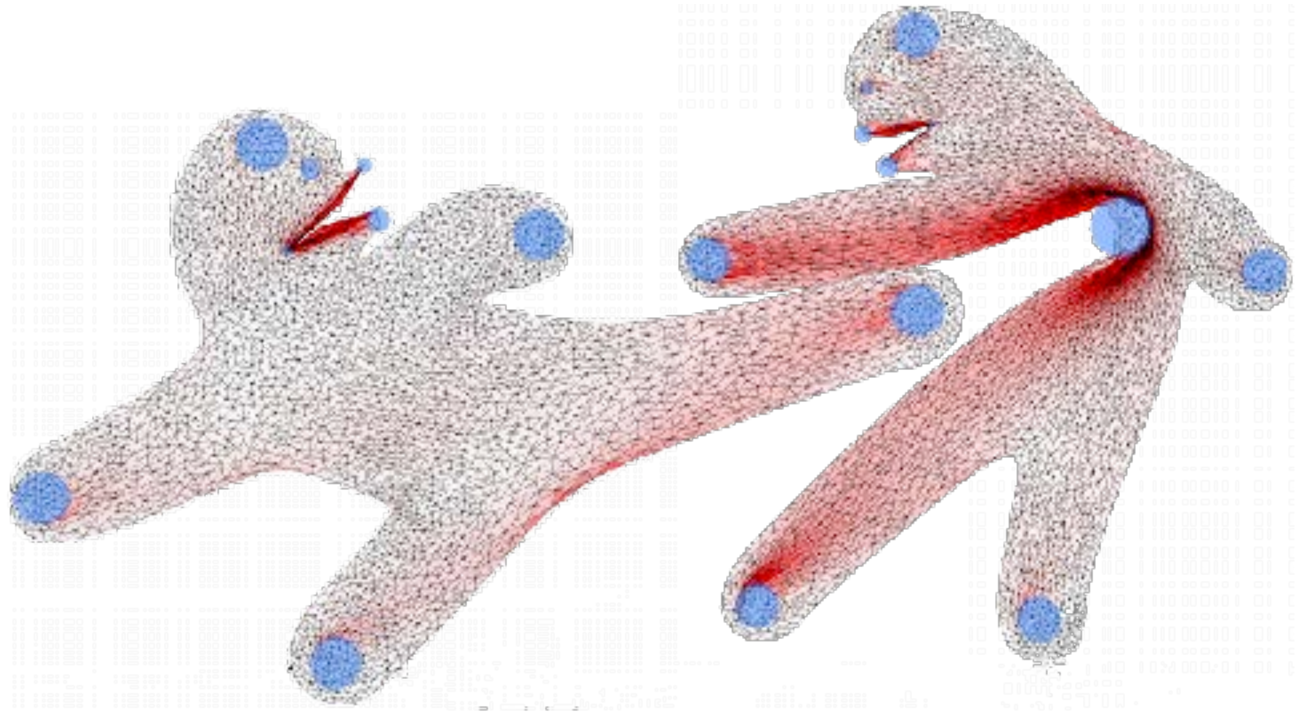


[Hormann & Greiner et al. 2000]

Local/Global Bijectivity



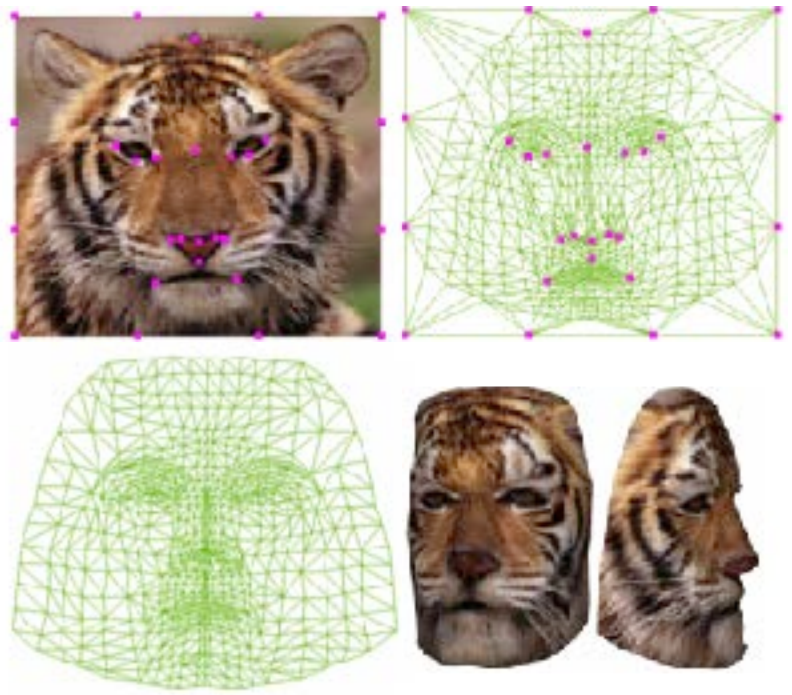
Control with Positional Constraints



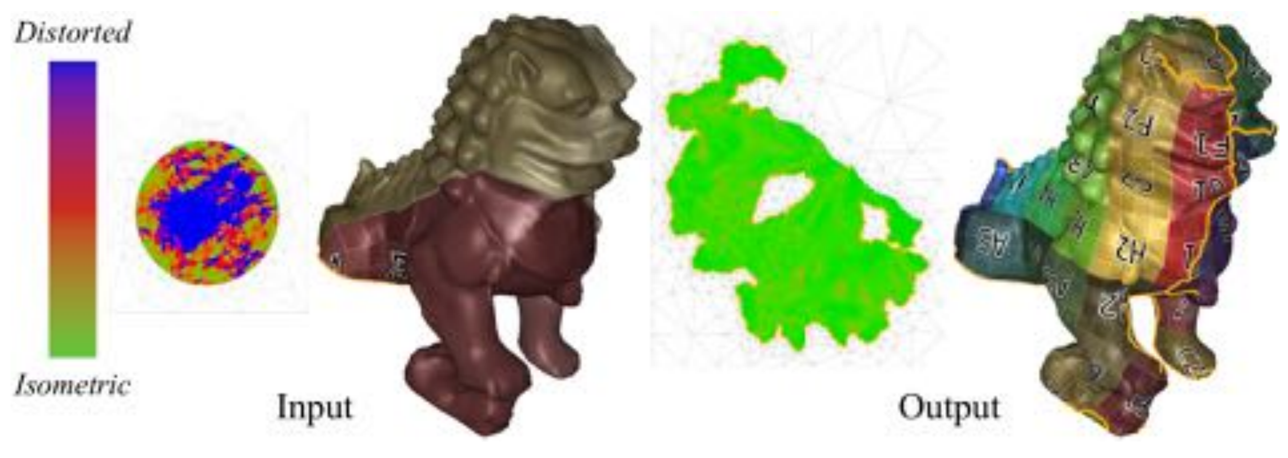
[Schüller et al. 2013]

Efficiency & Scalability

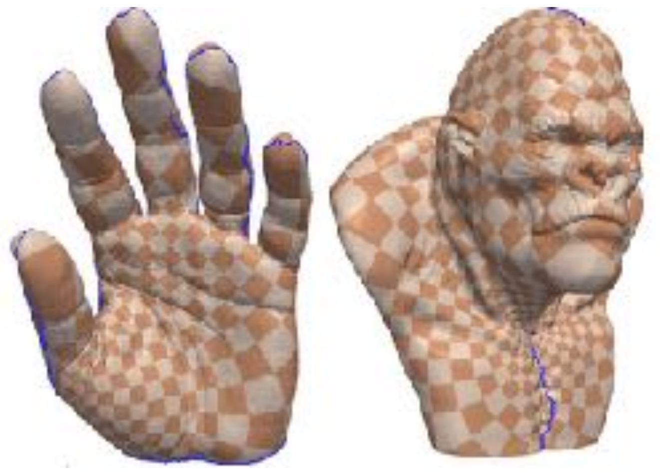
Distortion-Minimizing Mappings



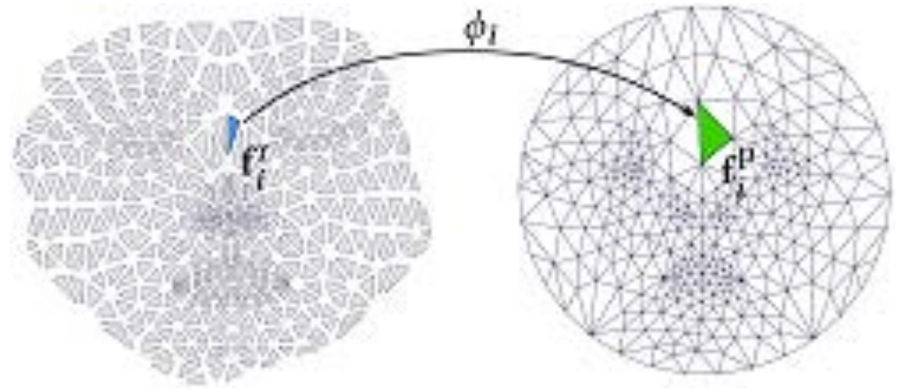
[Kraevoy et al. 2003]



[Smith and Schaefer 2015]

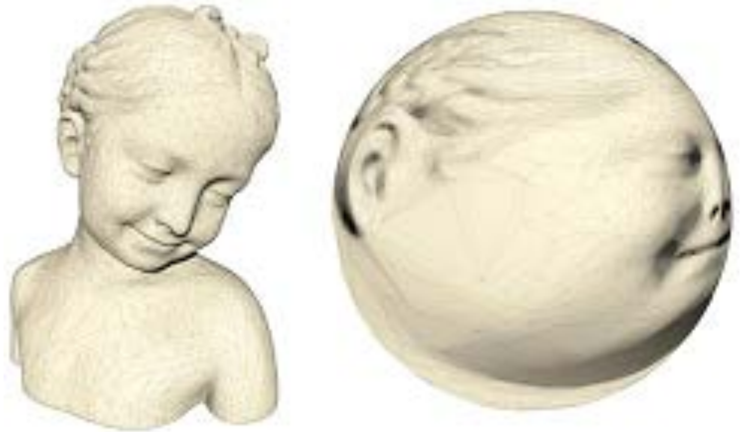


[Claici et al. 2017]



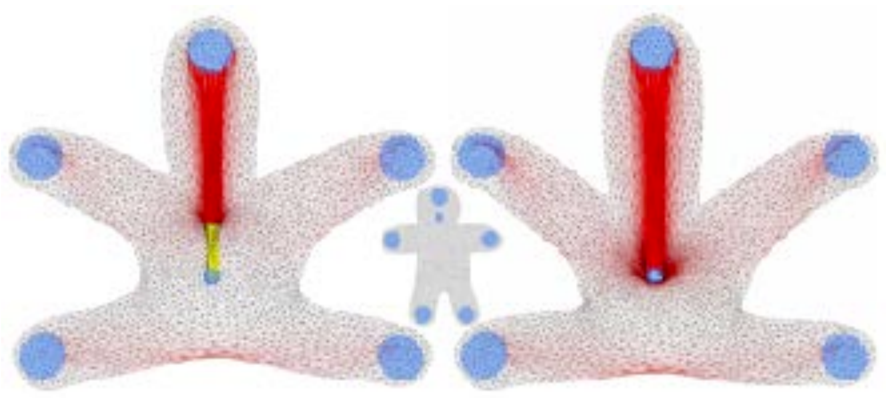
[Liu et al. 2018]

Tutte Embedding

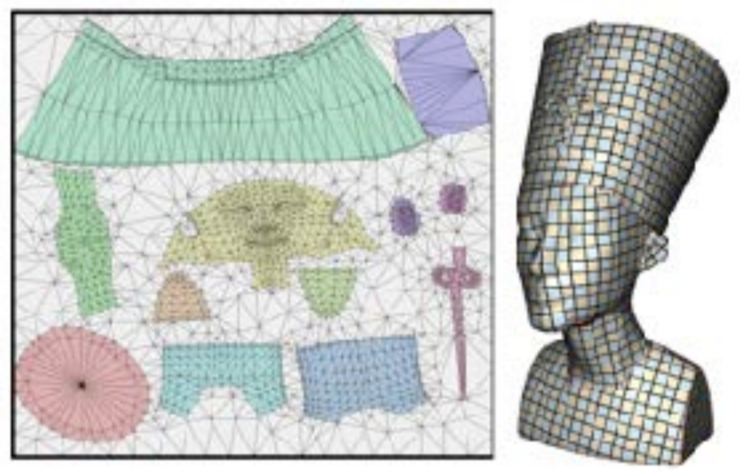


[Wang et al. 2016]

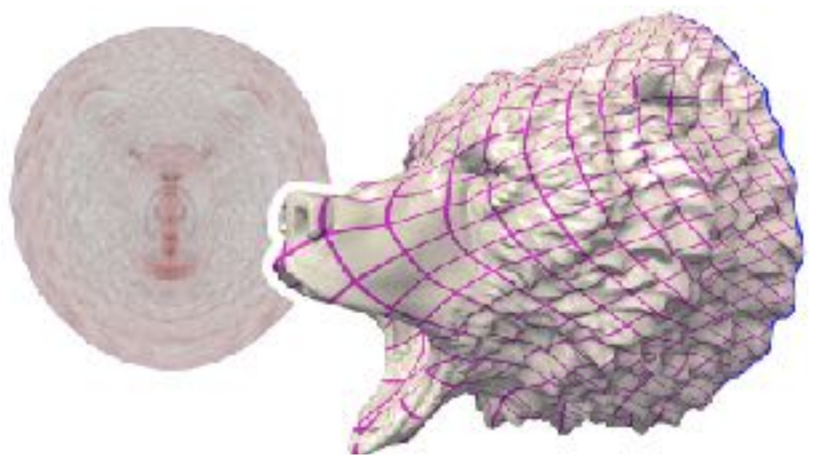
[Rabinovich et al. 2016]



[Schüller et al. 2013]



[Jiang et al. 2017]

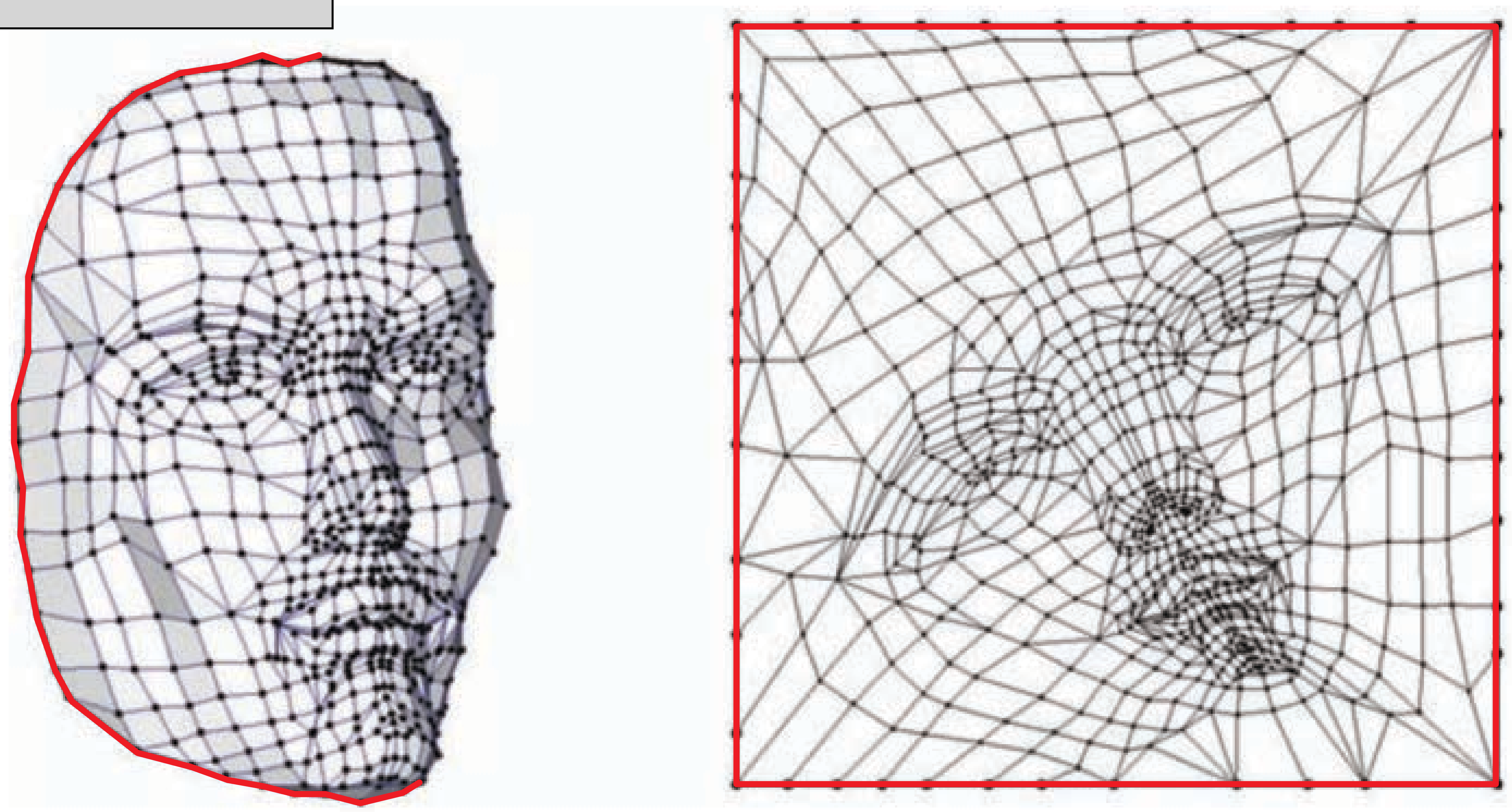


[Shtengel et al. 2017]

[Fu et al. 2015]

Tutte Embedding

How to draw a graph [Tutte. 1963]



Map a disk mesh to the interior of **convex** boundary

Guarantee for bijectivity

How about floating point coordinates?

(Meshes are assumed to be 3-connected)

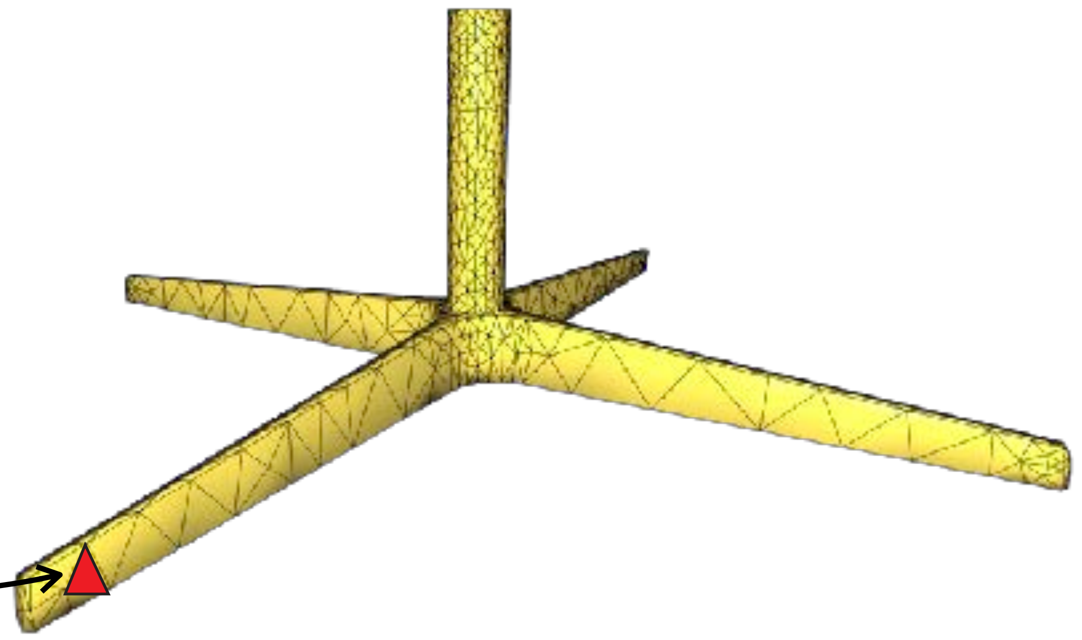
Robustness Test

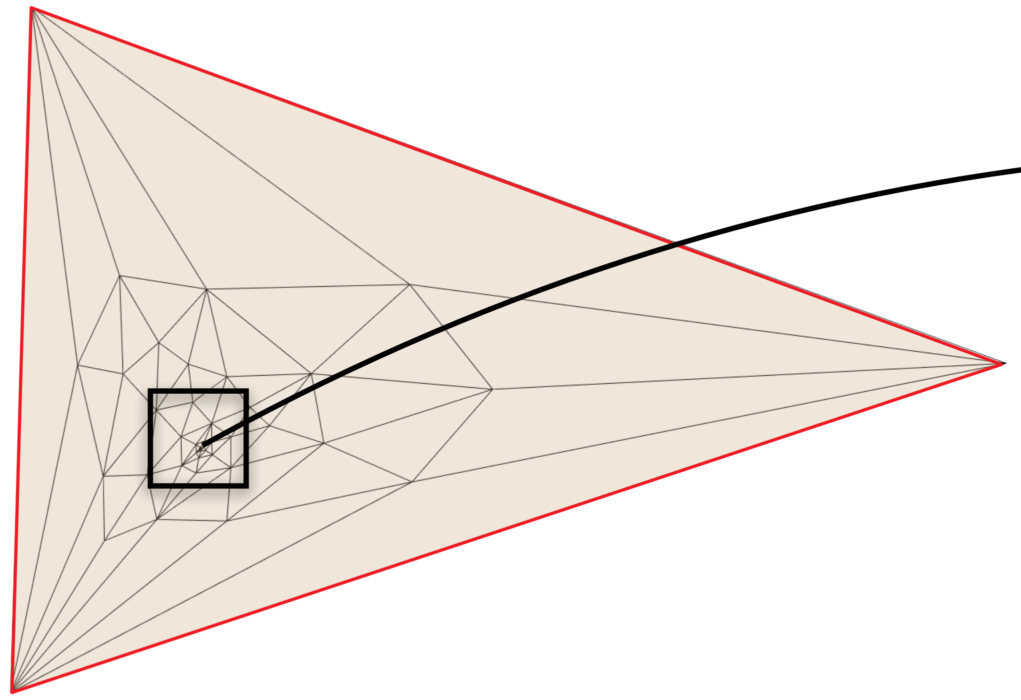
The genus 0 models from **Thingi10k** dataset [Zhou & Jacobson 2016]

Floating Point Implementation of Tutte Embedding from **libigl** [Jacobson et al. 2016]

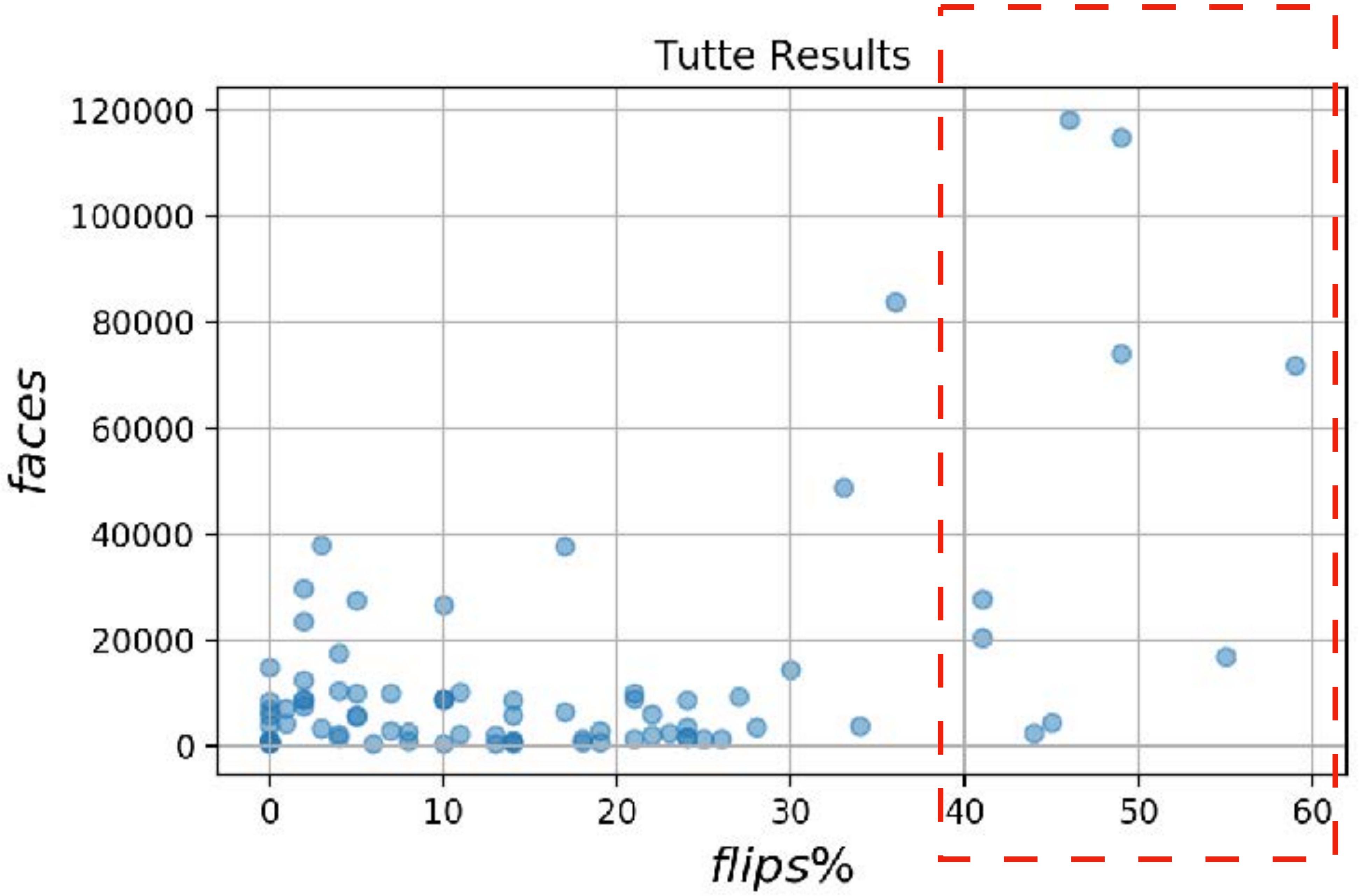
Robustness Test

Randomly drop one triangle





70 flipped elements



80 / 2718 models have flipped elements

Fixing Tutte Embedding

Obvious Ways to Try

Multi-precision

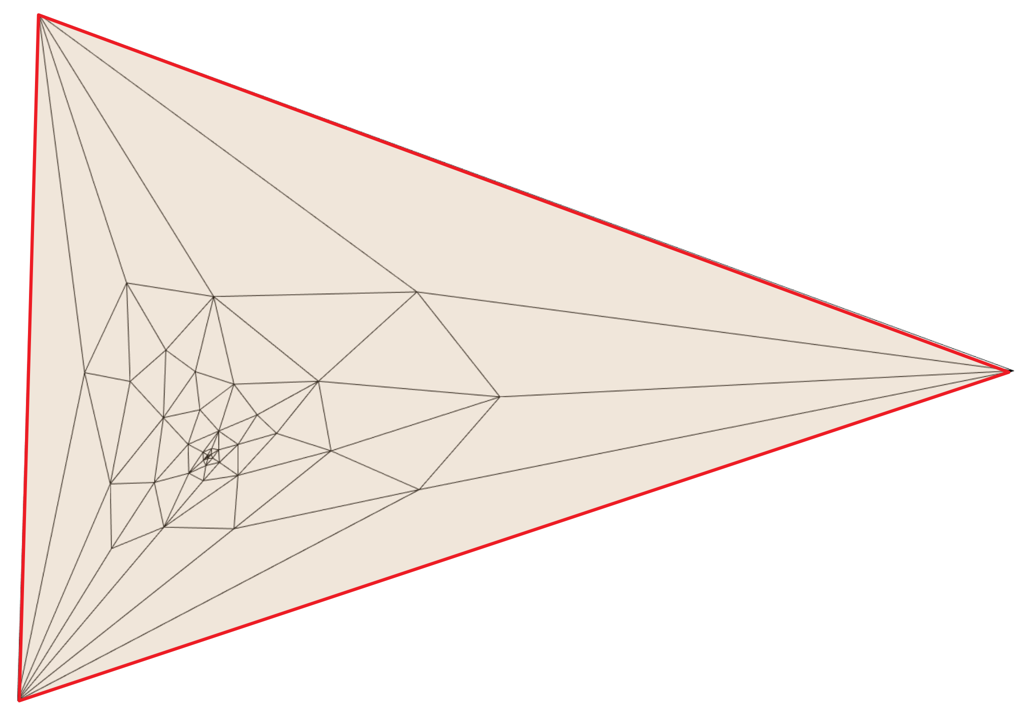
Snap Rounding

Fixing Tutte Embedding

Obvious Ways to Try: Multi-precision

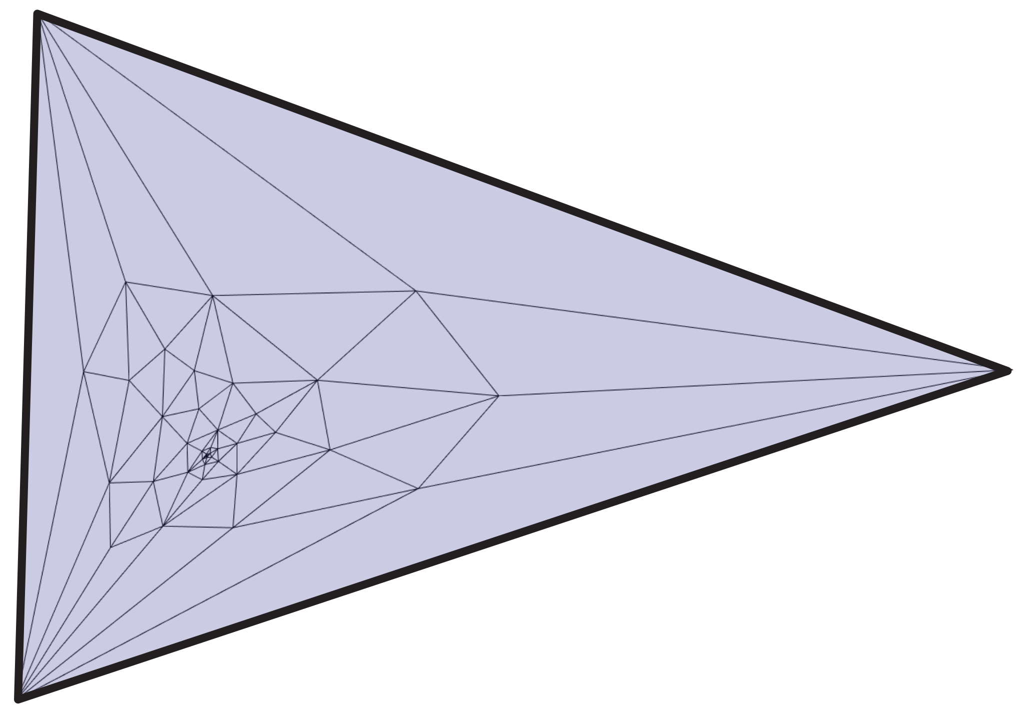
Floating Point Arithmetic

MPFR The *MPFR* library [Fousse et al. 2007]



64 bits

70 flipped elements



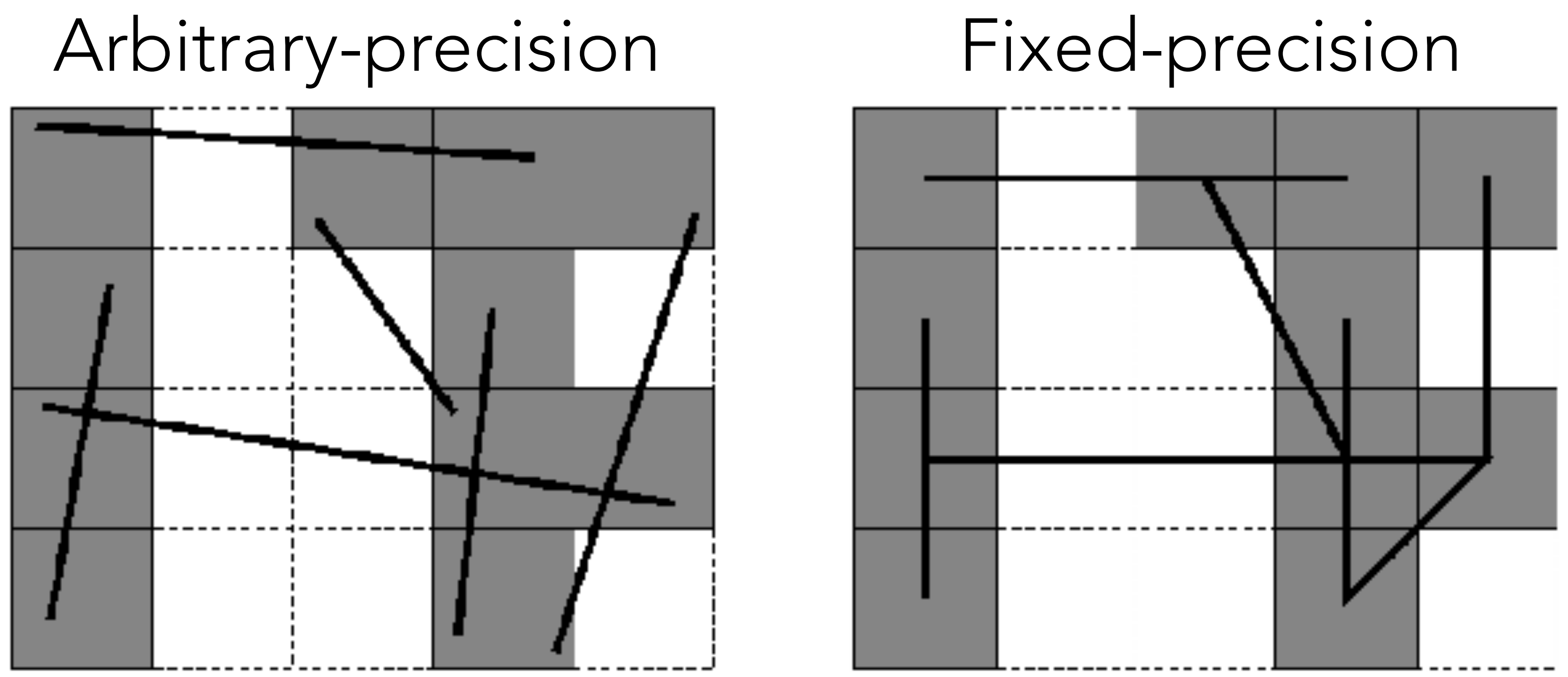
128 bits

0 flipped elements

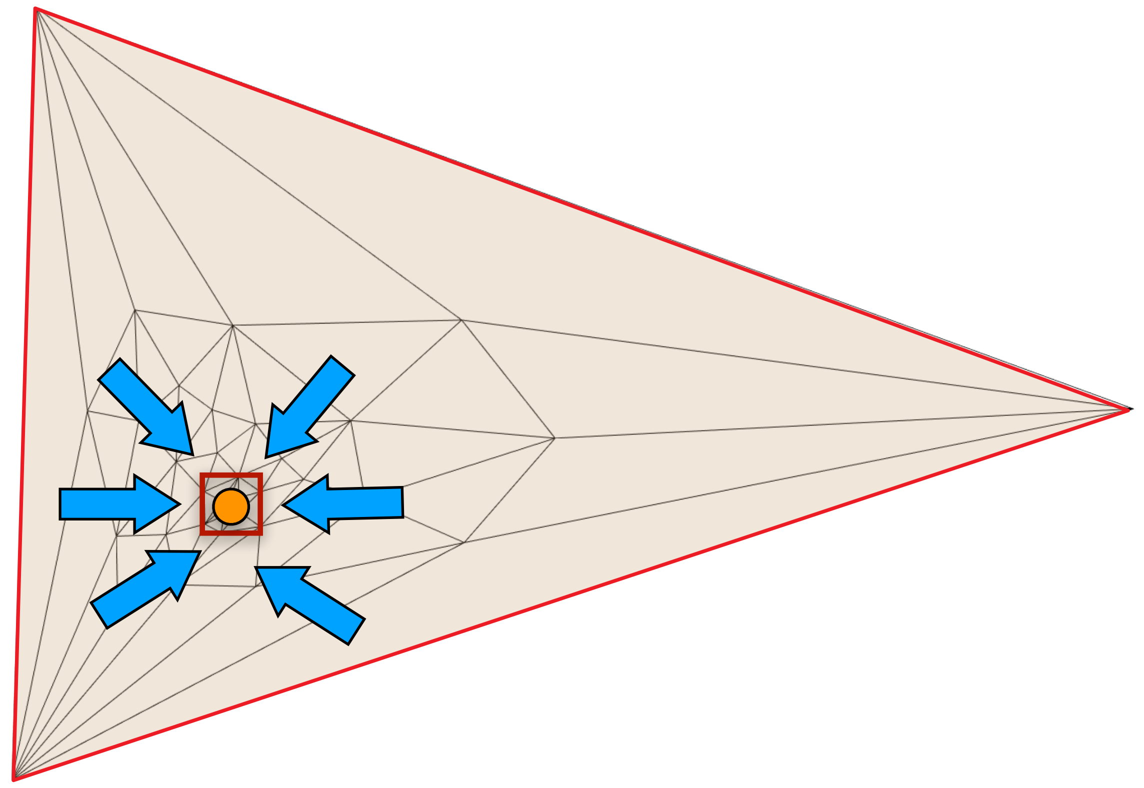
- Not usable for downstream applications.
- Naive rounding strategy may lead to flips!

Fixing Tutte Embedding

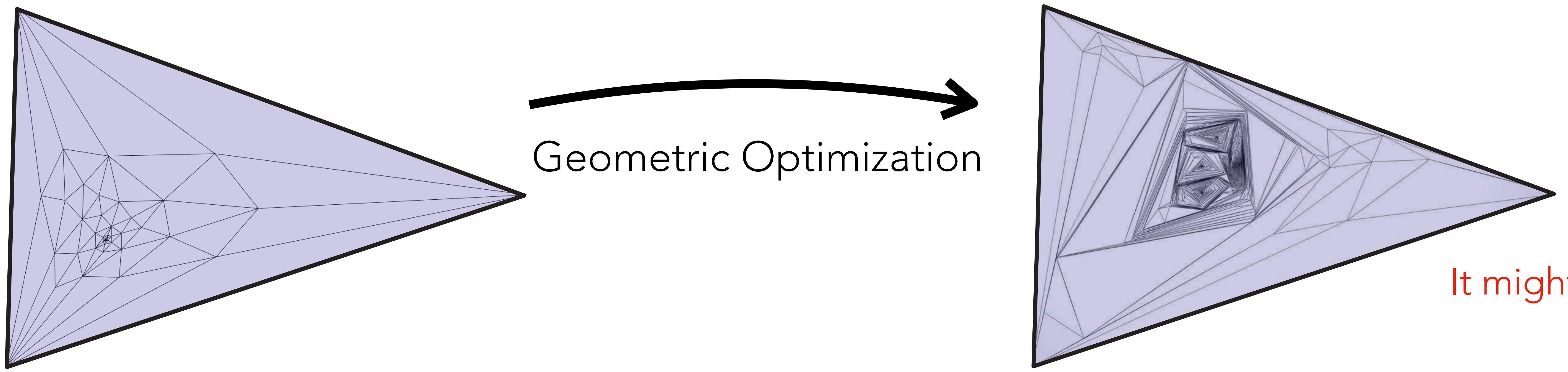
Obvious Ways to Try: Snap Rounding



Snap Rounding [Packer 2018]



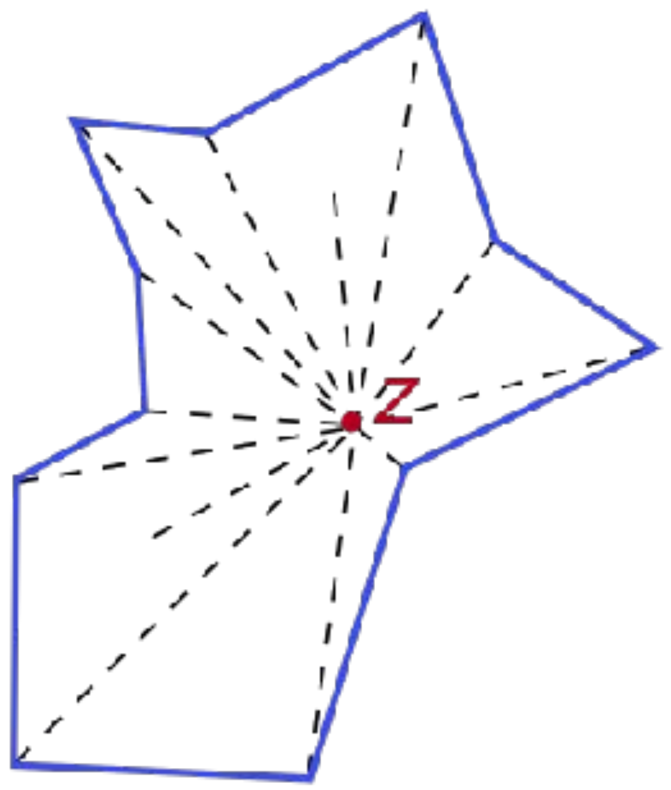
A large part of mesh is snapped to a single point!

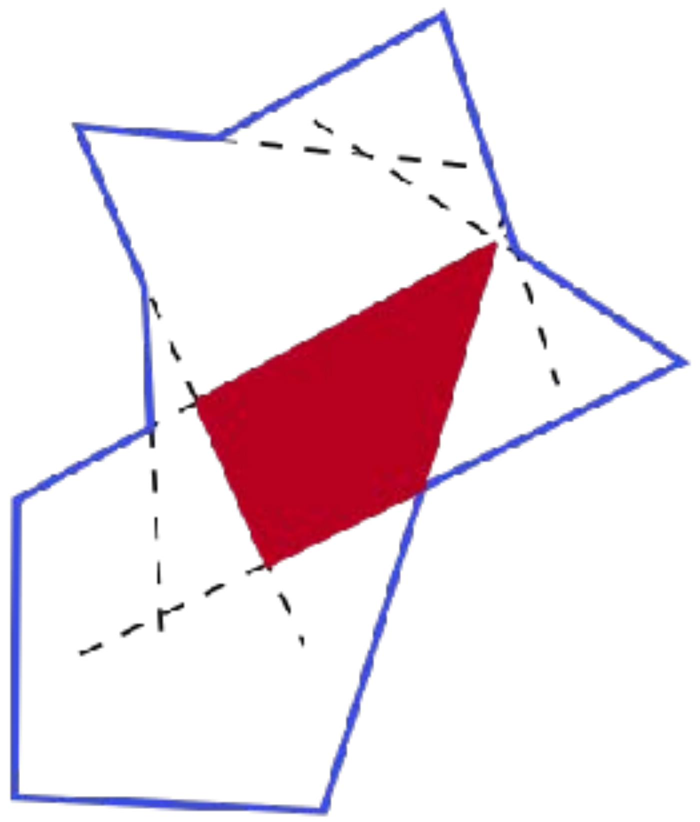


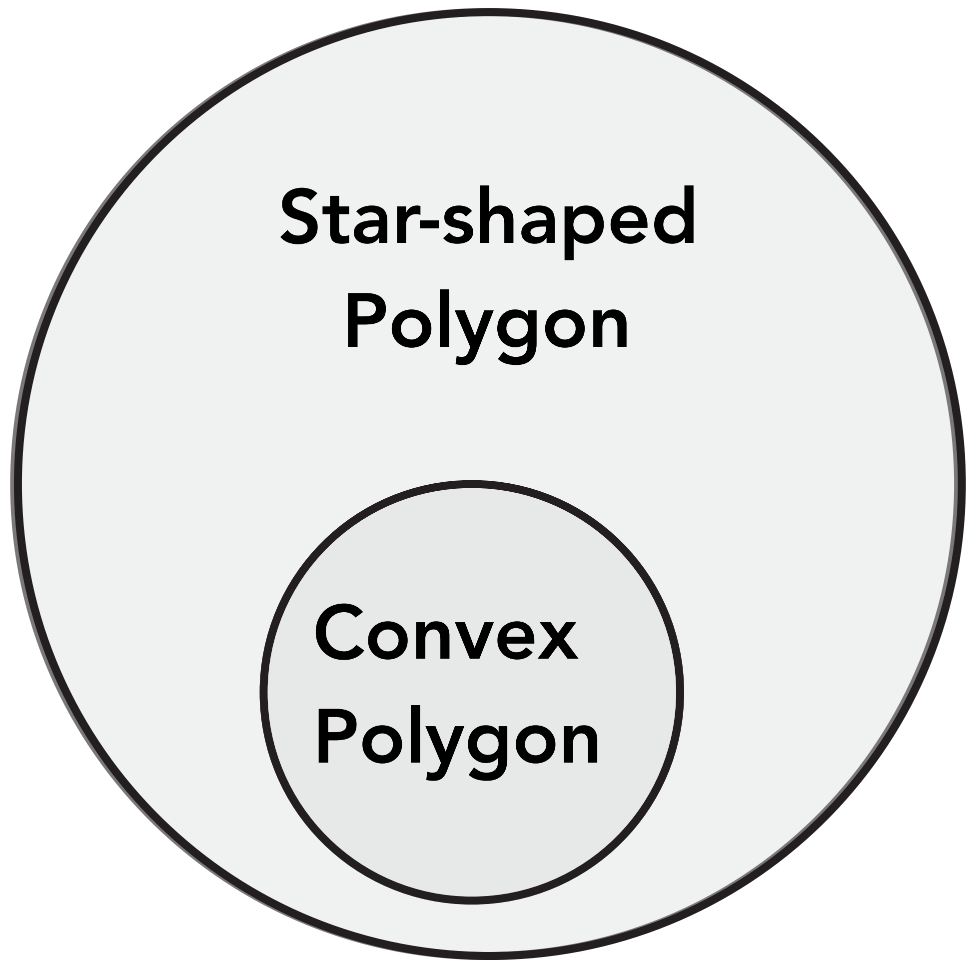
It might stuck during update!

Problem Formulation

Given a possibly invalid embedding with convex boundary for a disk-like surface mesh as input, how do we generate an inversion-free embedding using floating point coordinates?



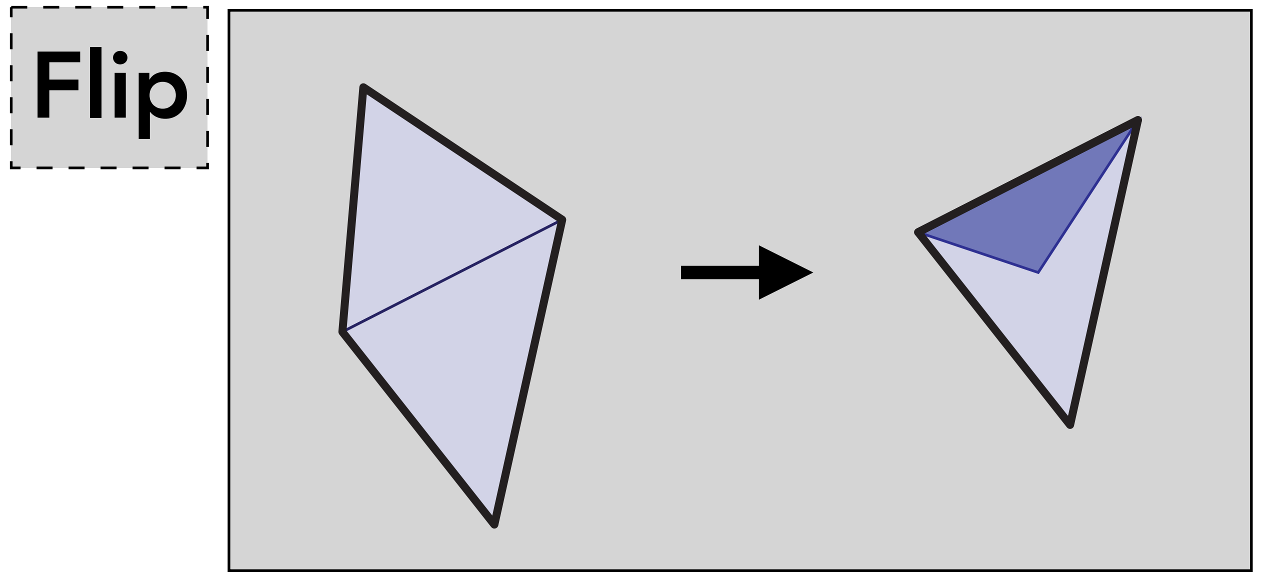




Invalid Element

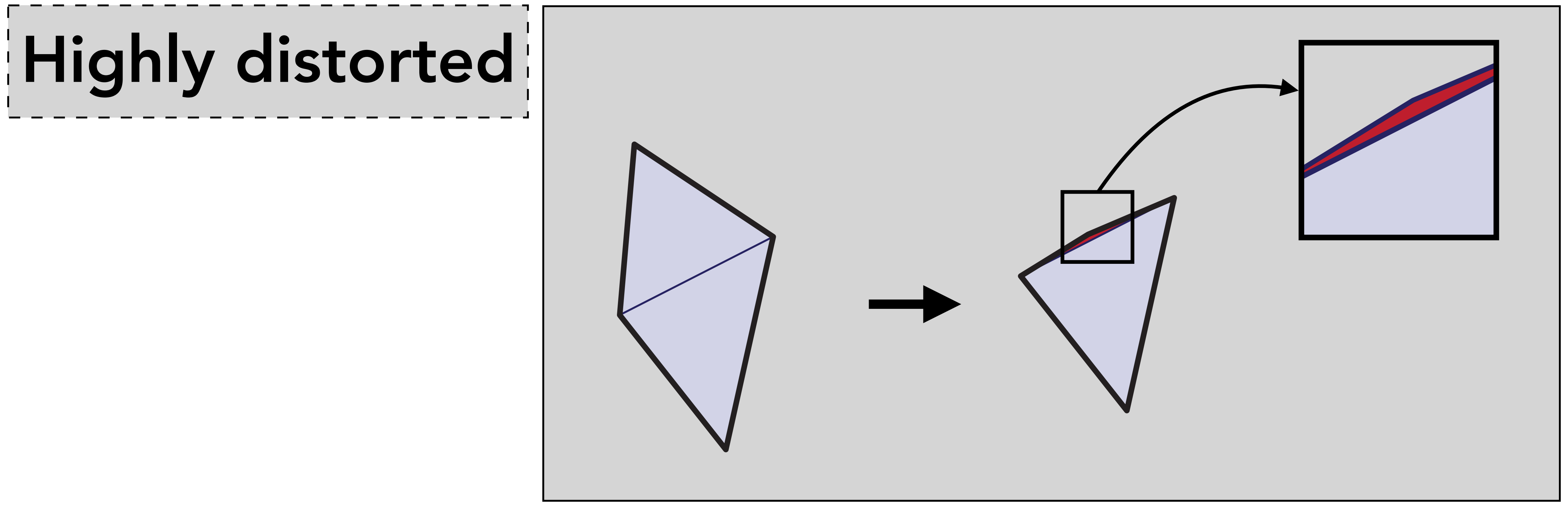
A triangle is said to be invalid if

1. Its signed area is negative



* Exact floating point predicates from CGAL [Brönnimann et al. 2018]

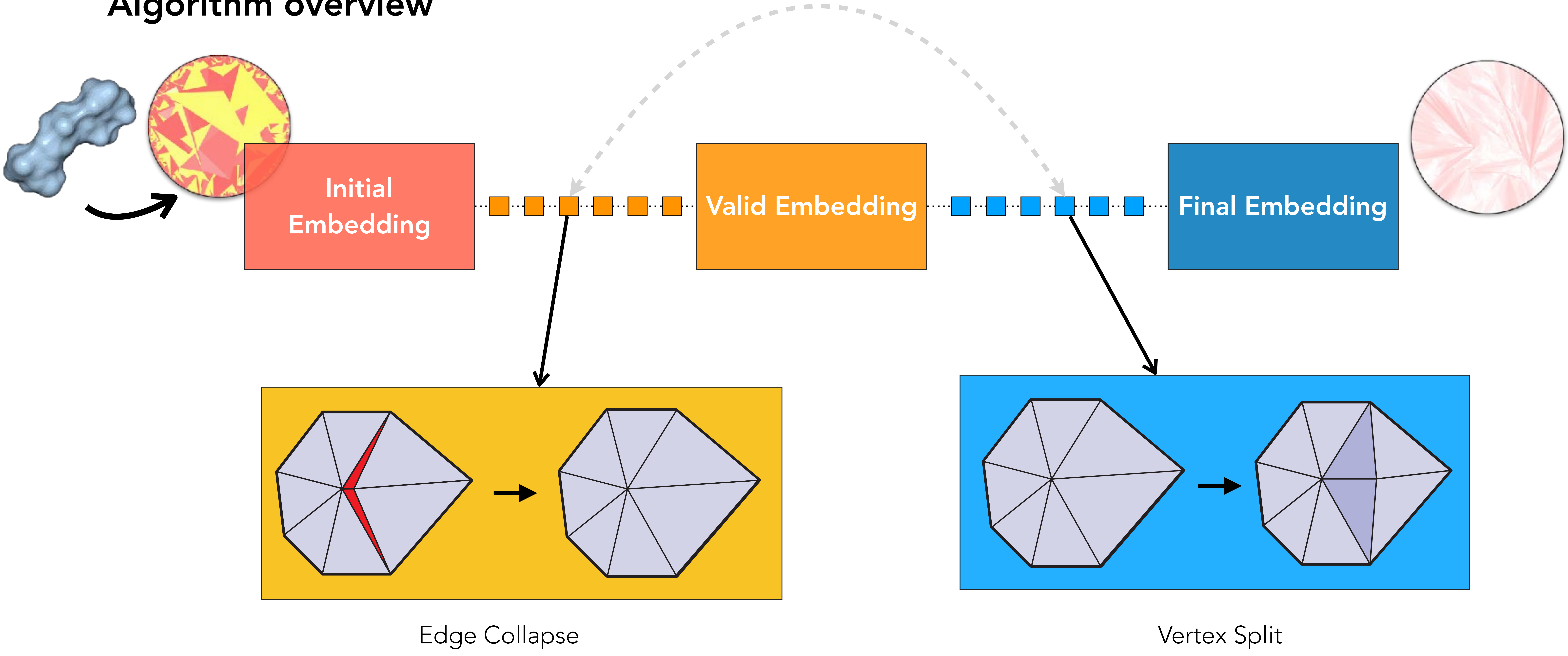
2. Its quality measure is below a threshold



* Symmetric Dirichlet Energy [Smith & Schaefer 2015]

Progressive Embedding

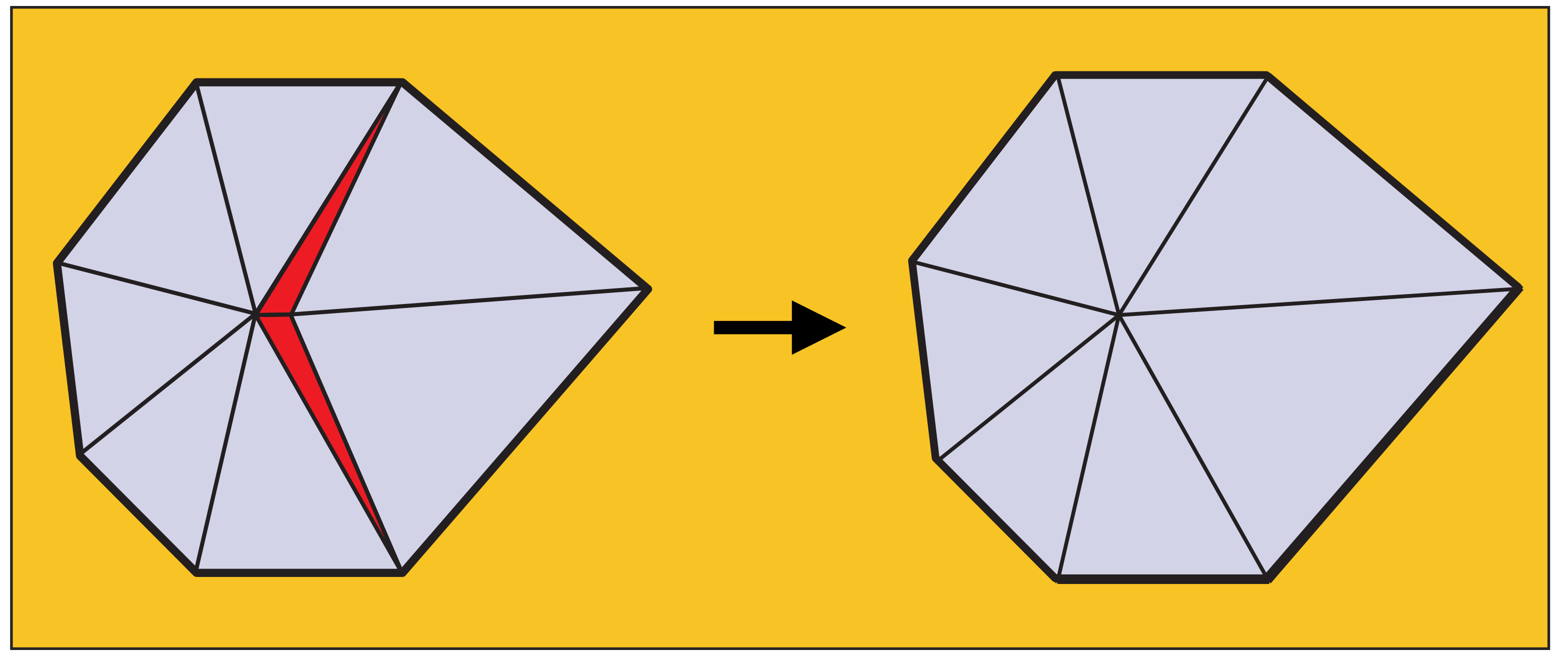
Algorithm overview



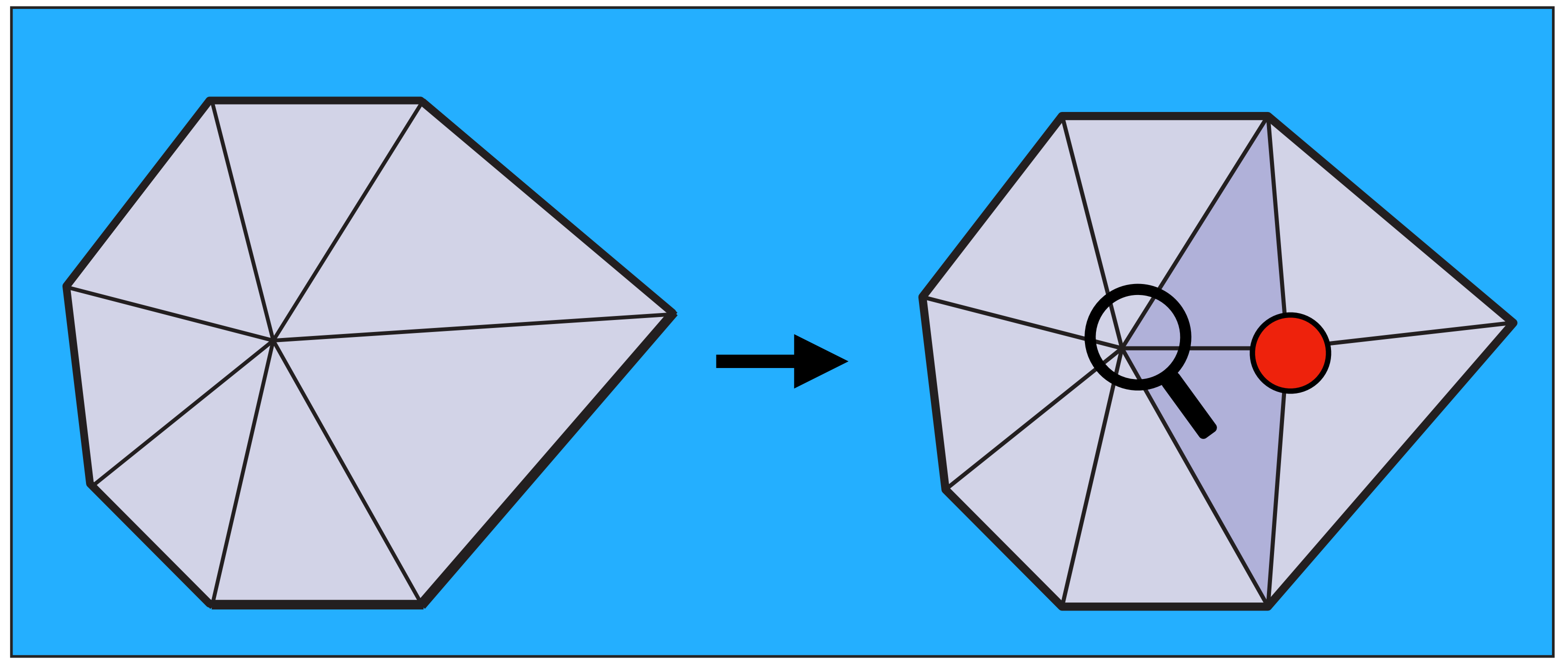
Basic Operations

Progressive Meshes
[Hoppe 1996]

- **Edge Collapse**: Takes two connected vertices and replaces them with a single vertex.

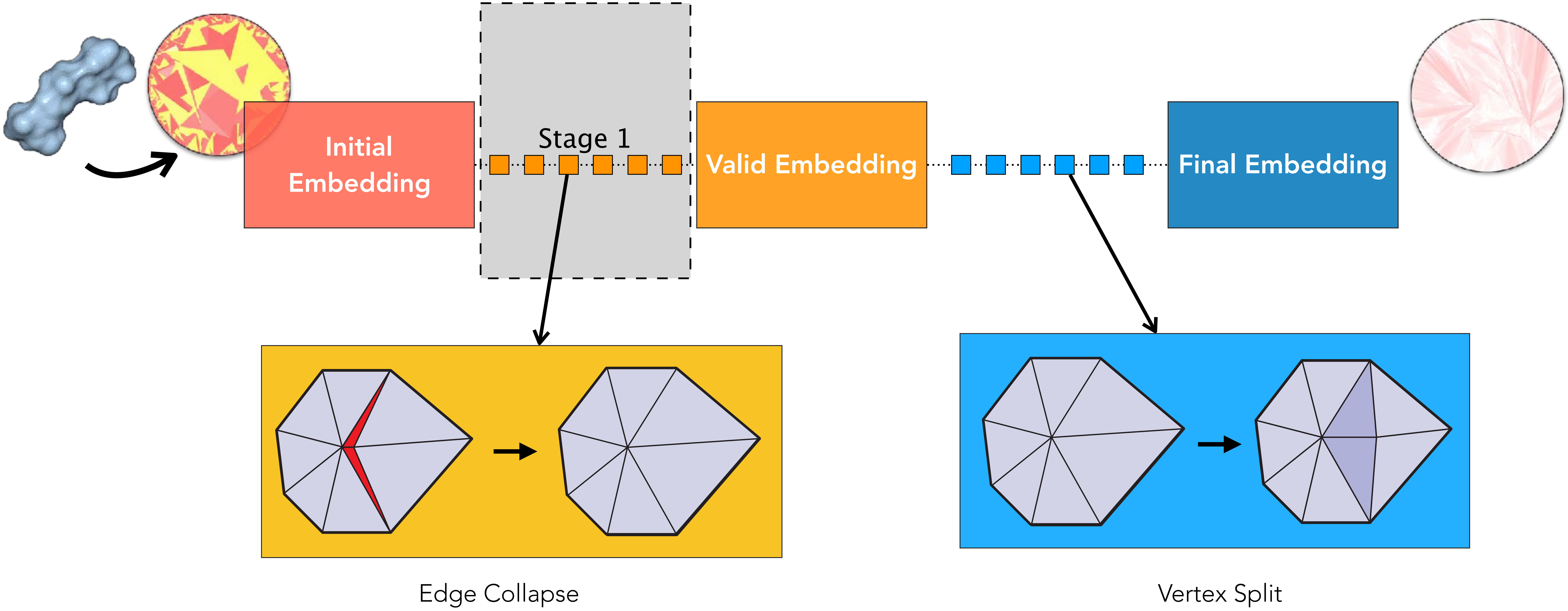


- **Vertex Split**: Divides the vertex into two new vertices, creates two new triangles



Progressive Embedding

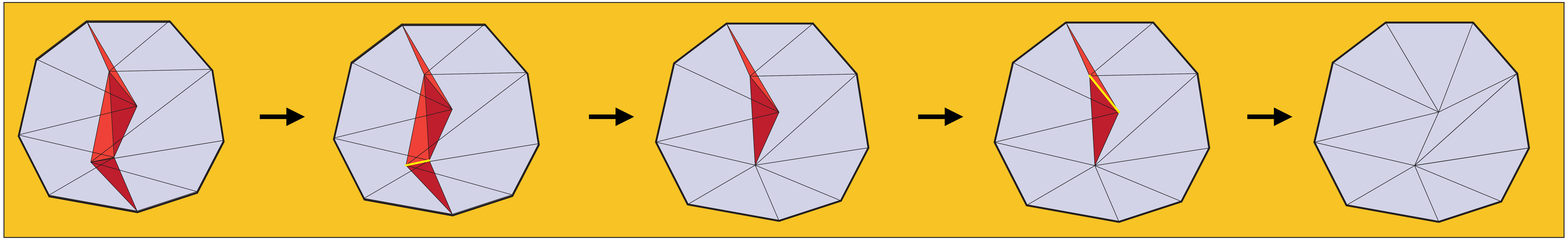
Algorithm overview



Stage 1: Collapse

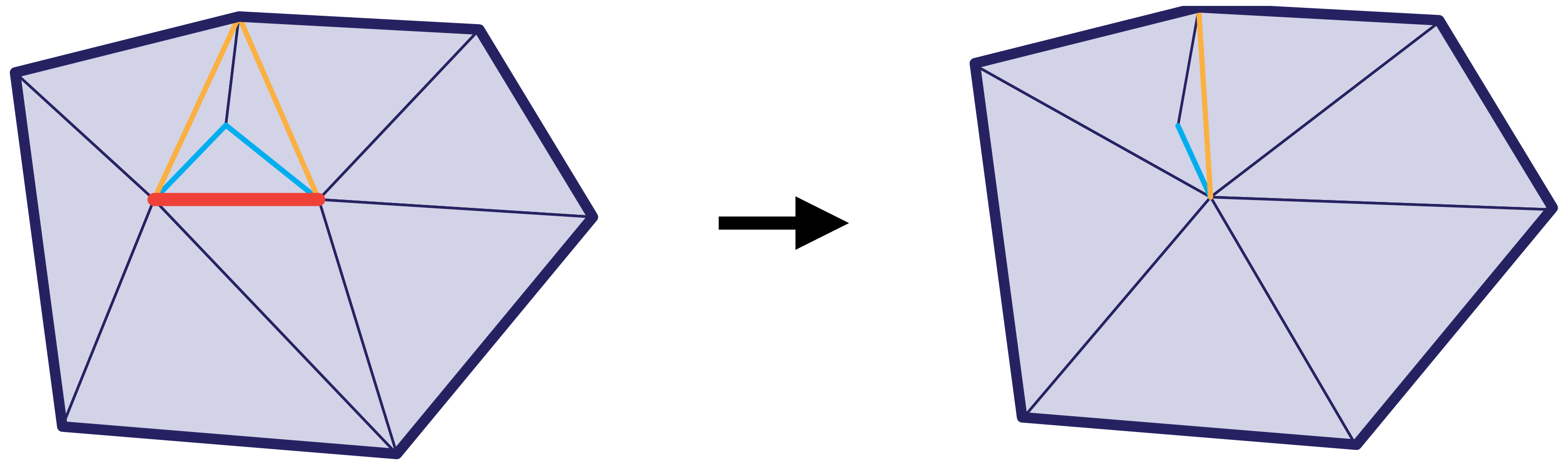
Initial embedding, with invalid elements

No invalid elements are left



An edge collapse operation is illegal if the edge violates link condition

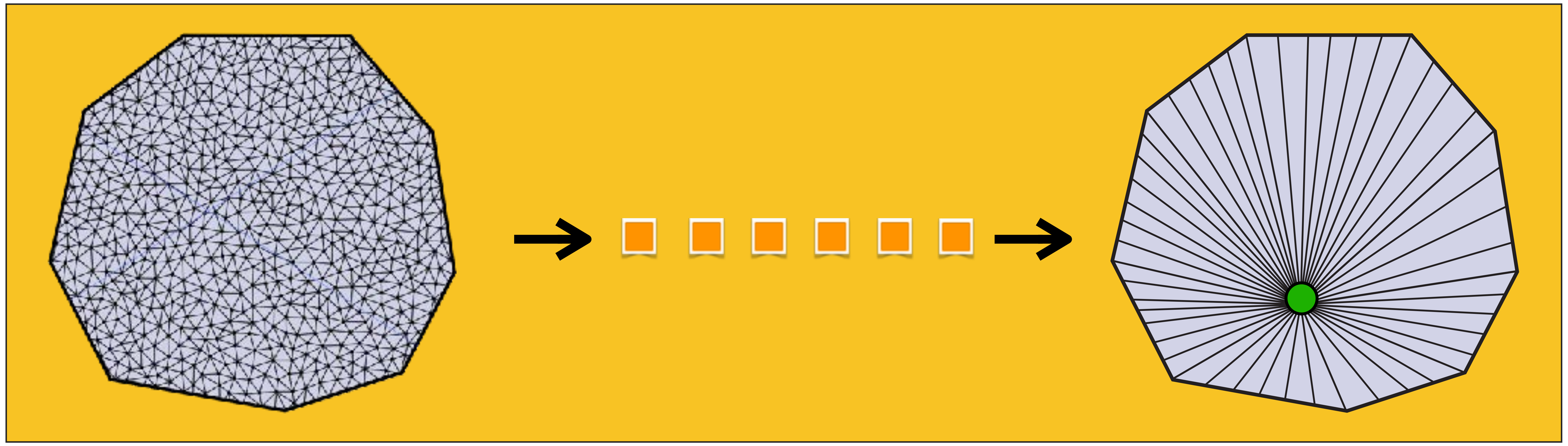
Non-manifold!



* Iteratively collapse edges
* Stop when no invalid elements remain

Existence of Collapsing Sequence

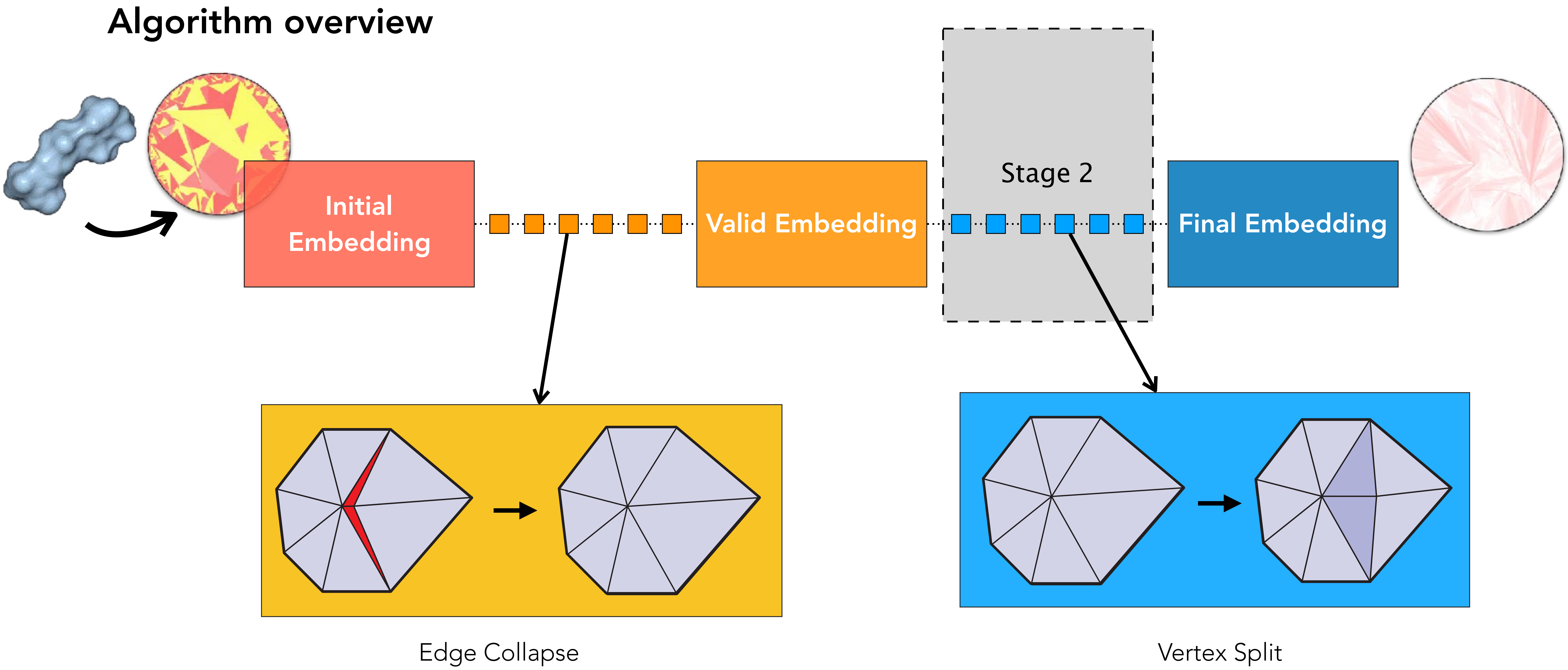
Theorem: We can always find an edge to collapse until only one interior vertex left.



0 flips!

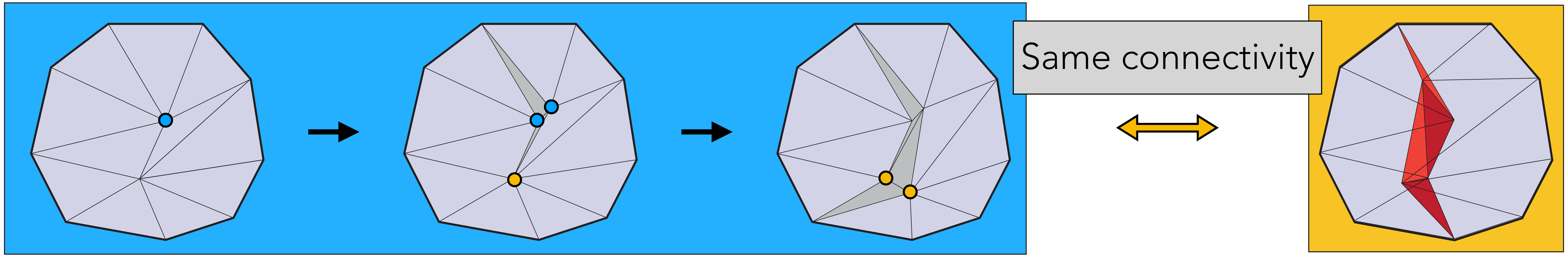
(For a planar and 3-connected mesh)

Progressive Embedding



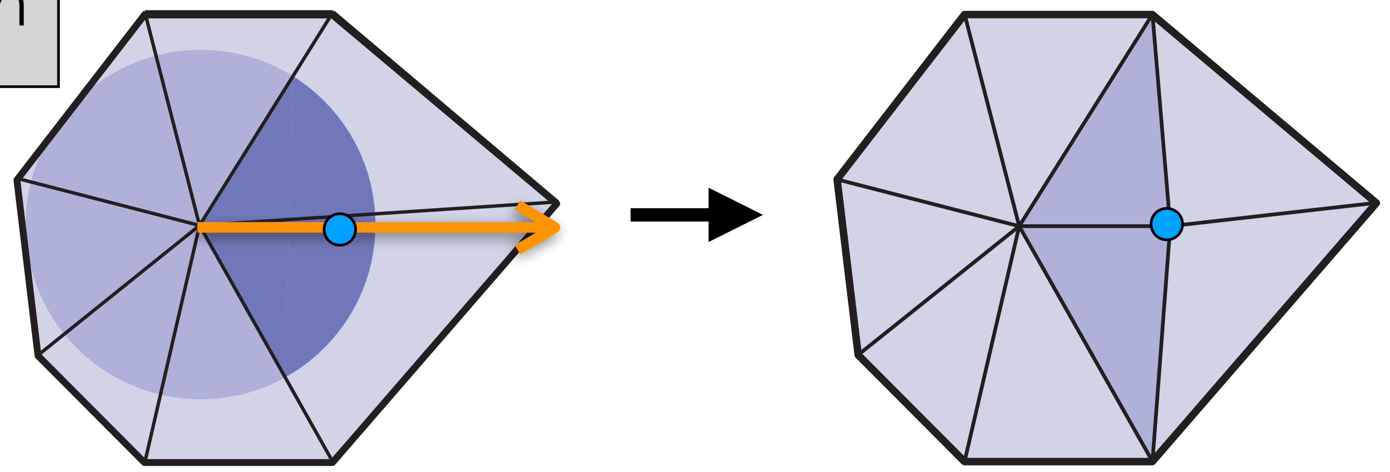
Stage 2: Insertion

Result of stage 1



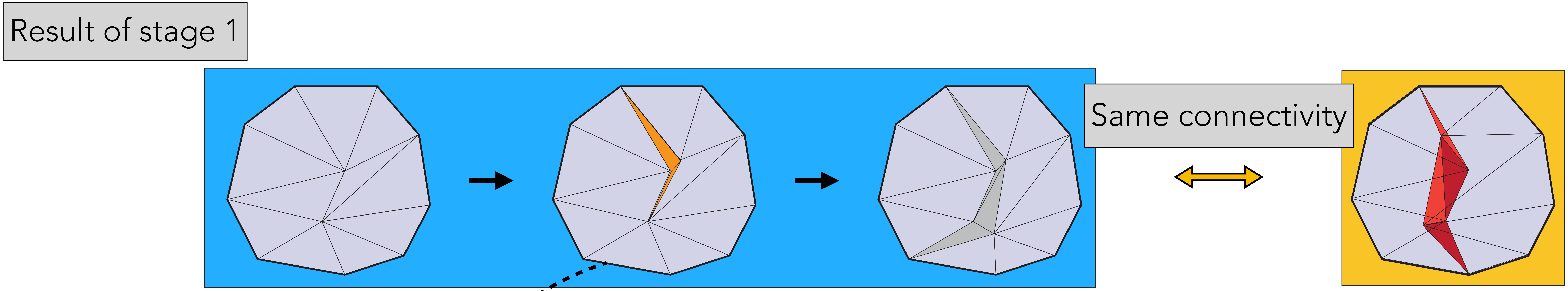
Theorem: A vertex split reversing any collapse can always be done.

Standard Line Search

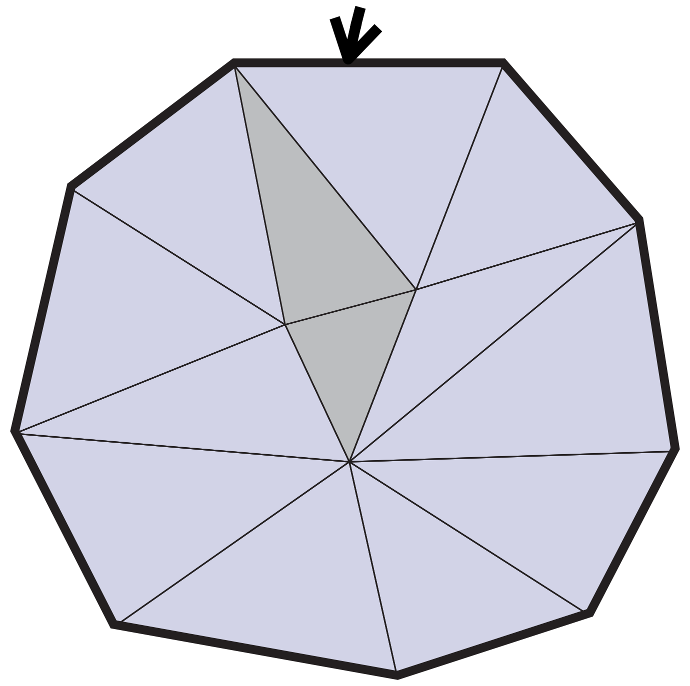


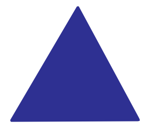
For implementation simplicity: we consider the inscribed cycle of the 1-ring neighbor of vertex to be split as valid region.

Stage 2: Insertion



Local Smoothing: improve quality after each insertion



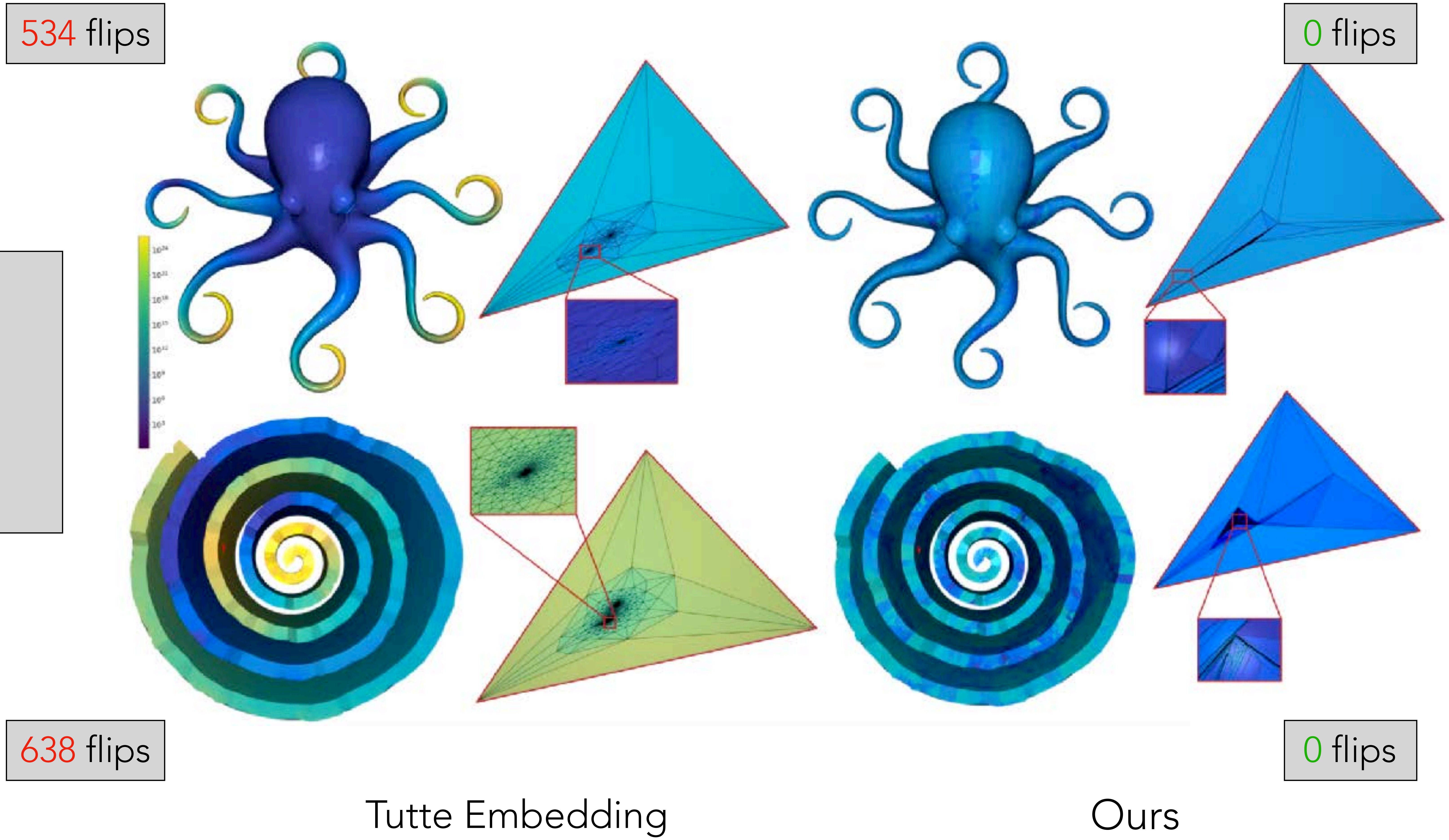


Reference Shape: Equilateral triangle with average area of the whole mesh in the parametric domain

- Symmetric Dirichlet Energy [Smith & Schaefer 2015]
- One vertex at a time using Newton iterations [Fu et al. 2015; Hormann & Greiner 2000; Labsik et al. 2000]
- Updated in parallel independently.

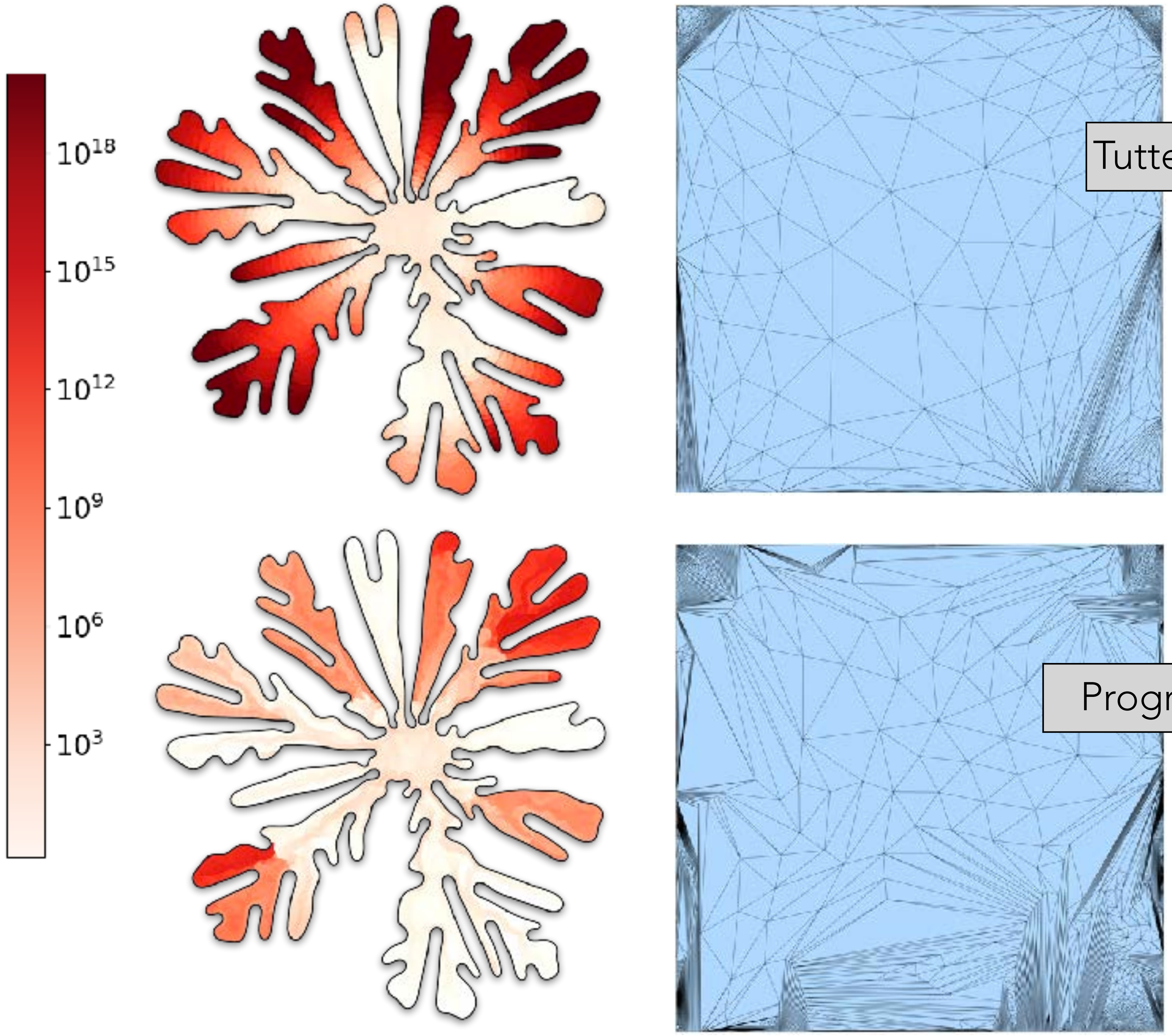
Robustness Test

0 / 2718 model have flipped elements



Results

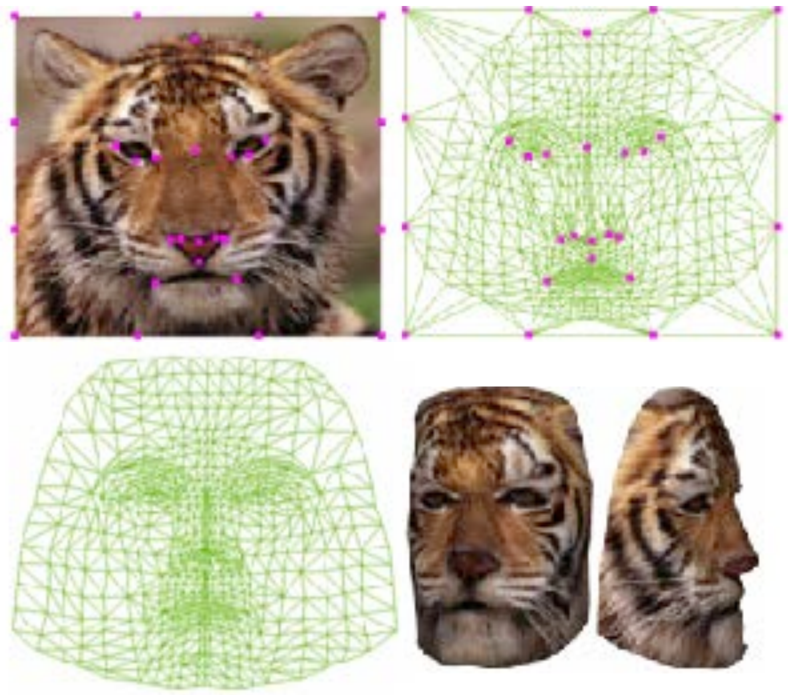
Mapping triangulated Hele-Shaw polygon [Segall et al. 2016] to the interior of a square



Tutte Embedding: 46 flips!

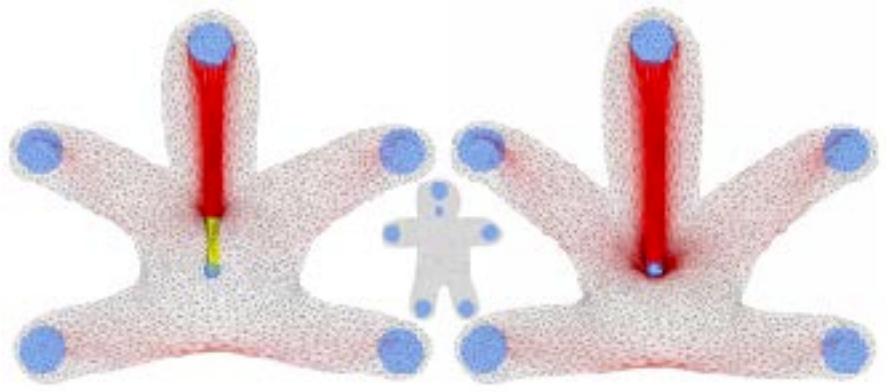
Progressive Embedding: 0 flips

Distortion-Minimizing Mappings



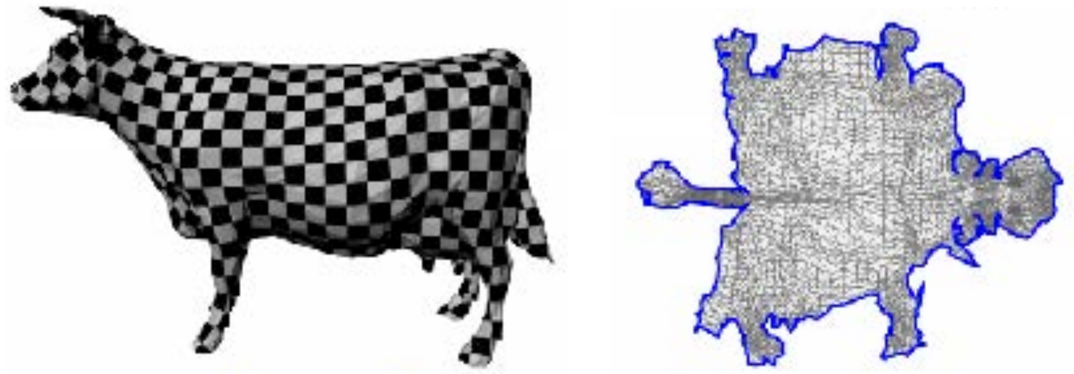
[Kraevoy et al. 2003]

[Rabinovich et al. 2016]

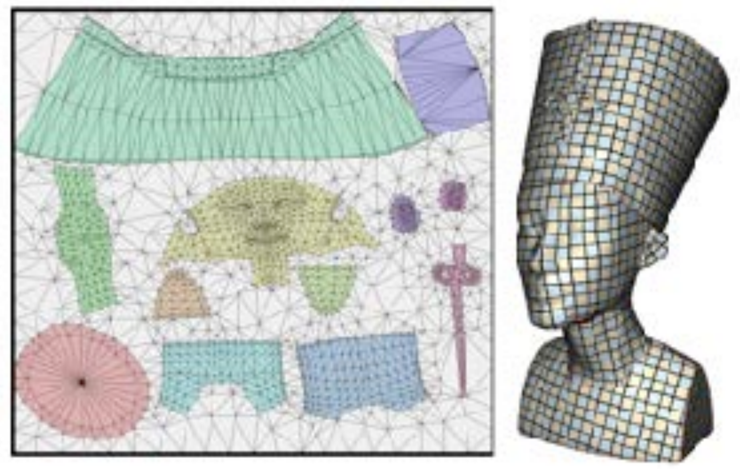


[Schüller et al. 2013]

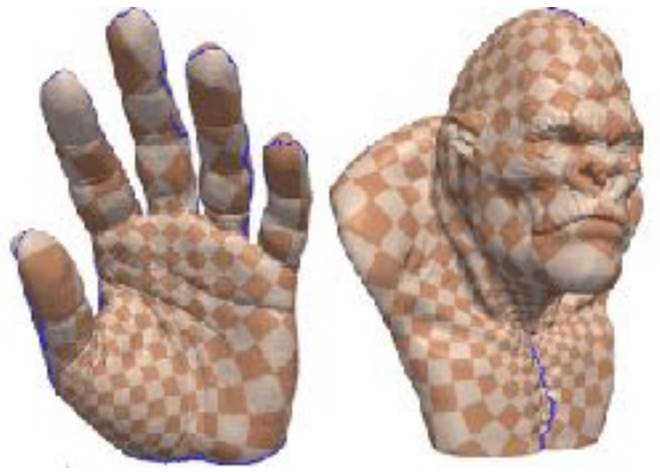
[Li et al. 2018]



[Smith and Schaefer 2015]

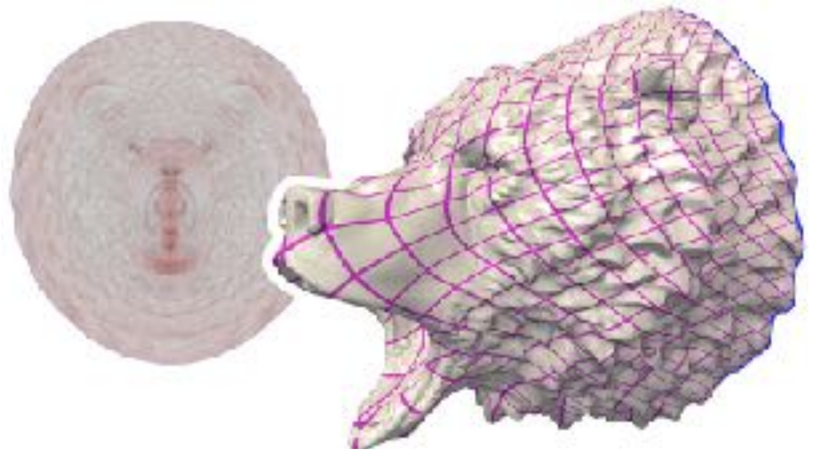


[Jiang et al. 2017]

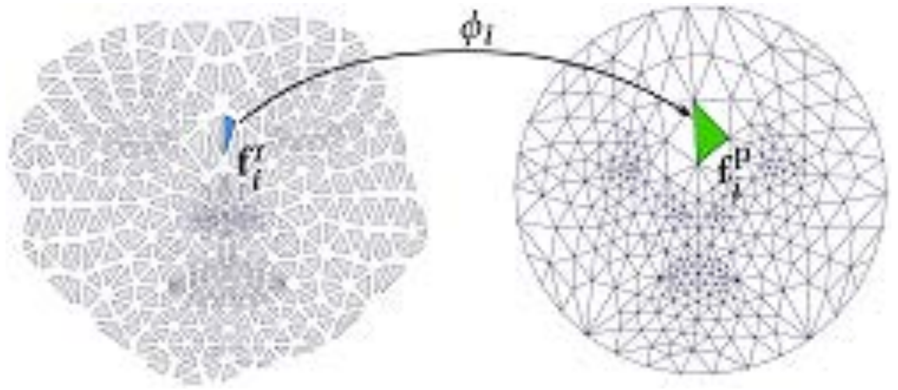


[Kovalsky et al. 2016]

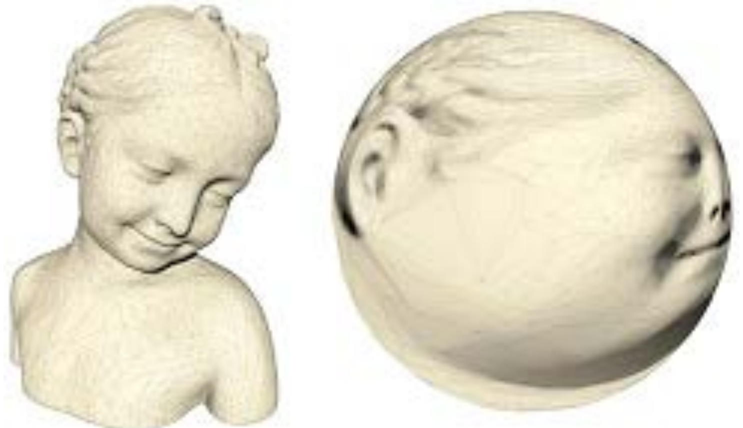
[Claici et al. 2017]



[Shtengel et al. 2017]



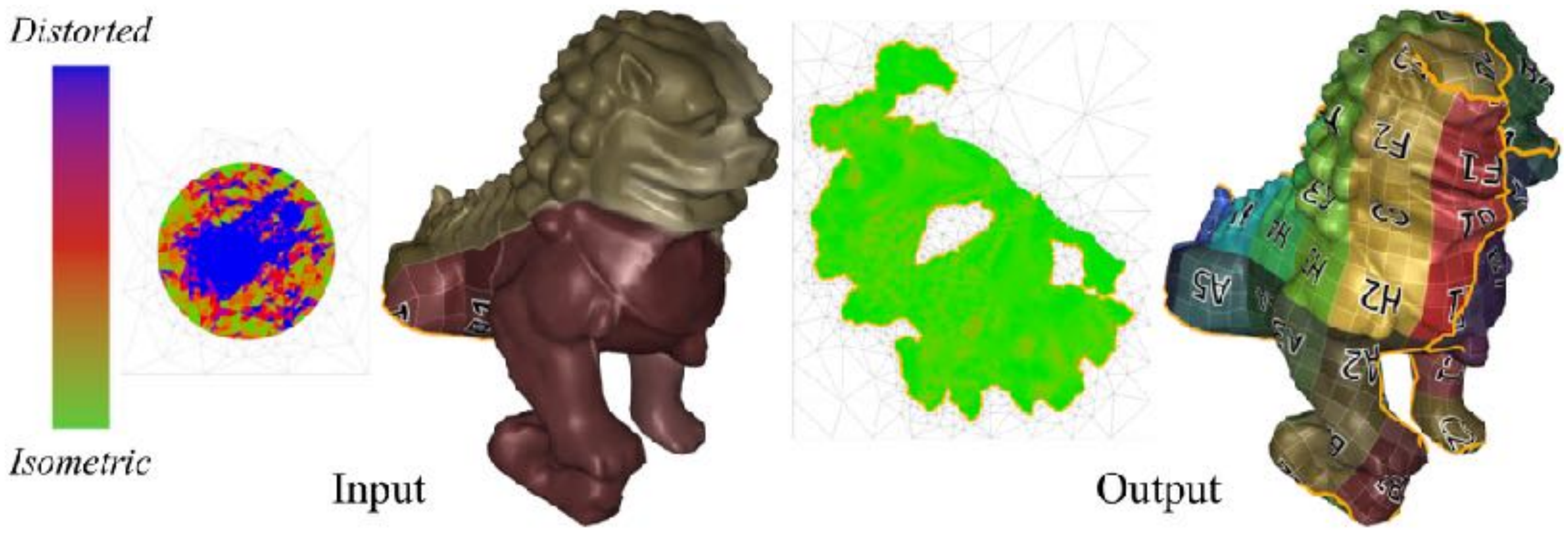
[Liu et al. 2018]



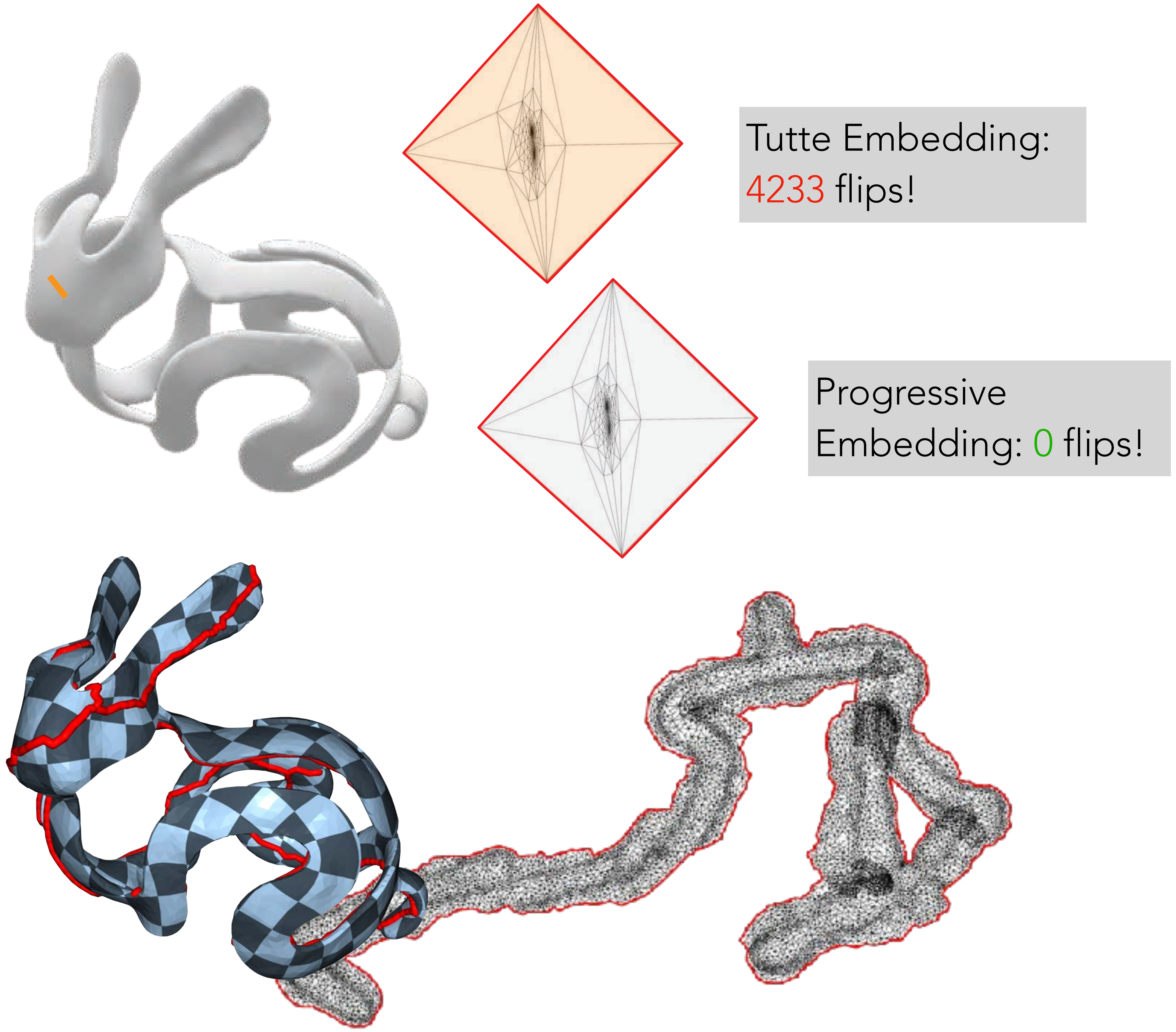
[Wang et al. 2016]

[Fu et al. 2015]

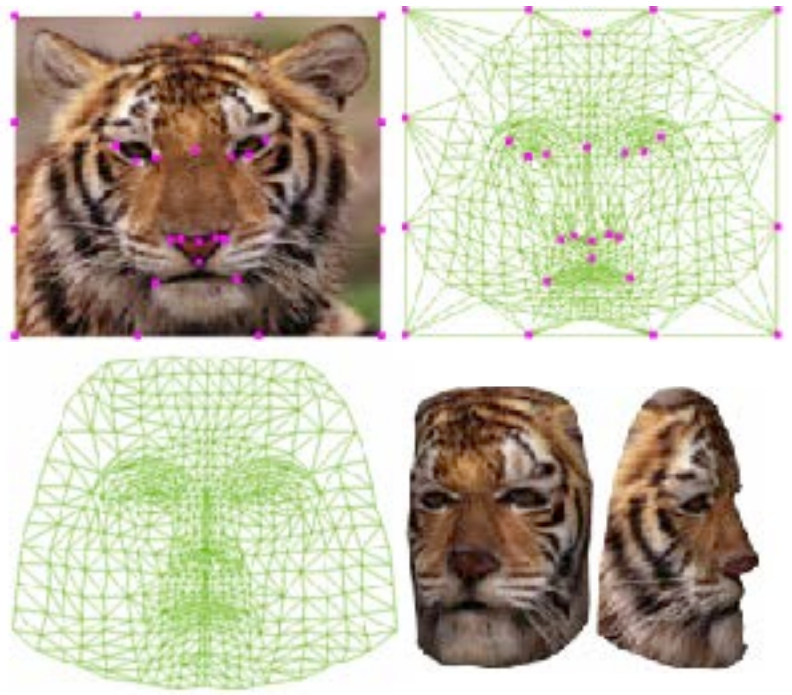
OptCuts

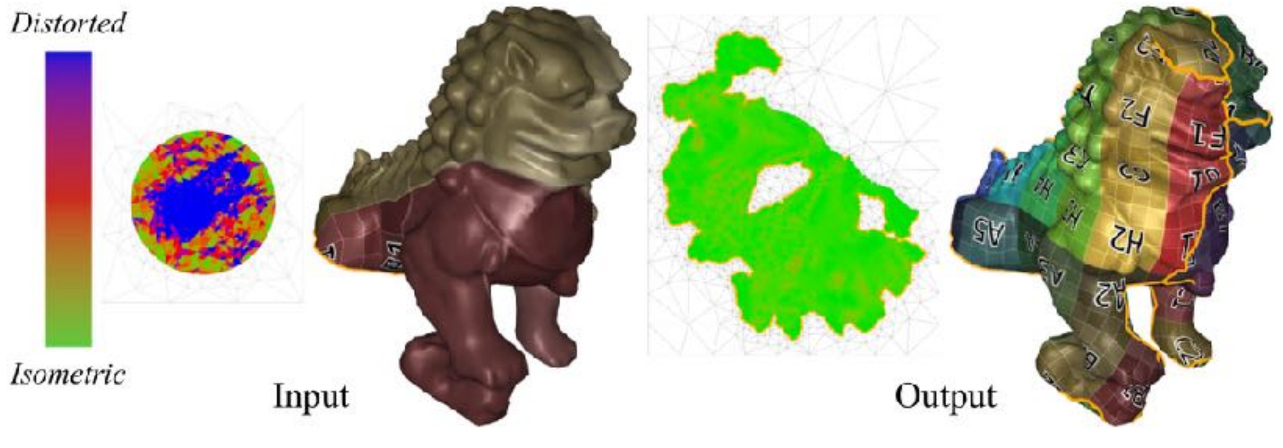


OptCuts [Li et al. 2018] Joint Optimization of Surface Cuts and Parameterization

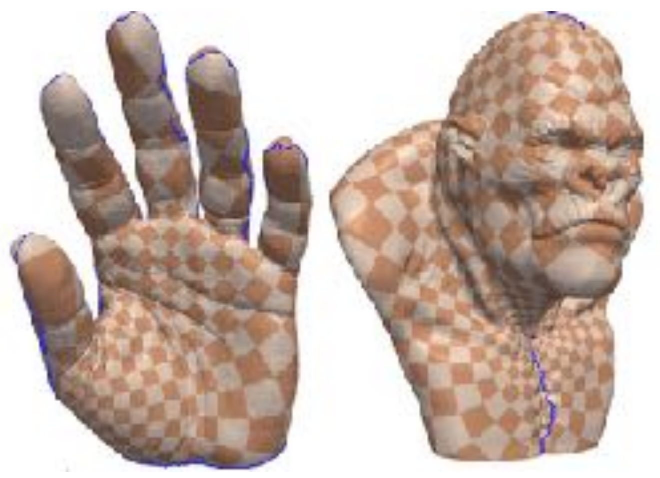


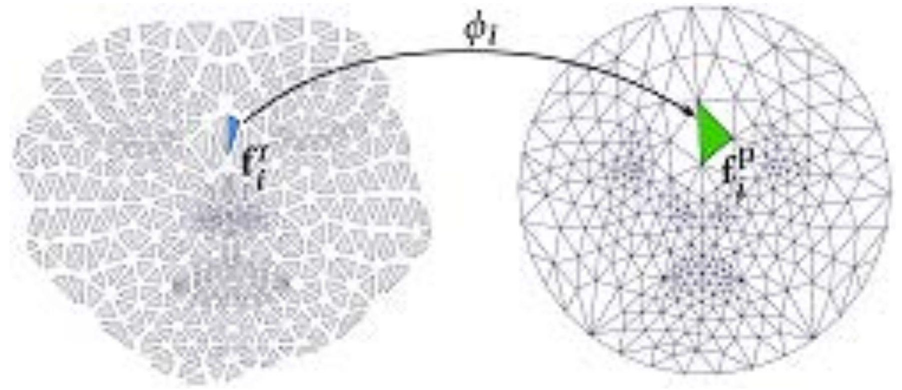
Distortion-Minimizing Mappings


[Kraevoy et al. 2003]

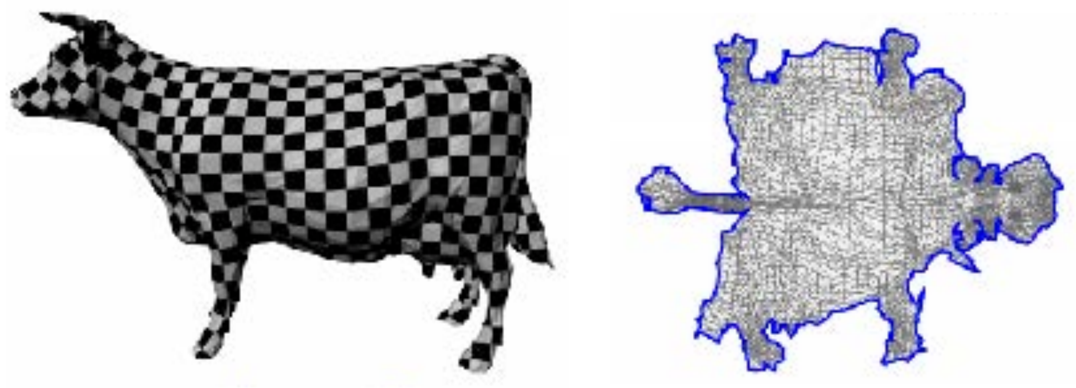


[Li et al. 2018]

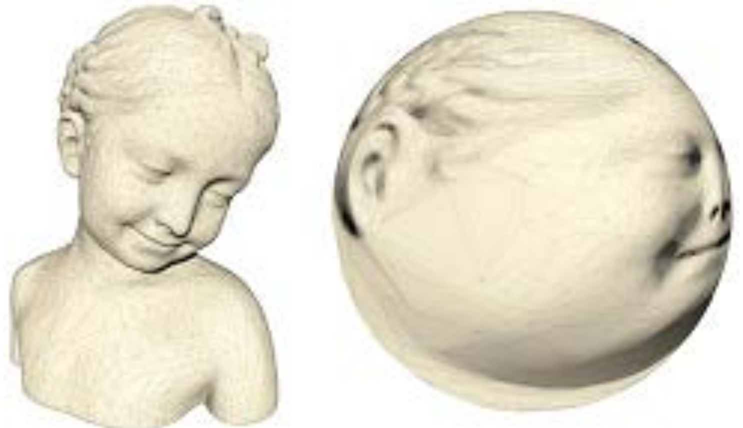

[Kovalsky et al. 2016]

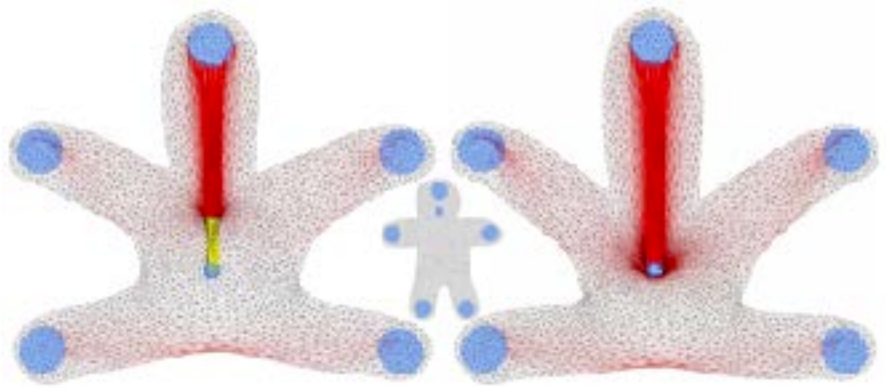

[Liu et al. 2018]

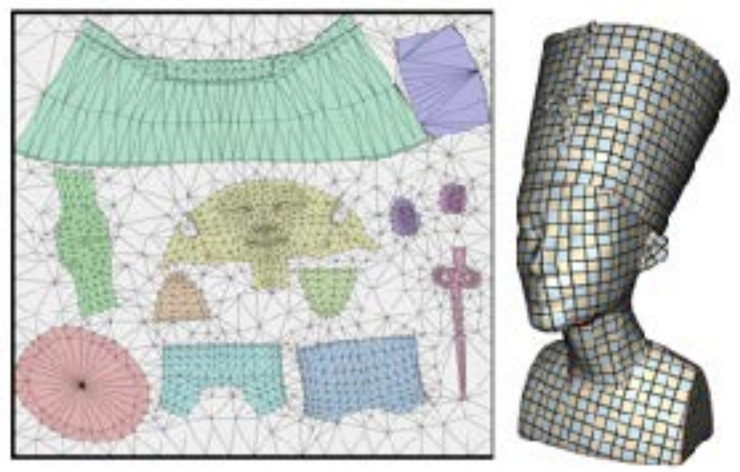

[Rabinovich et al. 2016]

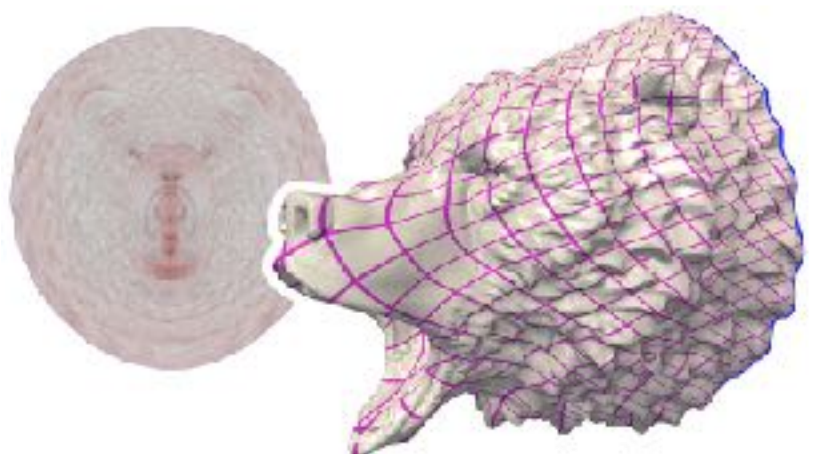

[Smith and Schaefer 2015]


[Claici et al. 2017]


[Wang et al. 2016]


[Schüller et al. 2013]

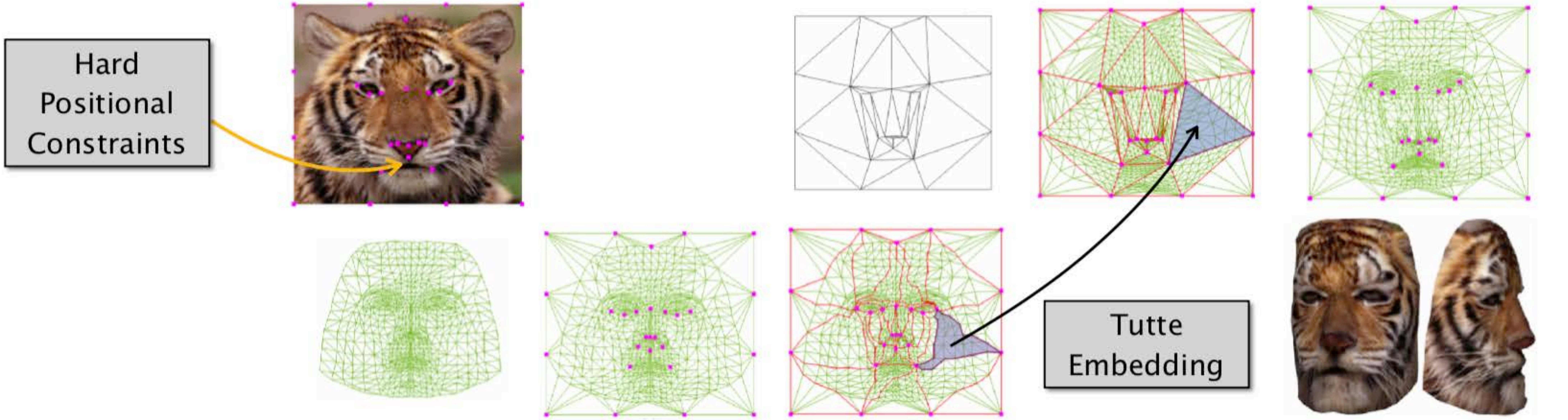

[Jiang et al. 2017]


[Shtengel et al. 2017]


[Fu et al. 2015]

Matchmaker

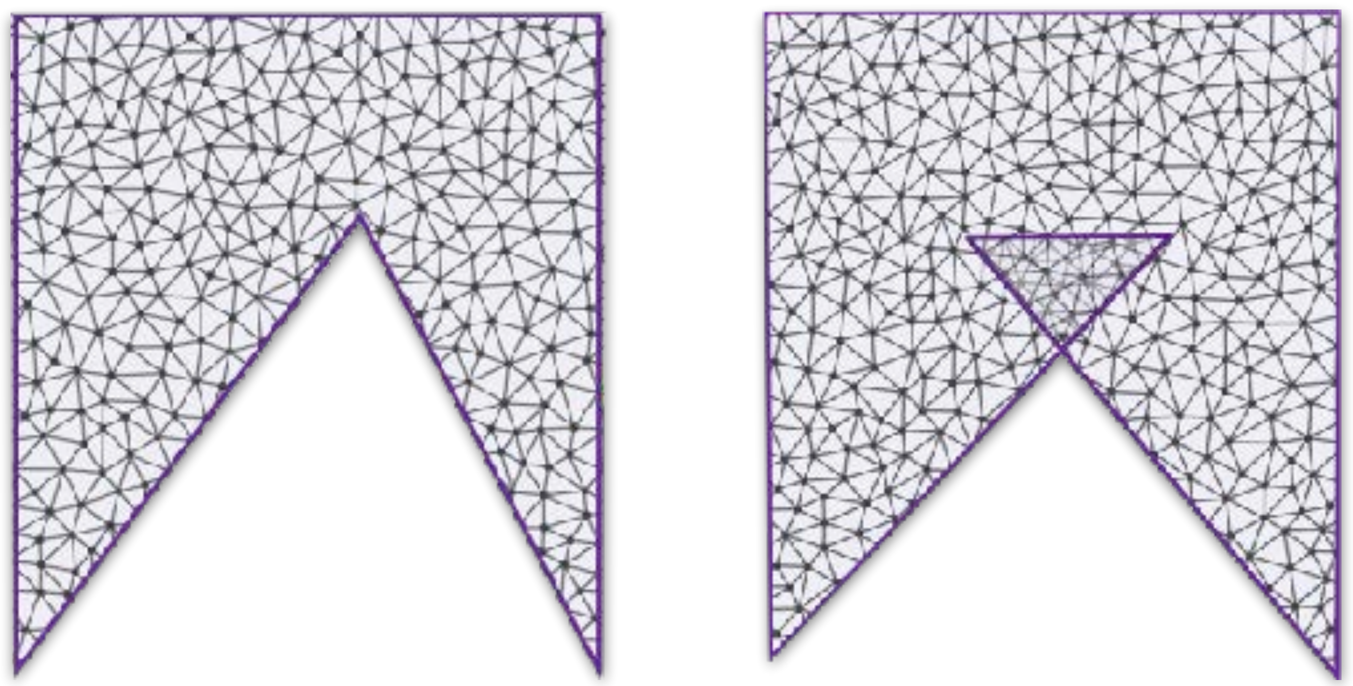
[Kraevoy et al. 2003]



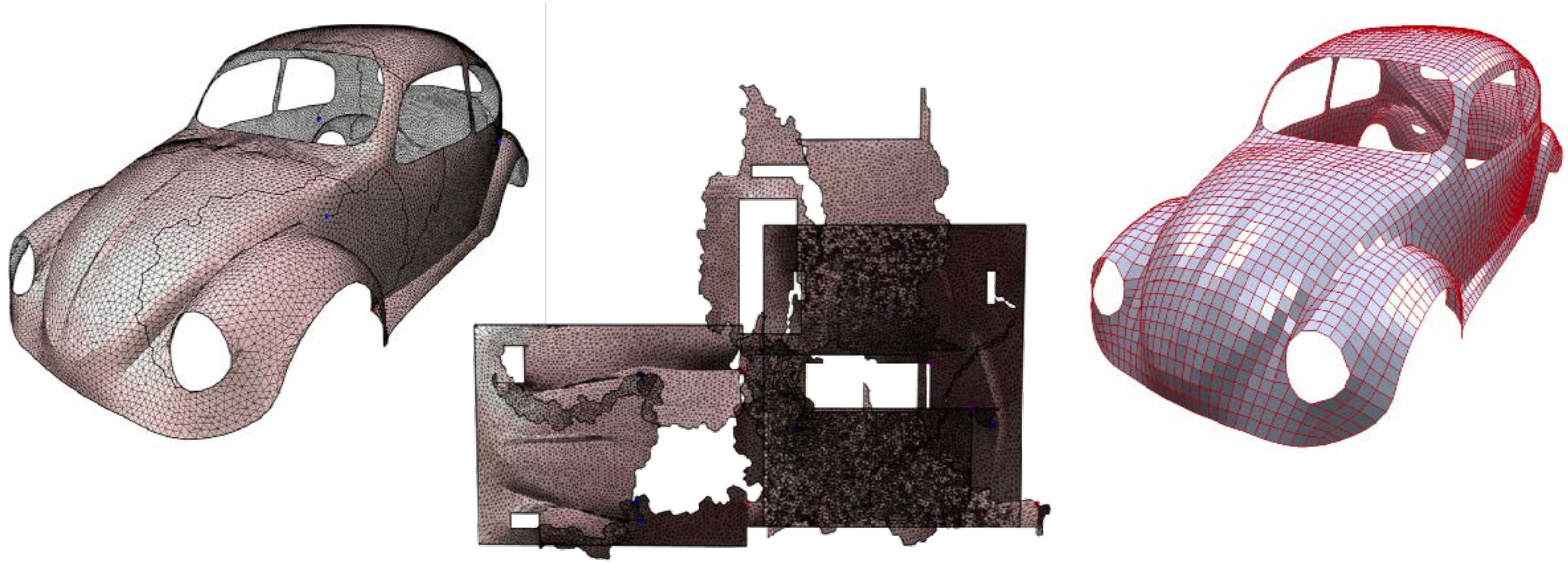
Issues:

* Tutte Embedding as a building block may introduce flips.
* Target domain must not have self-intersection.

Self-overlapping Domains

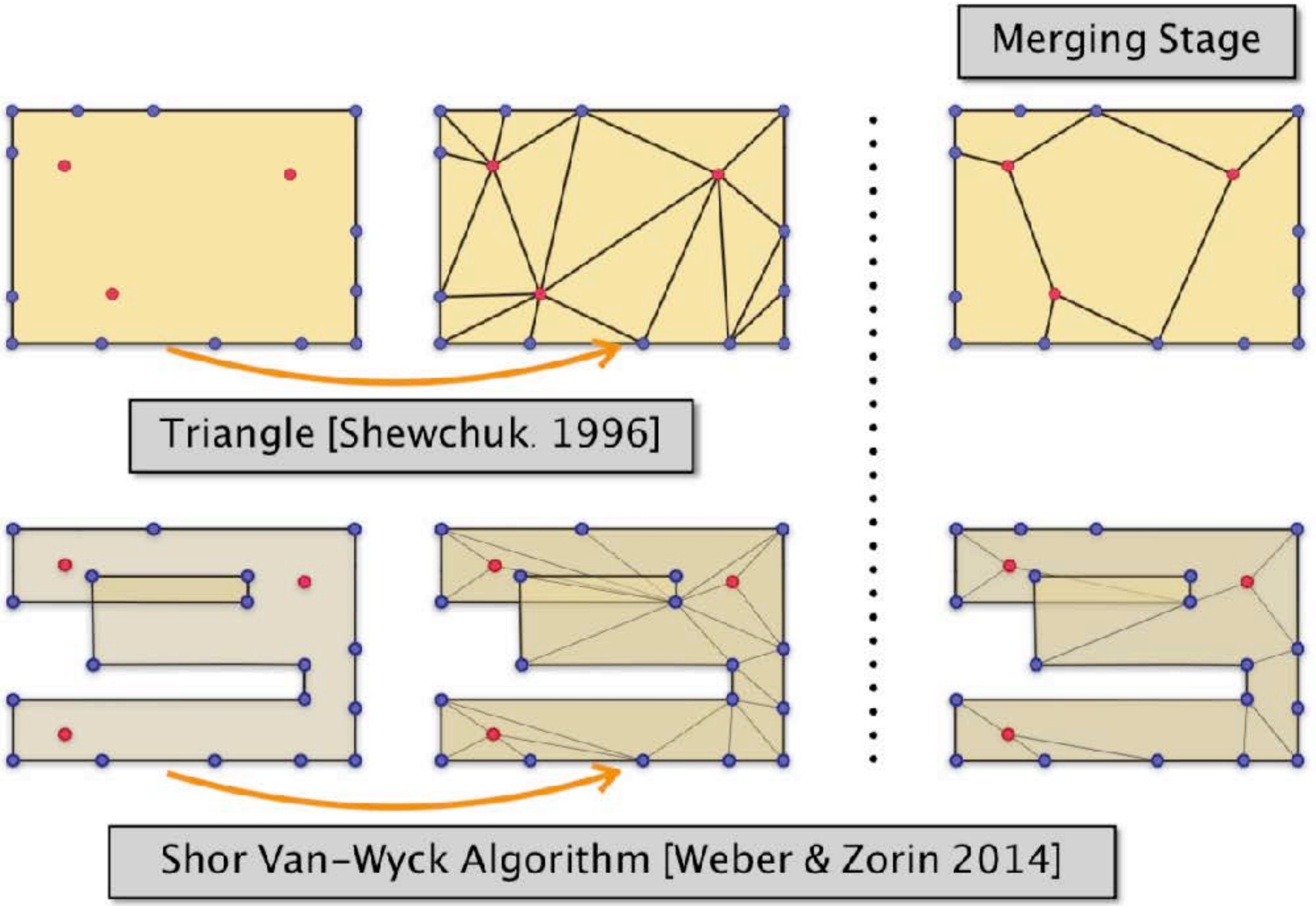


Self-overlapping boundary



[Weber & Zorin 2014, Bommes et al. 2009]

Domain Decomposition



Goal: divide the domain into multiple ~~triangles.~~

convex polygons

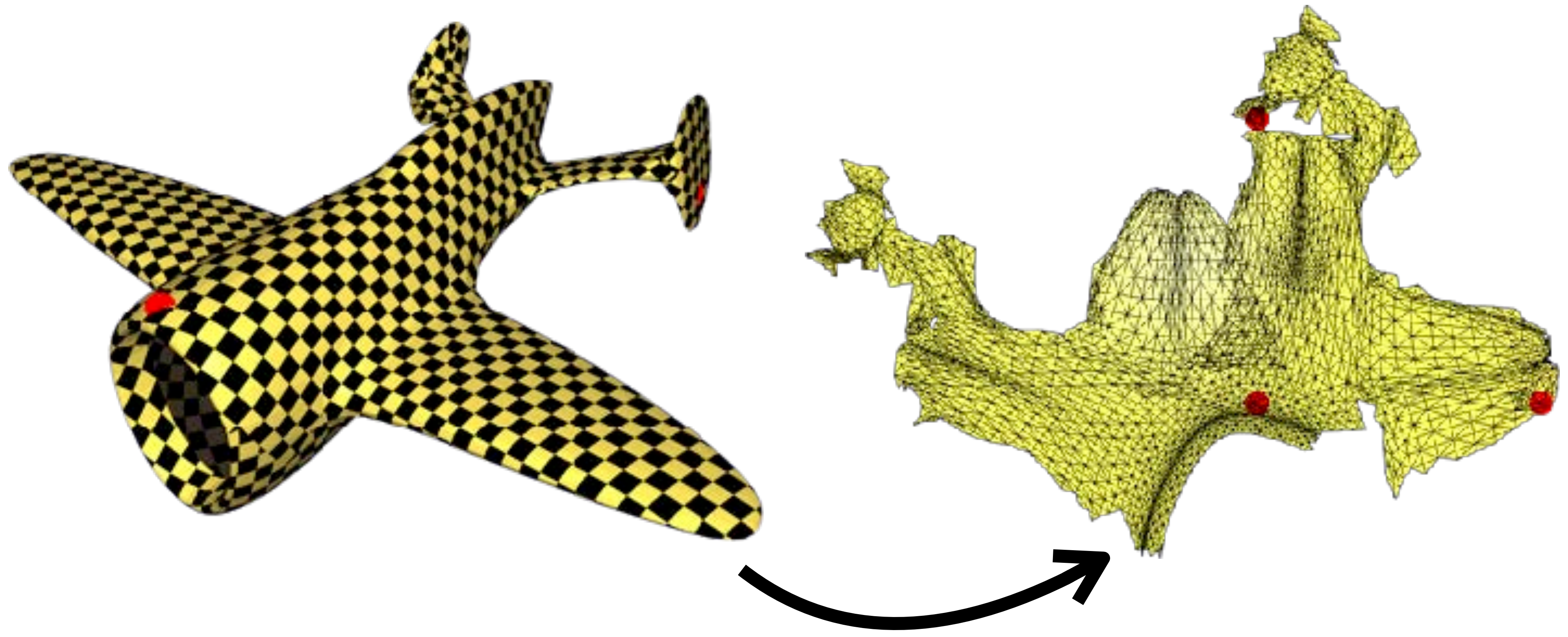
1. Triangulate the domain
2. Embed the hard constraints
3. Greedily merge adjacent triangles

Matchmaker++

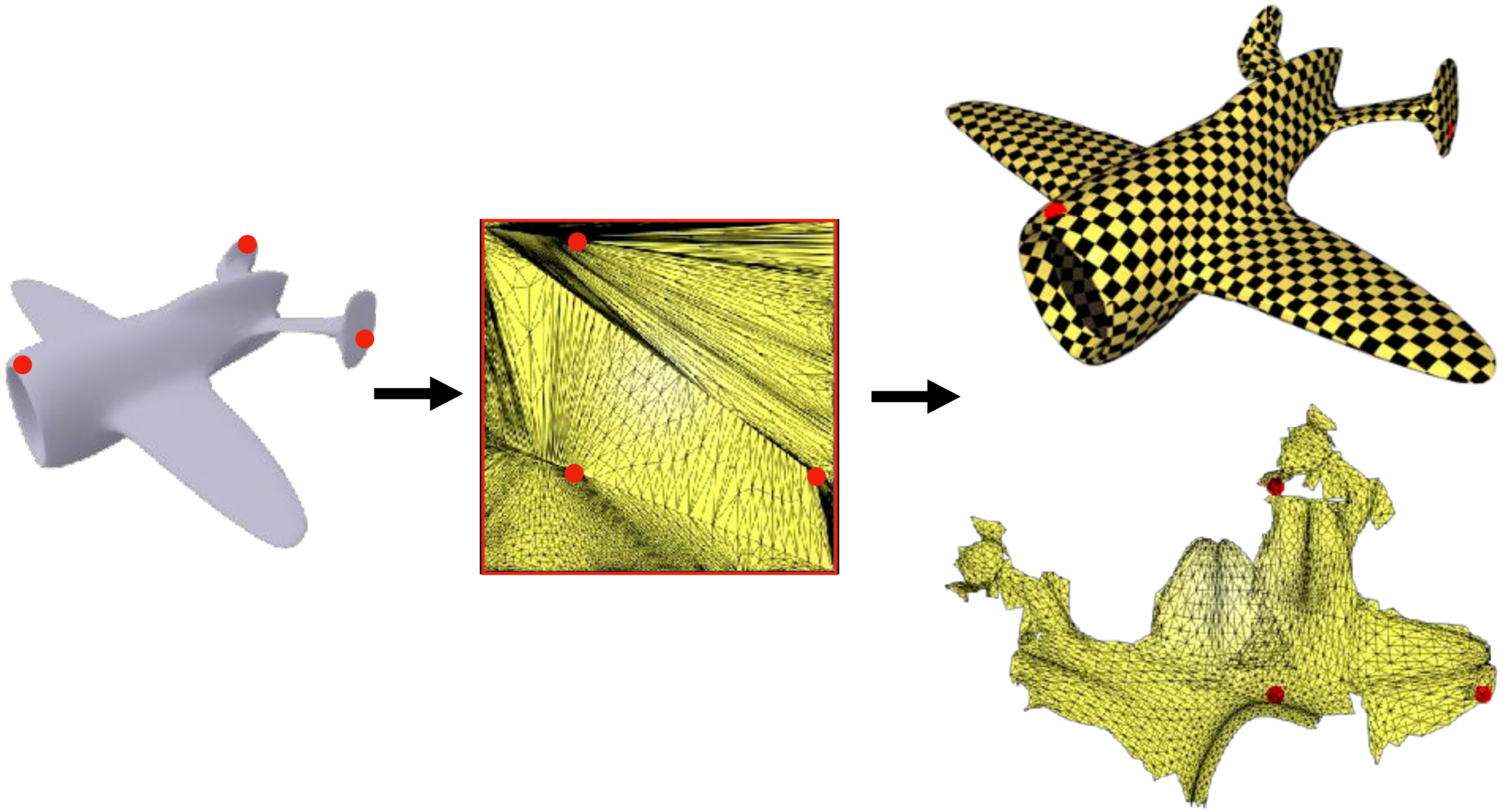
[Weber & Zorin 2014]

MatchMaker + Shor-Van Wyck Triangulation + Progressive Embedding

Robust surface parametrization method for multiply-connected domains with arbitrary point constraints

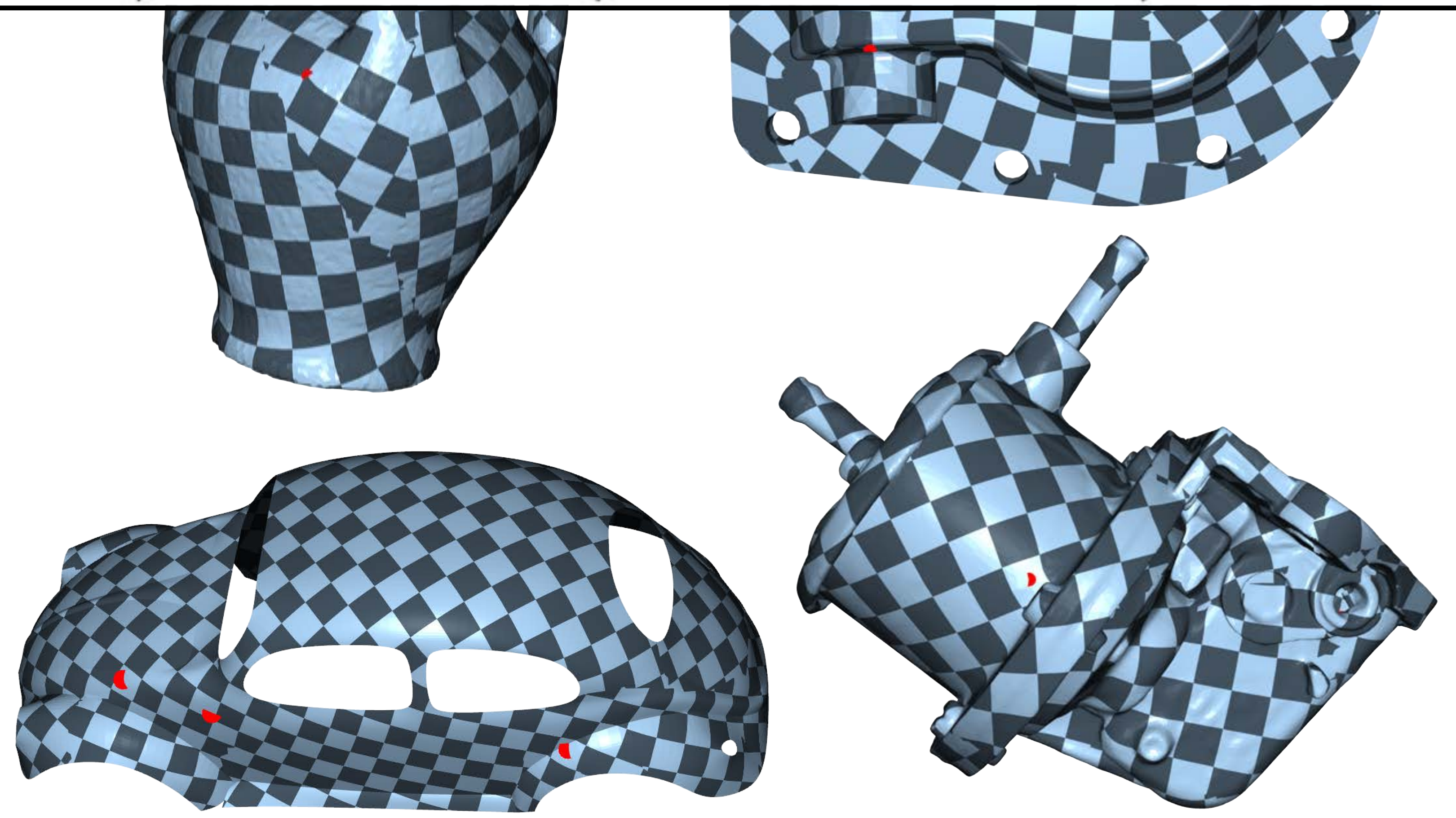


Random Hard Constraints



Dataset: 102 models from [Myles & Zorin 2014]

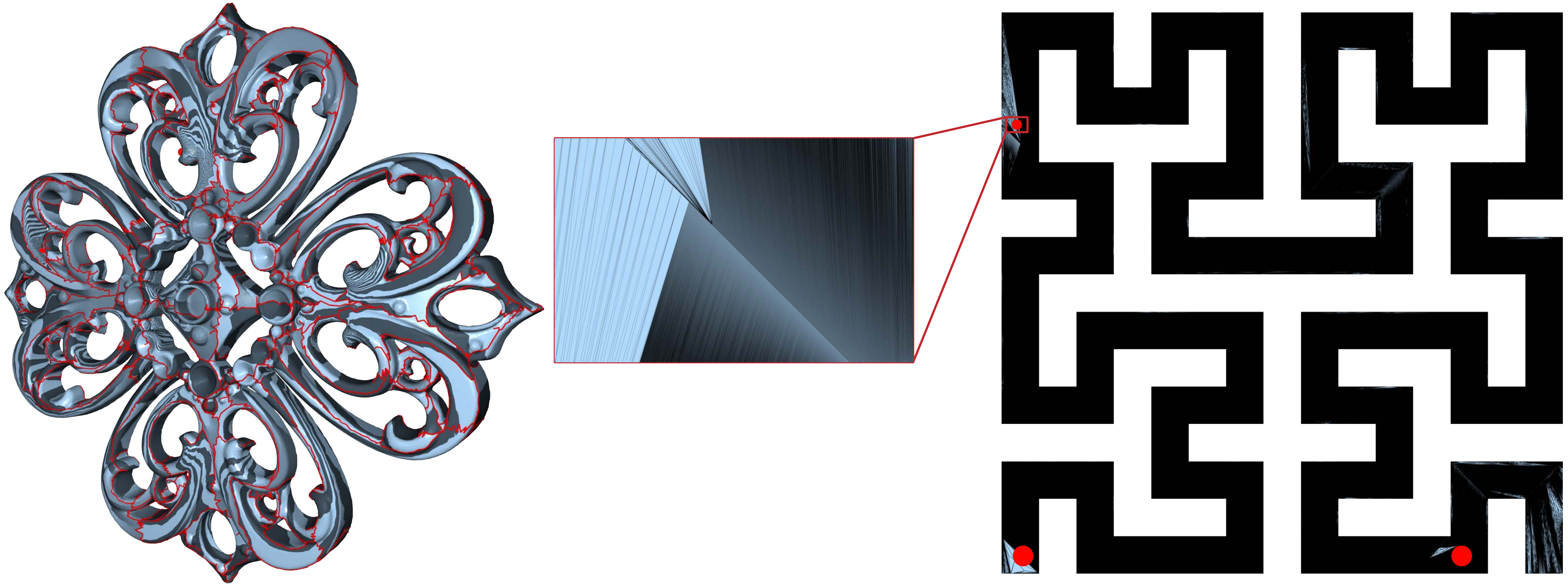
Name	#V	#F	#invalid	#flipped	PE(s)	MM++(s)
Botijo	43786	83788	0	0	NA	3.9
Beetle	20619	39276	0	0	NA	1.1
Casting	21236	39438	67	40	27.4	1.3
Oil pump	54135	103778	5	0	2.3	4.8



(Seams are part of the dataset)

Challenging Stress Test

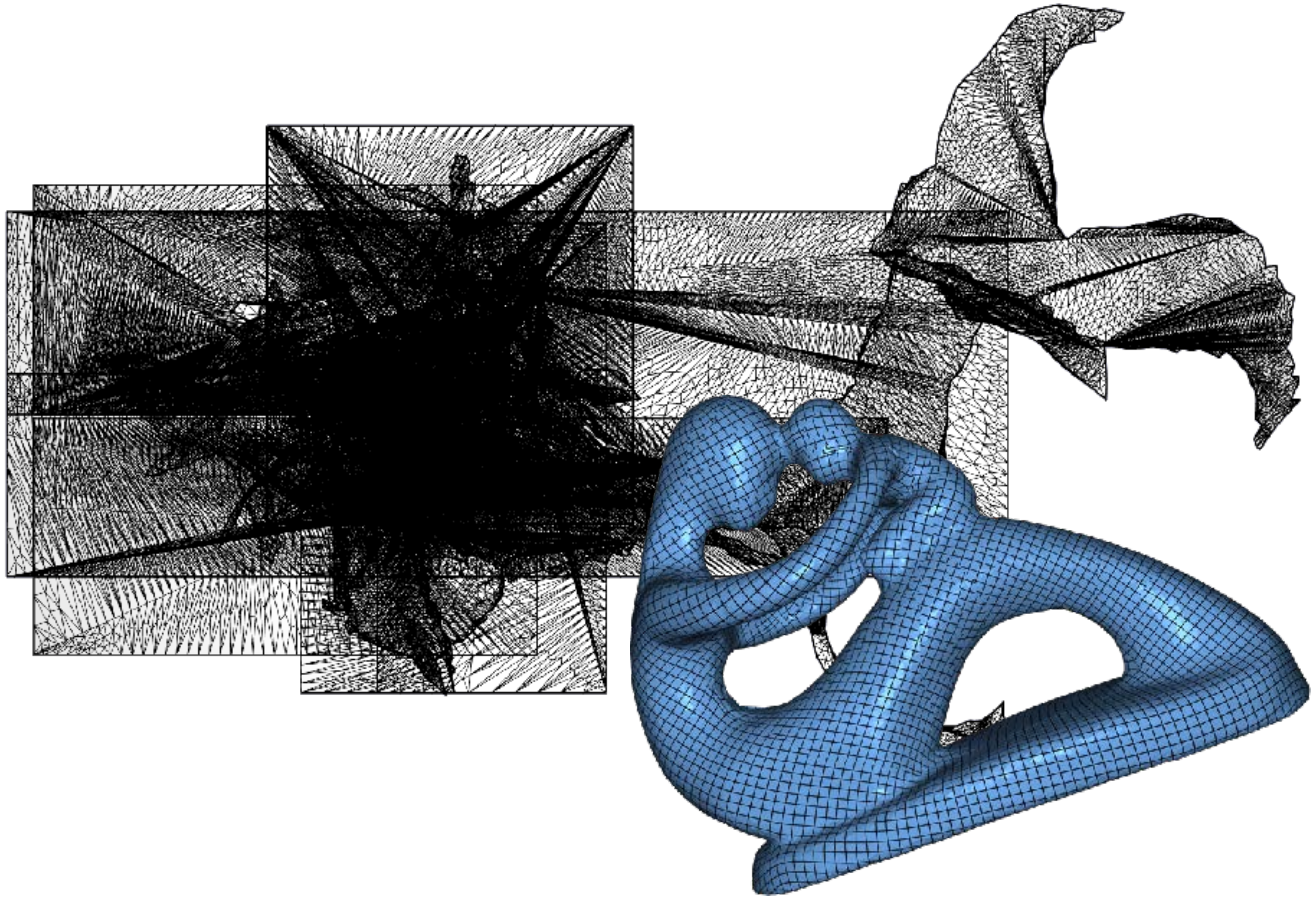
- 102 models from [Myles & Zorin 2014]
- 3 random point constraints

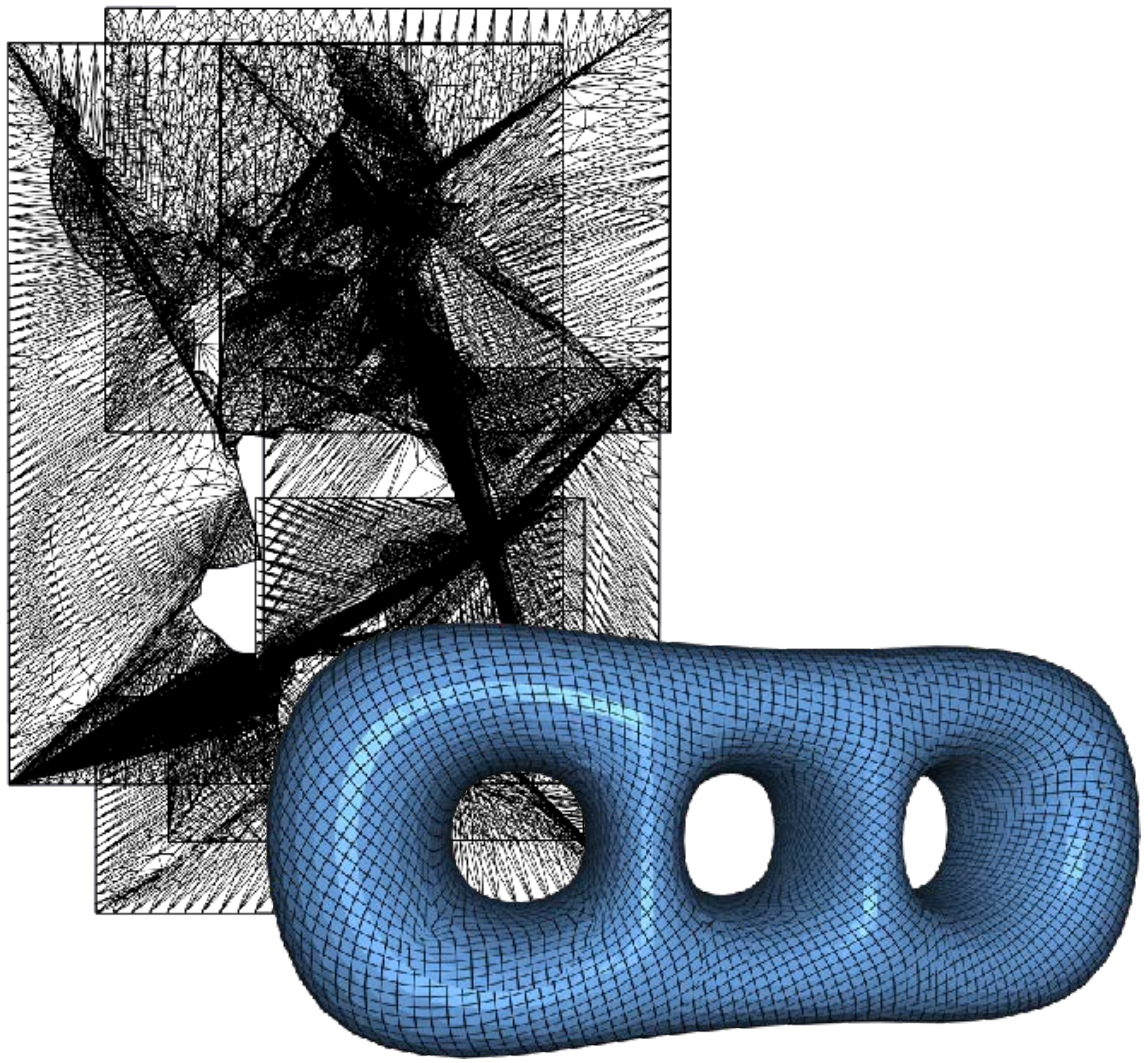


Name	#V	#F	#invalid	#flipped	PE(s)	MM++(s)
Filigree	49872	100000	32	0	72.4	30.8

Self-overlapping Boundary

Name	#V	#F	#invalid	#flipped	PE(s)	MM++(s)
Fertility	16508	33028	0	0	NA	582.4
3 holes	7440	14886	0	0	NA	107.4





Summary

- A robust algorithm to compute planar embeddings, which has the same guarantees as the Tutte Embedding, but works robustly in floating point coordinates.
- Paired with Matchmaker, our algorithm enables the robust generation of constrained locally injective maps with hard constraints.

Thank you!

This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise. This work was partially supported by the NSF CAREER award with number 1652515, the NSF grant IIS-1320635, the NSF grant DMS- 1436591, the NSF grant 1835712, a gift from Adobe Research, and a gift from nTopology.

Reference Implementation

https://github.com/hankstag/progressive_embedding