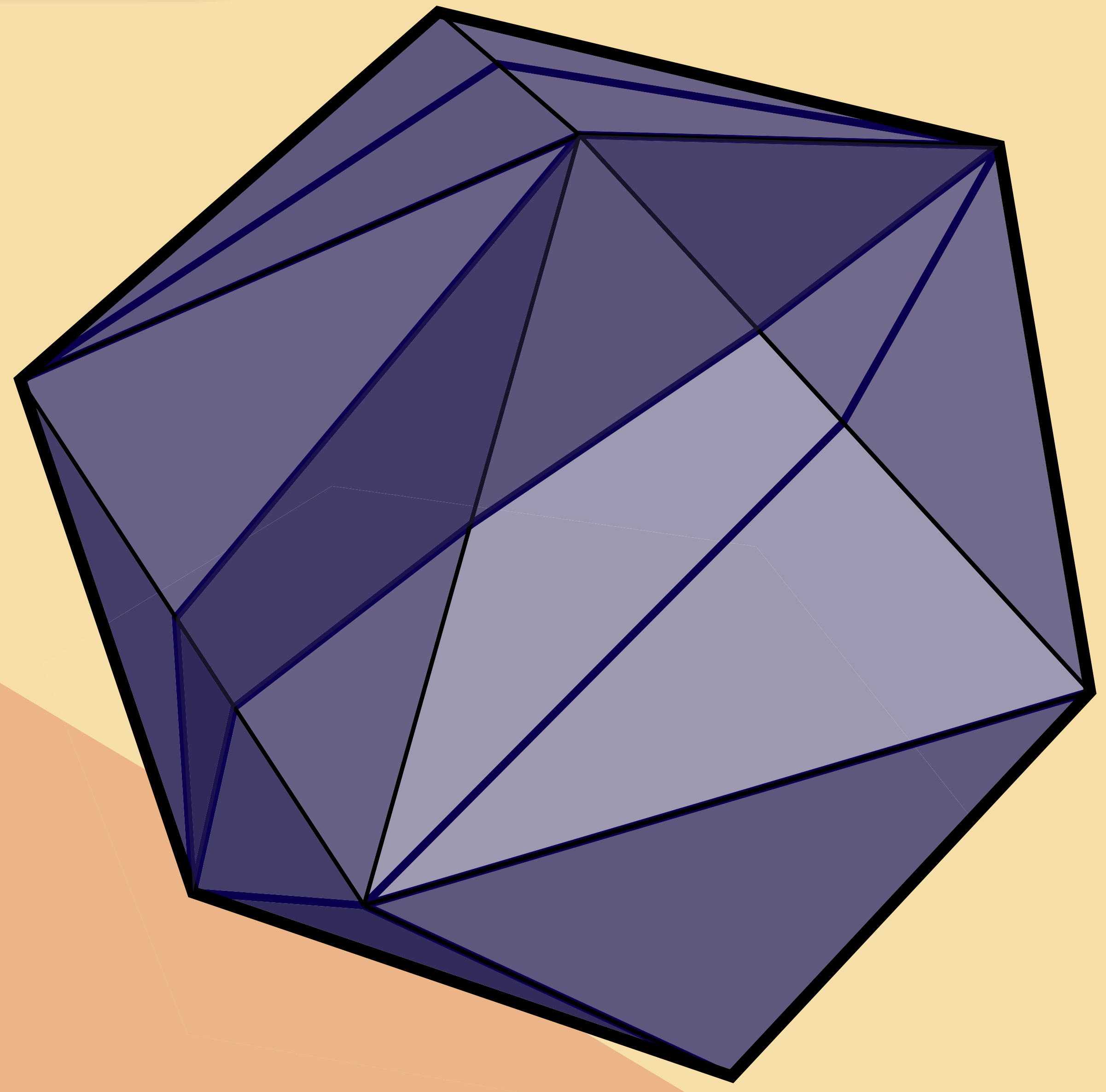
Intrinsic Triangulations in Geometry Processing

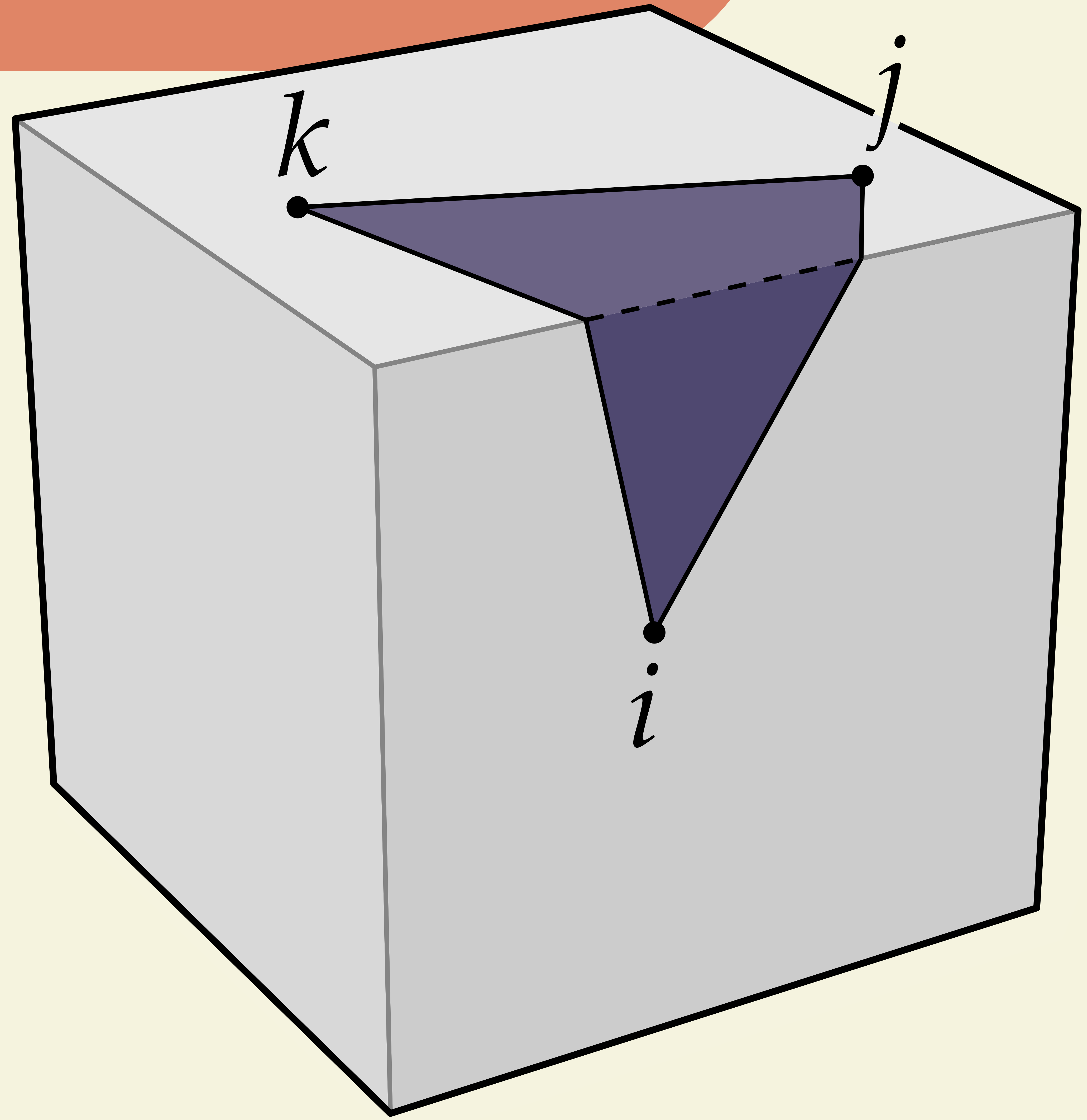
Mark Gillespie, Carnegie Mellon University

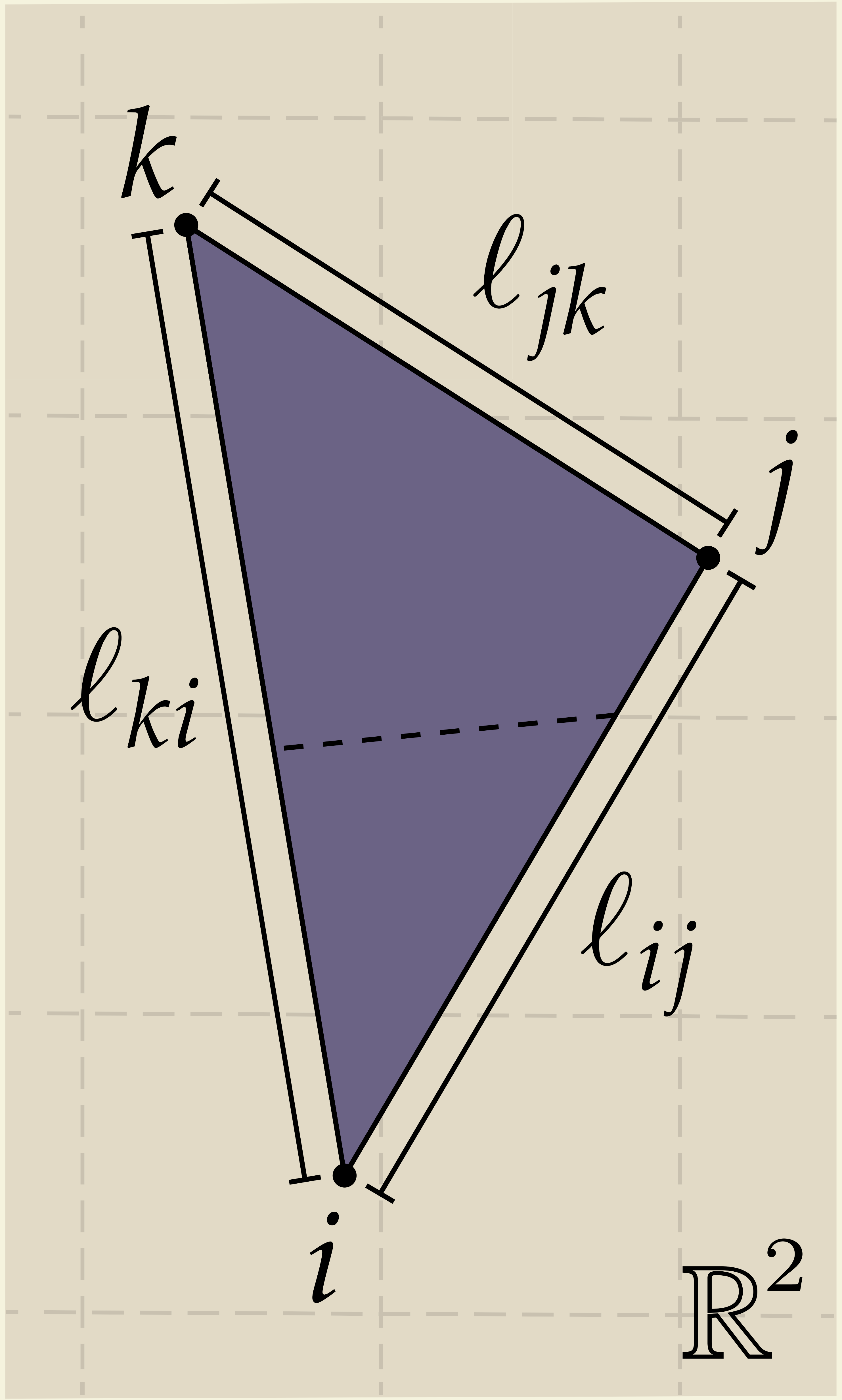


SFB TRR109 Colloquium

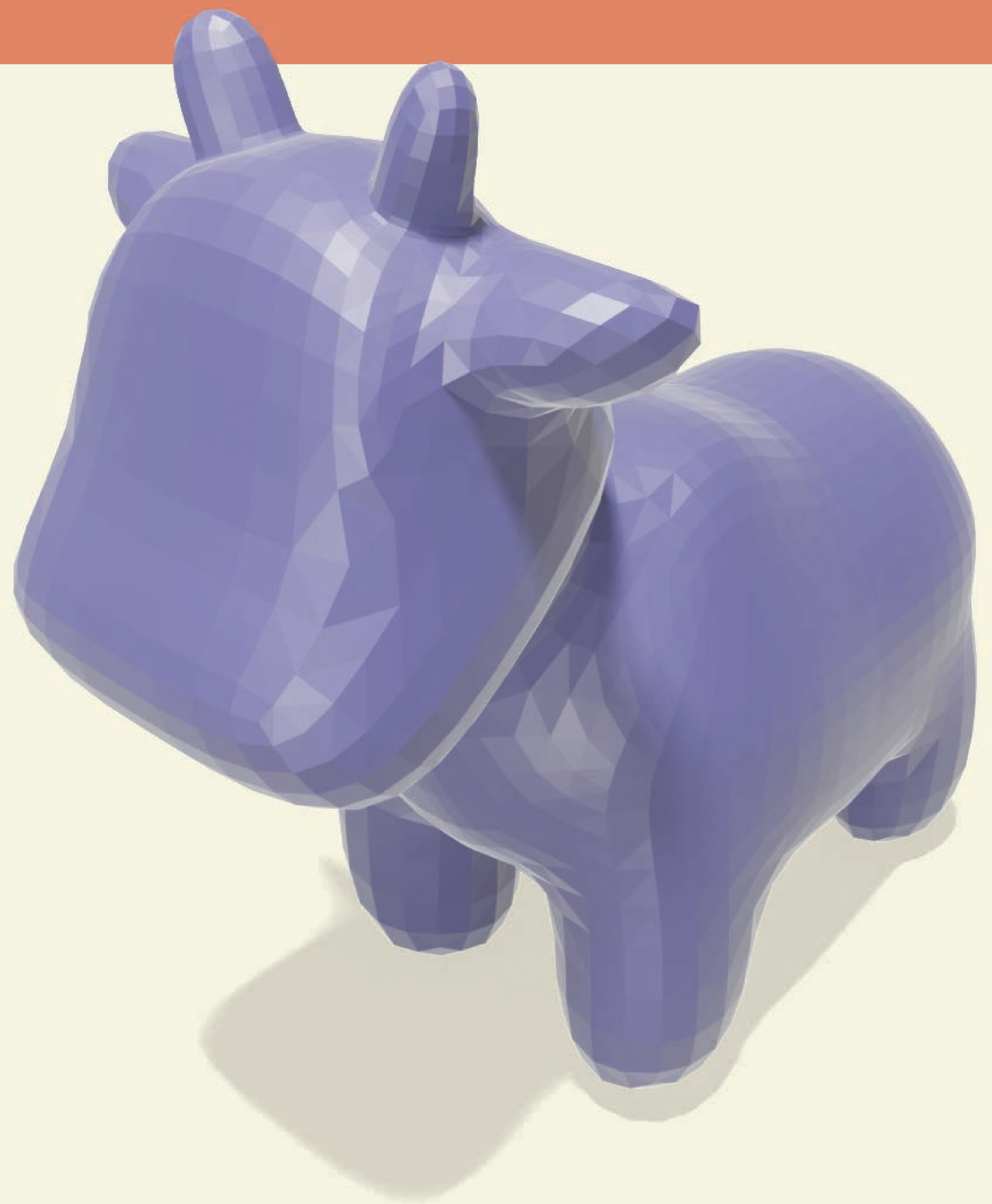
Tuesday 11th July 2023

Intrinsic triangles

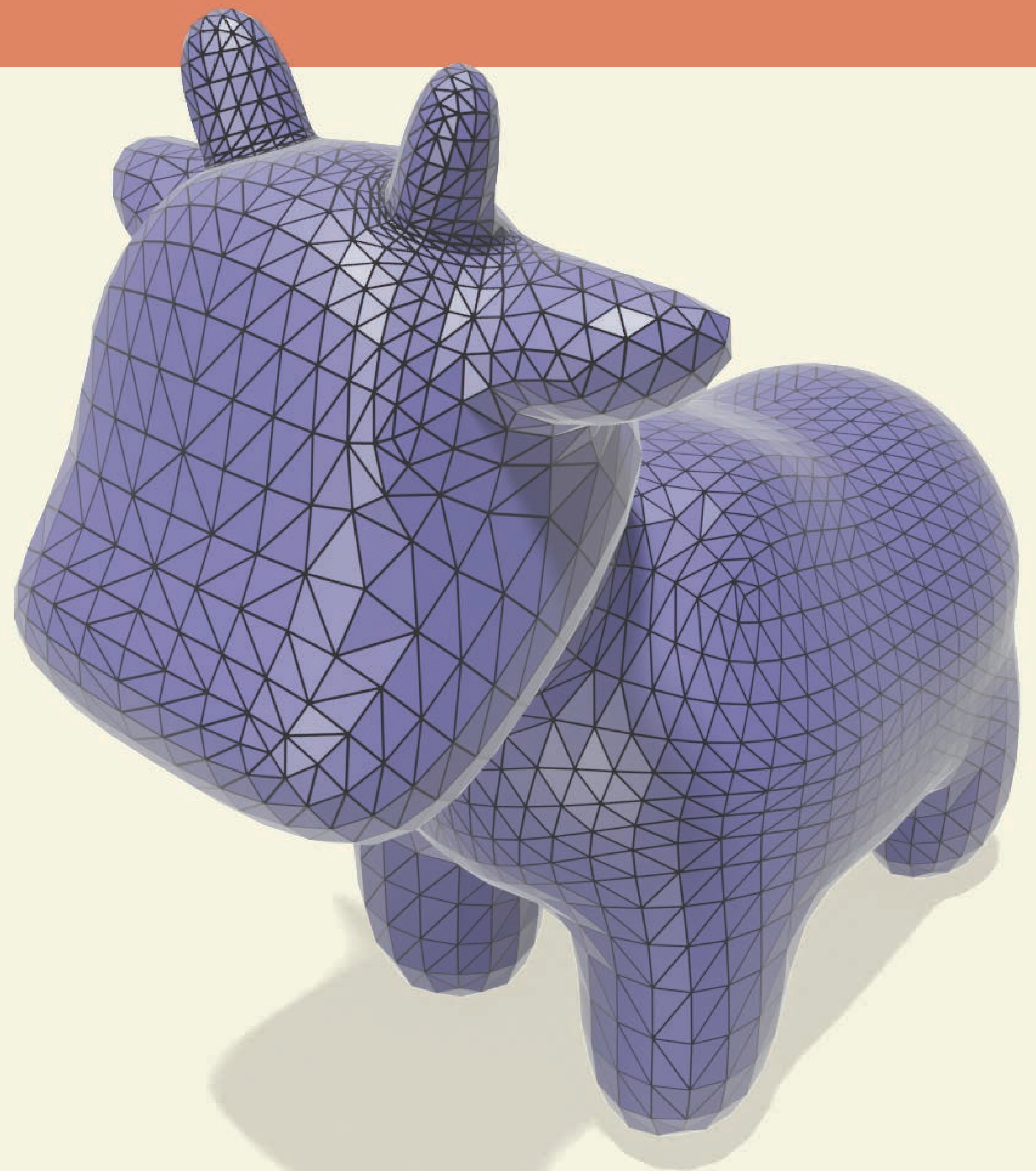


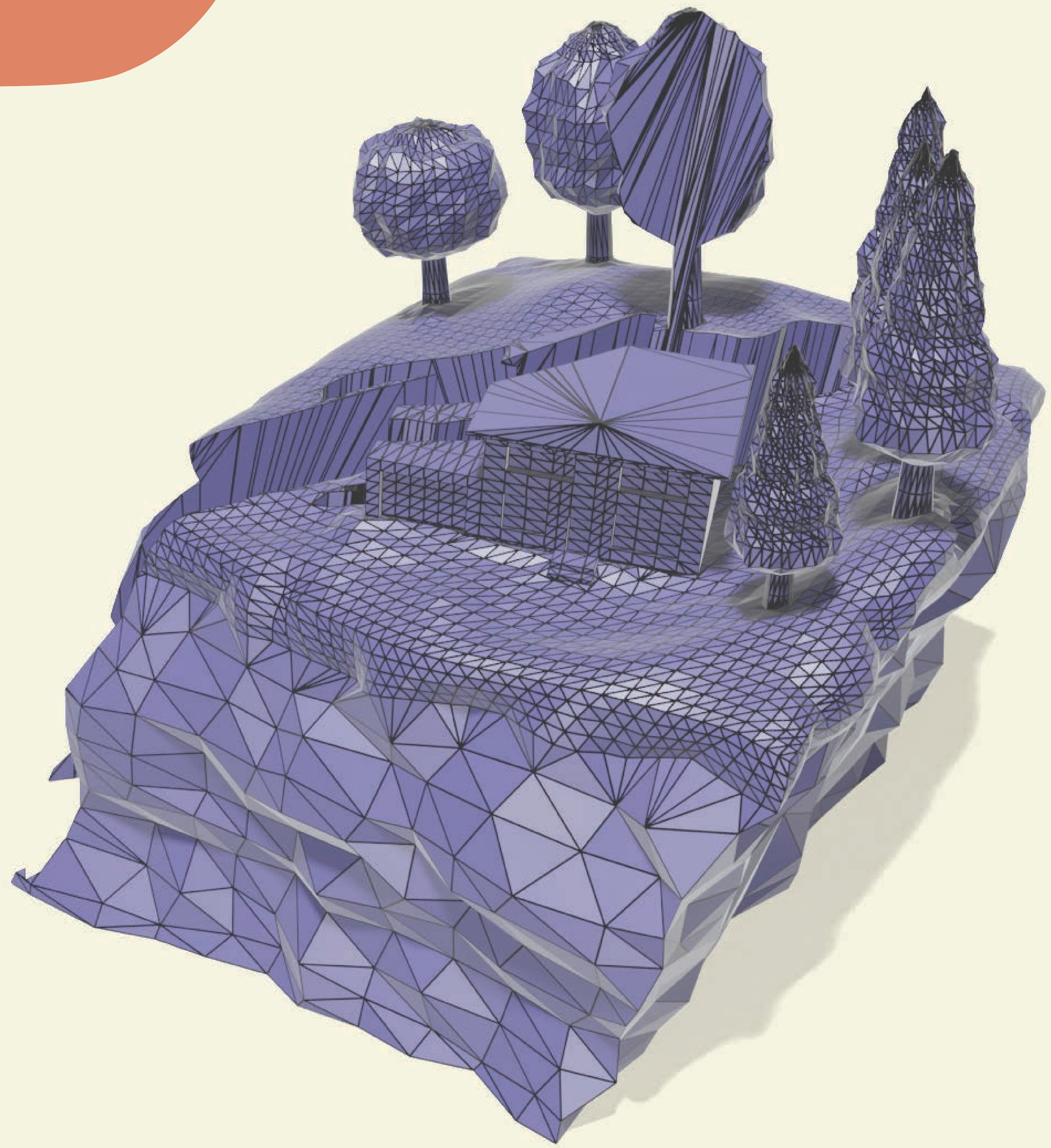


Triangle meshes can be very frustrating

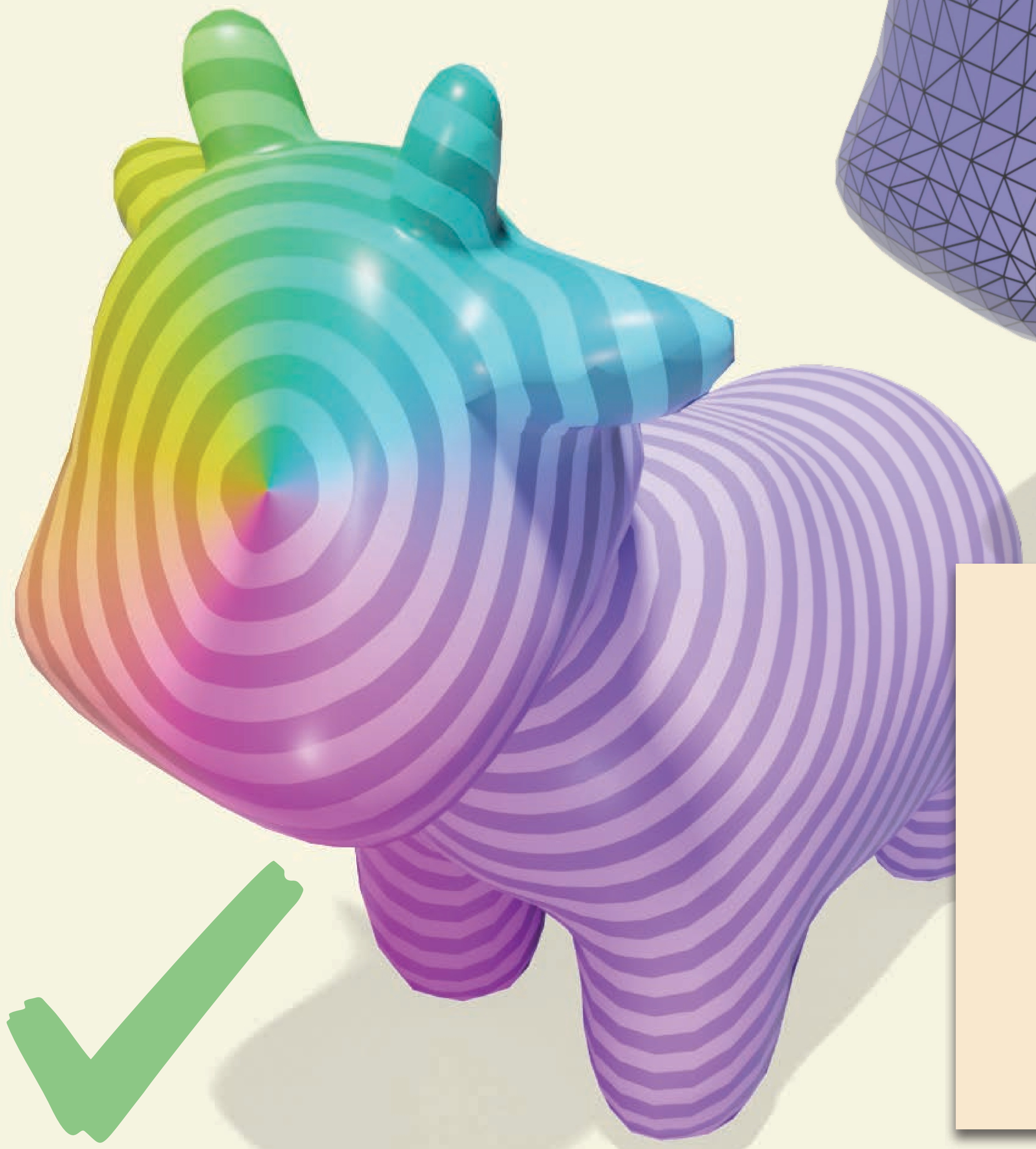


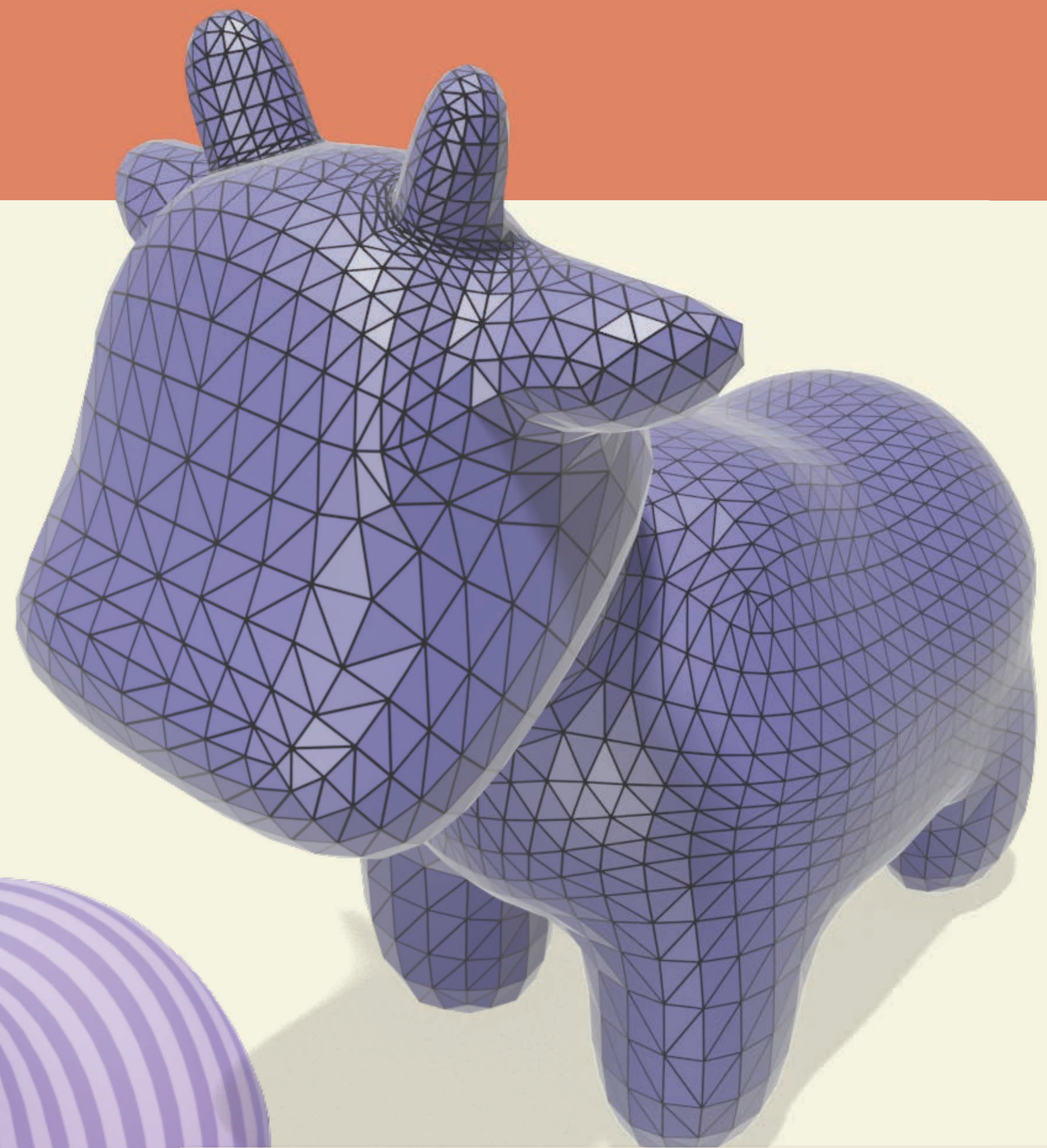
Triangle meshes can be very frustrating

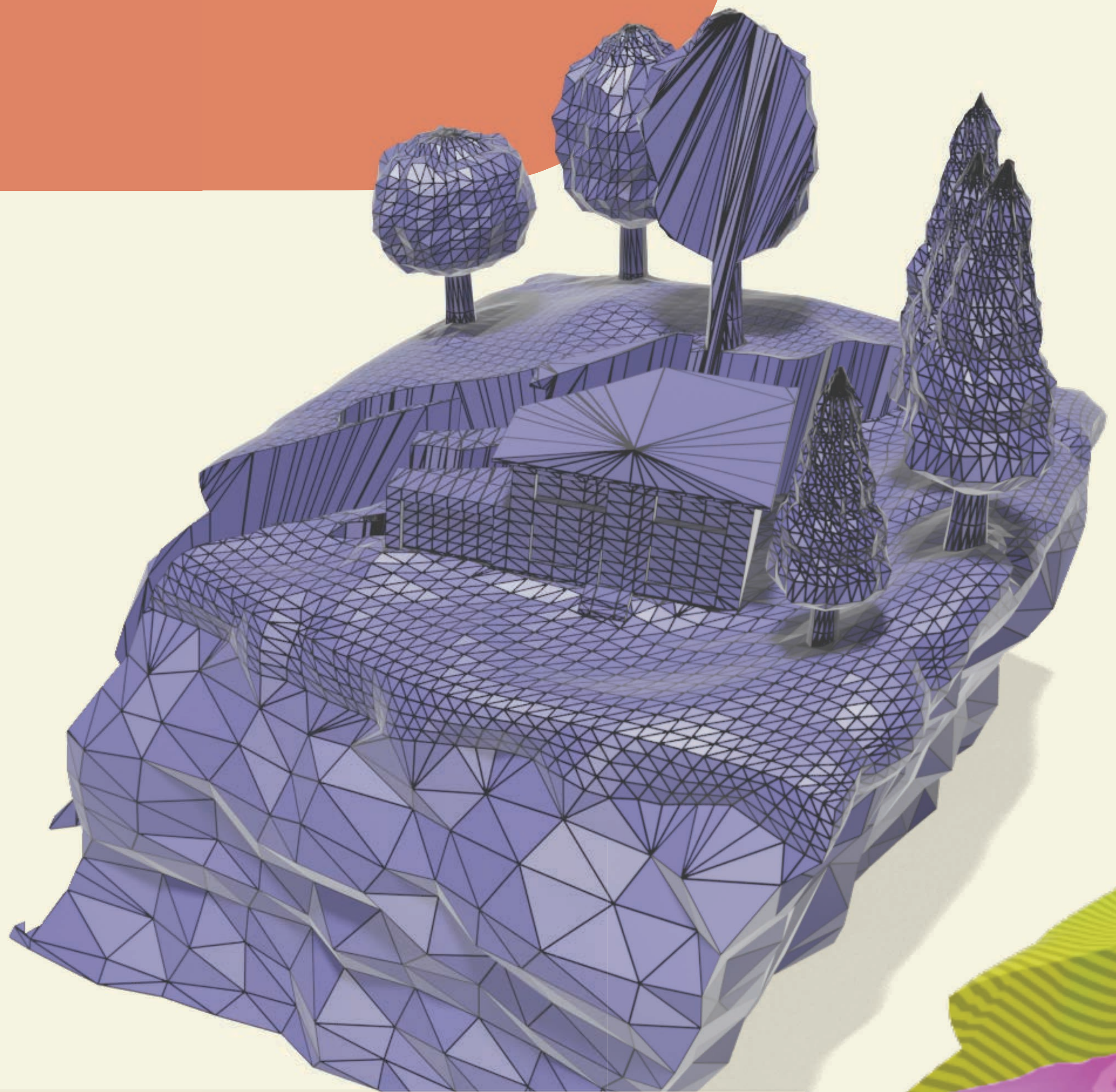


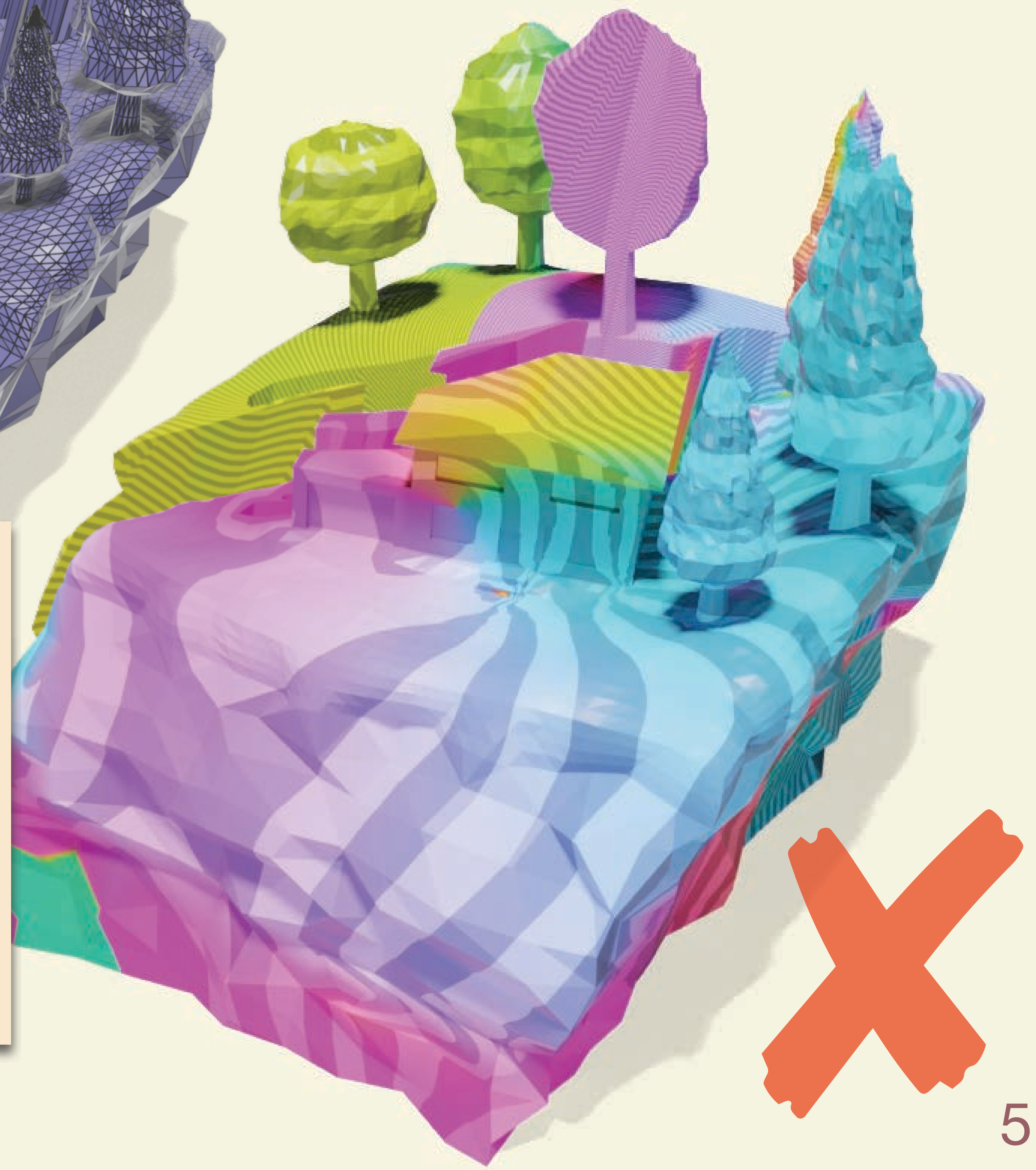


Triangle meshes can be very frustrating



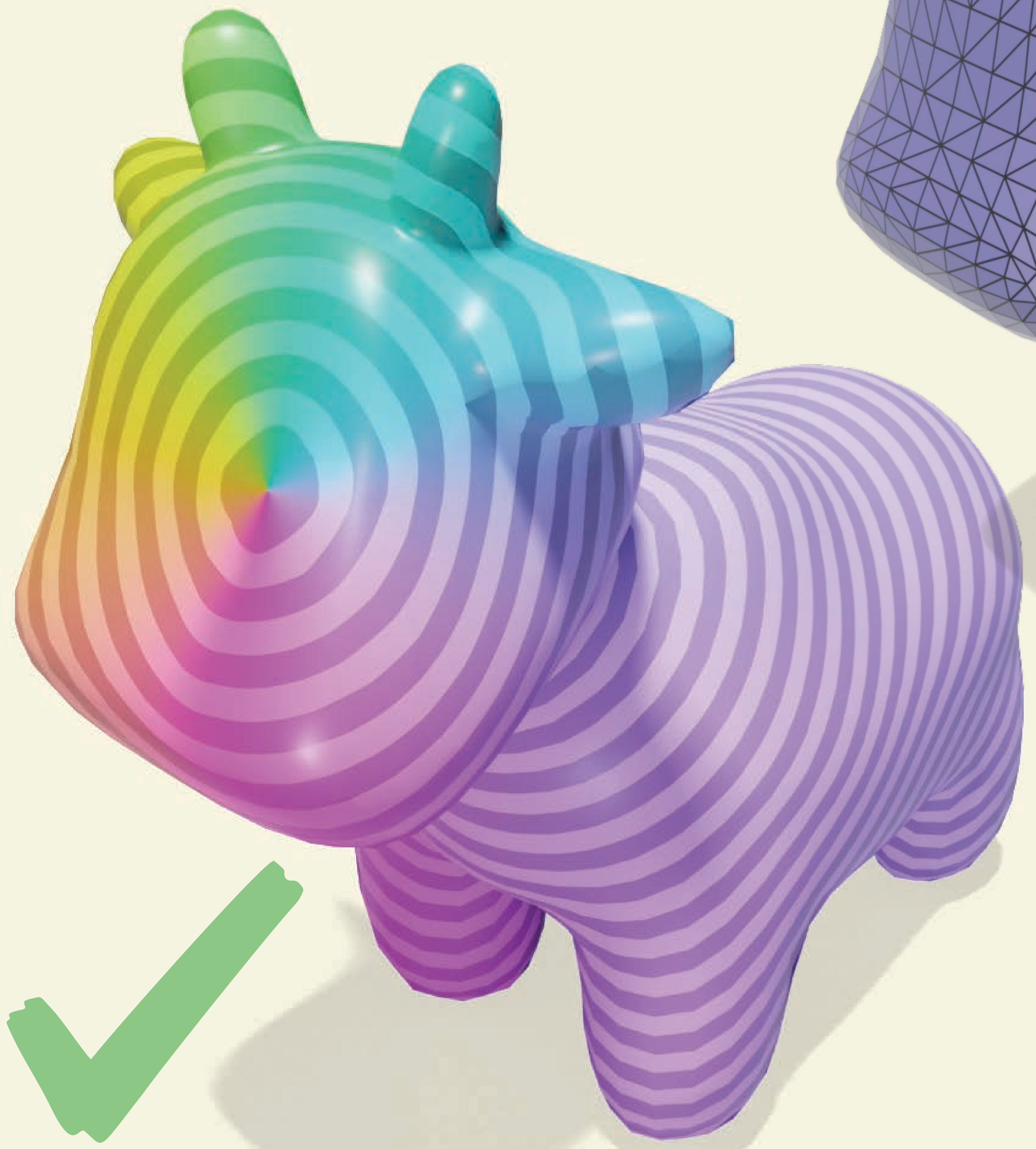
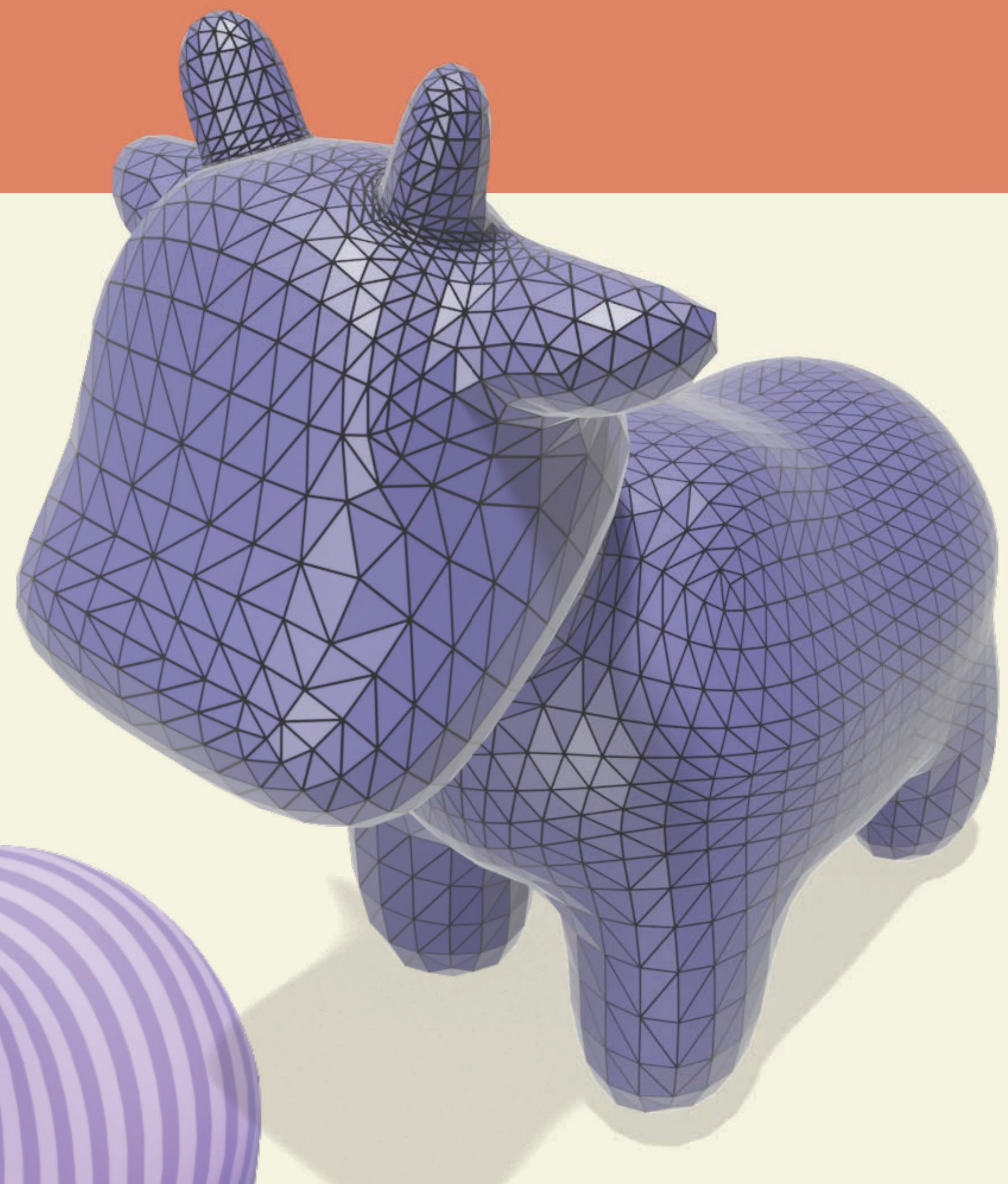
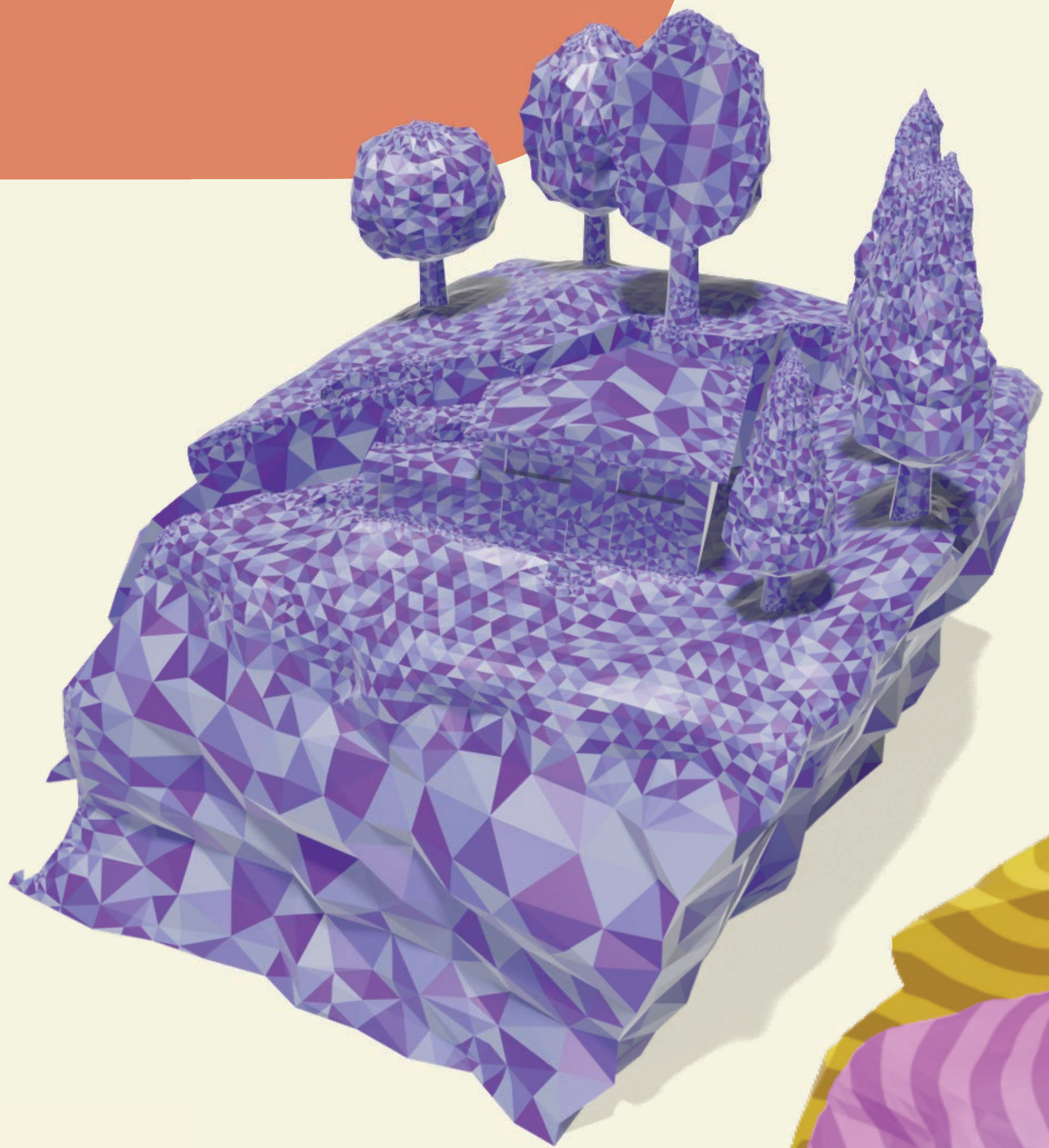
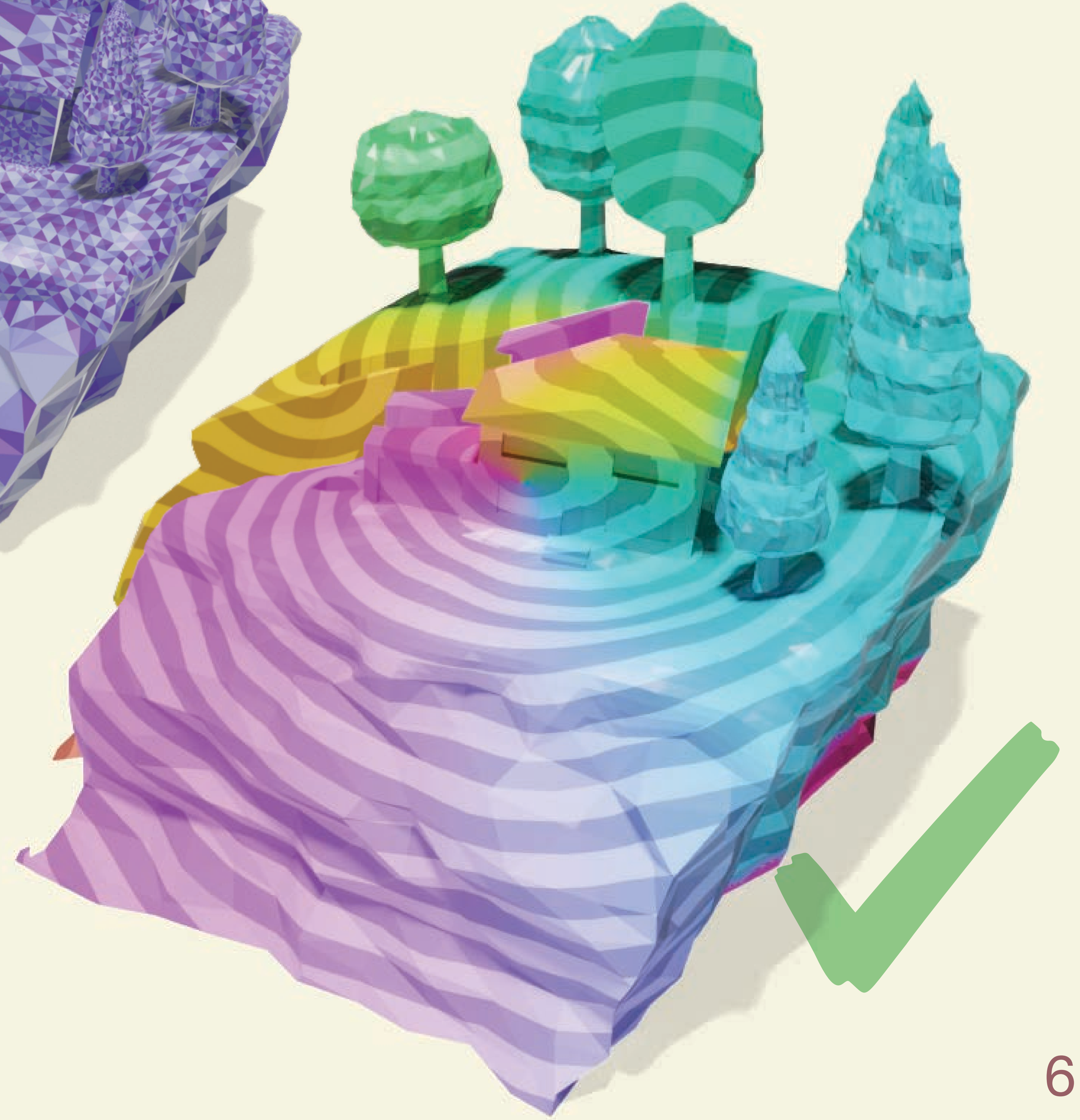






Problem: a mesh encodes both the *geometry* of a surface and a *function spaces* on that surface.

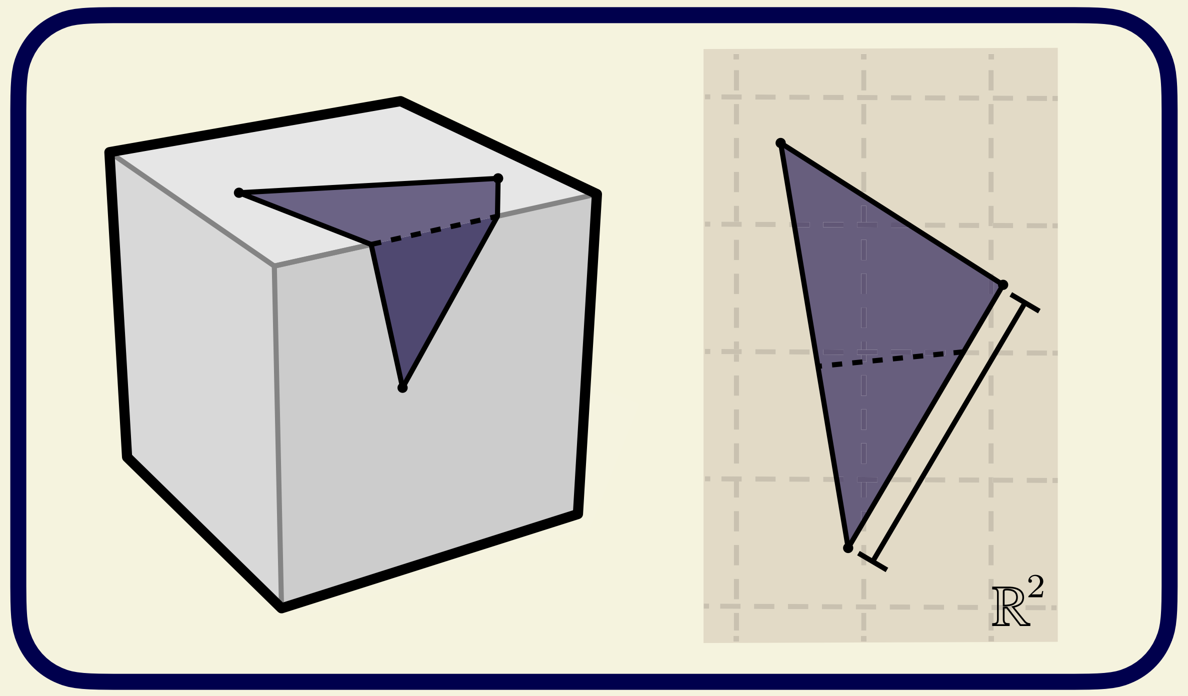
Triangle meshes can be very frustrating

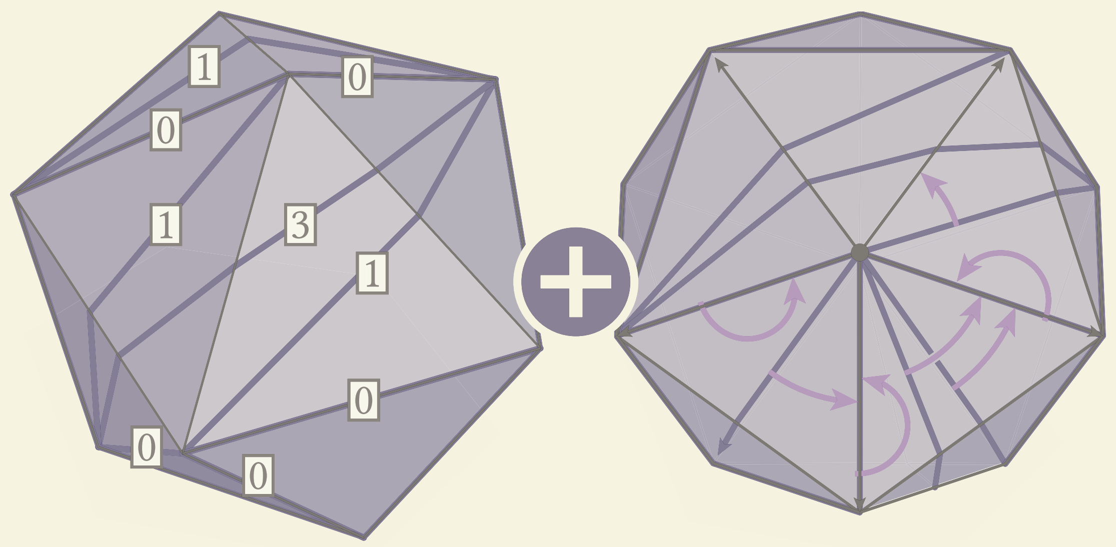





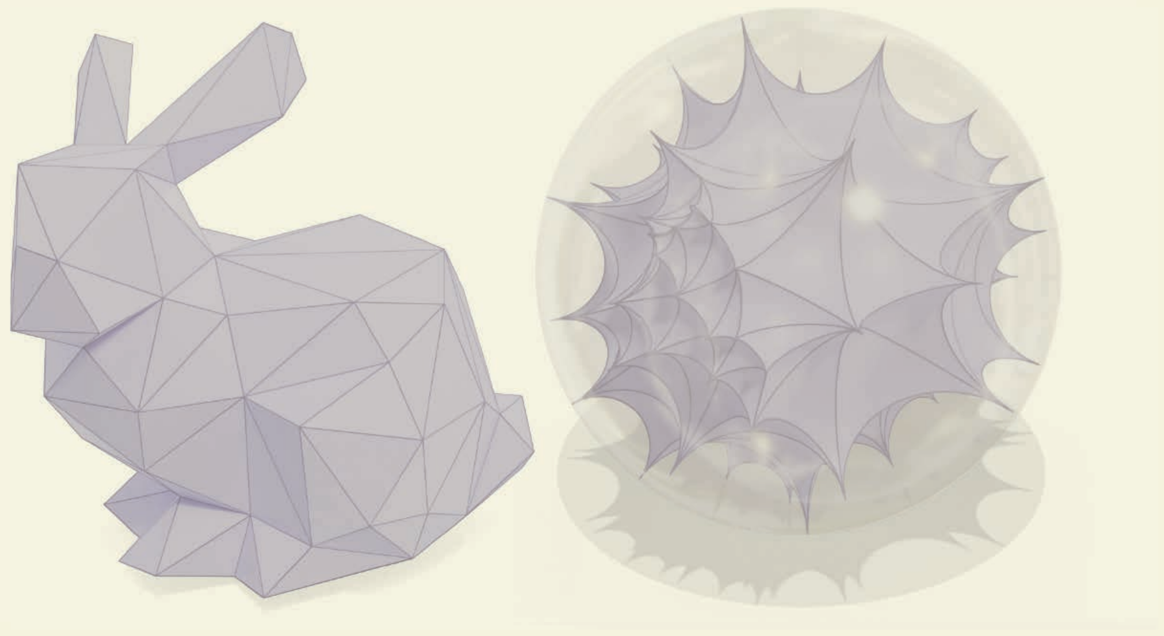
Intrinsic triangulations decouple surface geometry from other concerns

Outline

I. Preliminaries

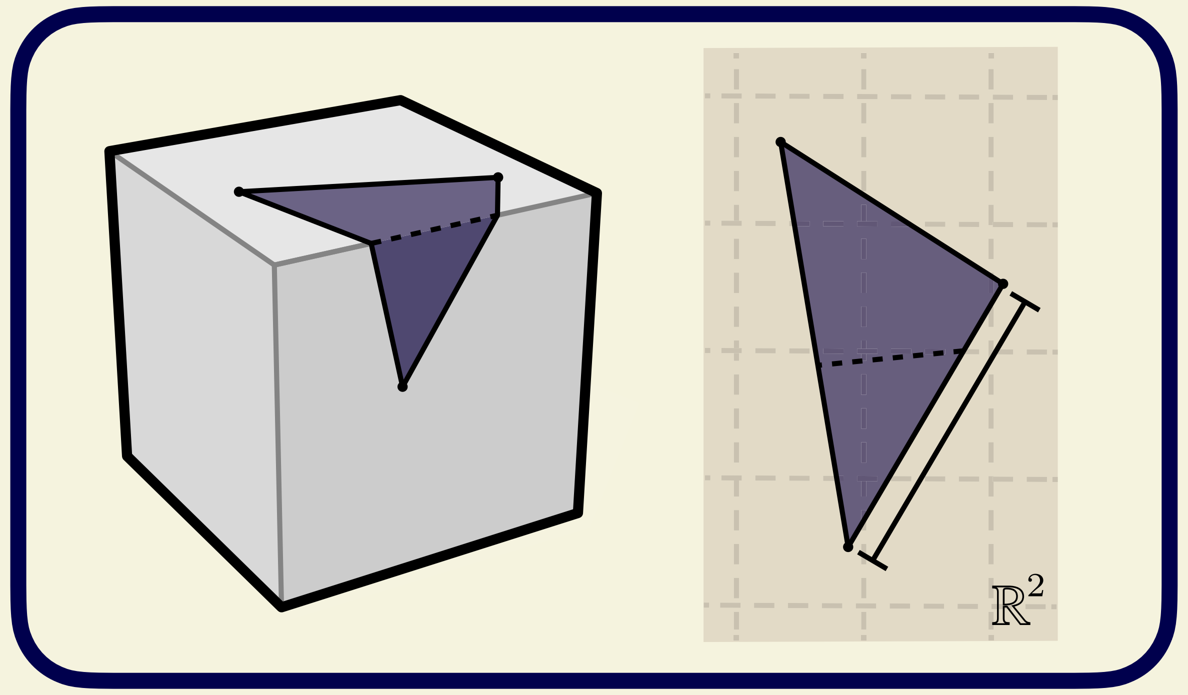




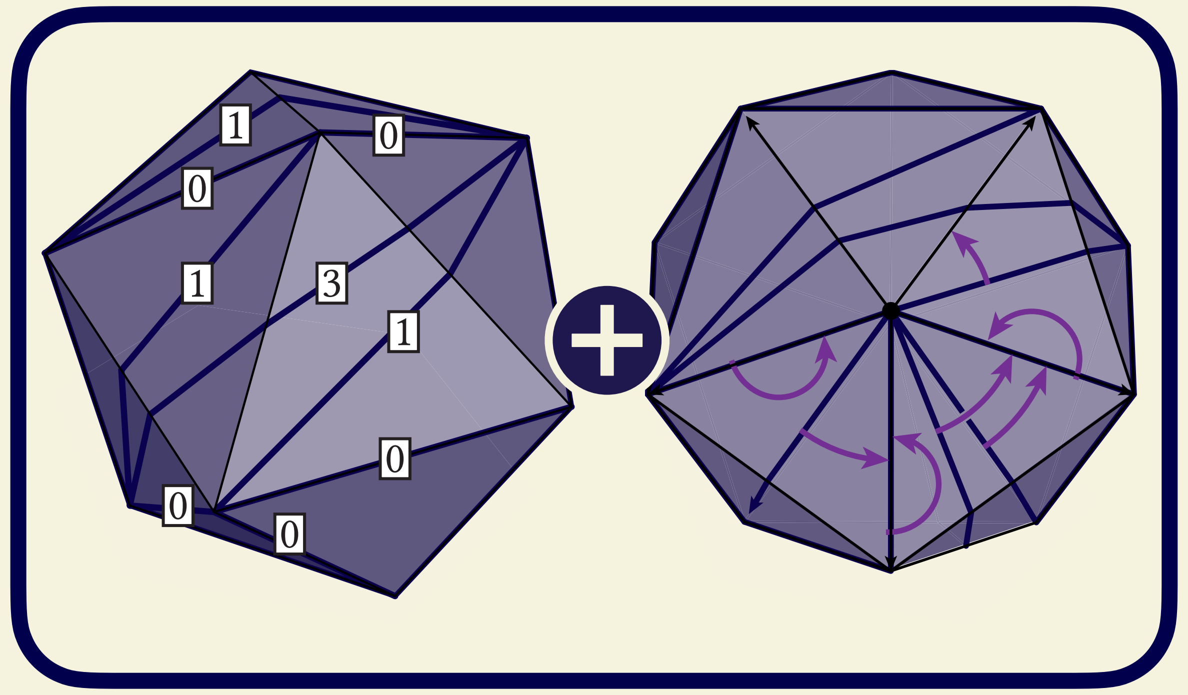


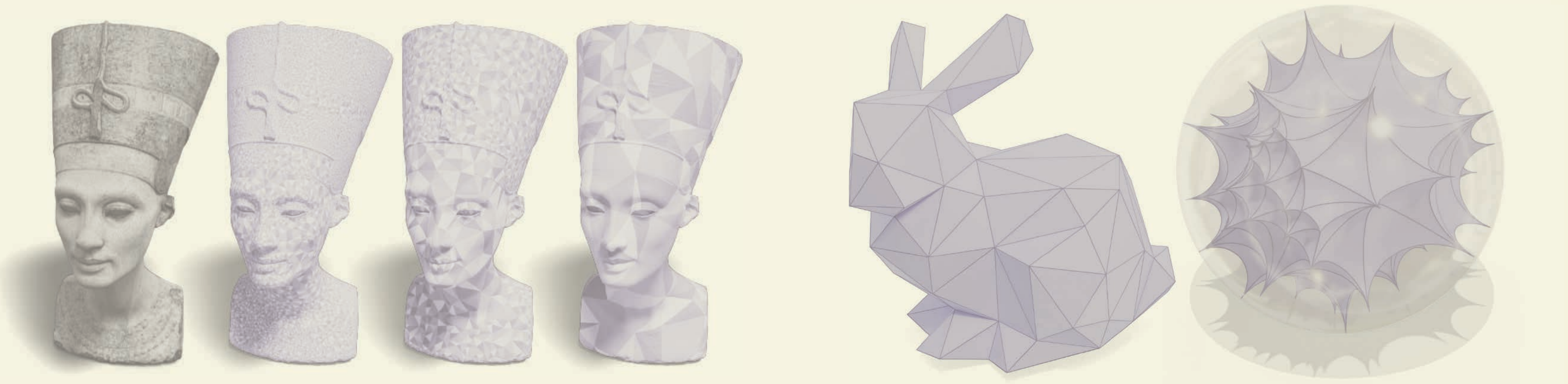
Outline

I. Preliminaries



II. Data structures for intrinsic triangulations



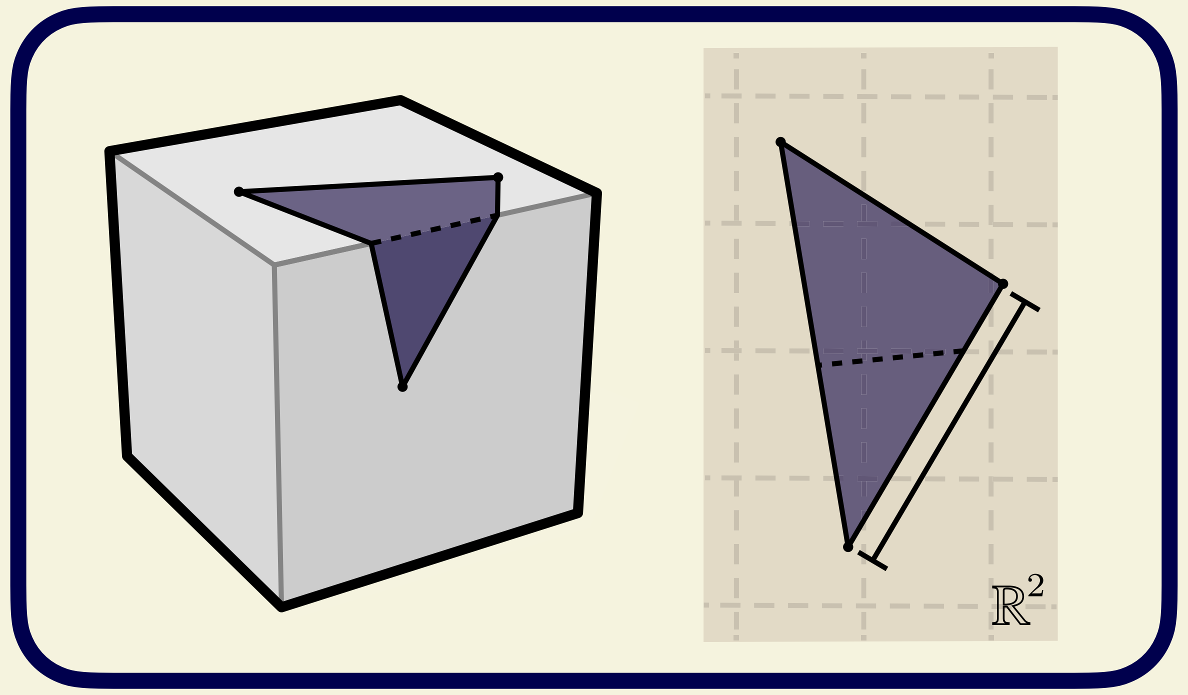


[**G.**, Sharp, & Crane. 2021. Integer coordinates for intrinsic geometry processing. *ACM TOG*]

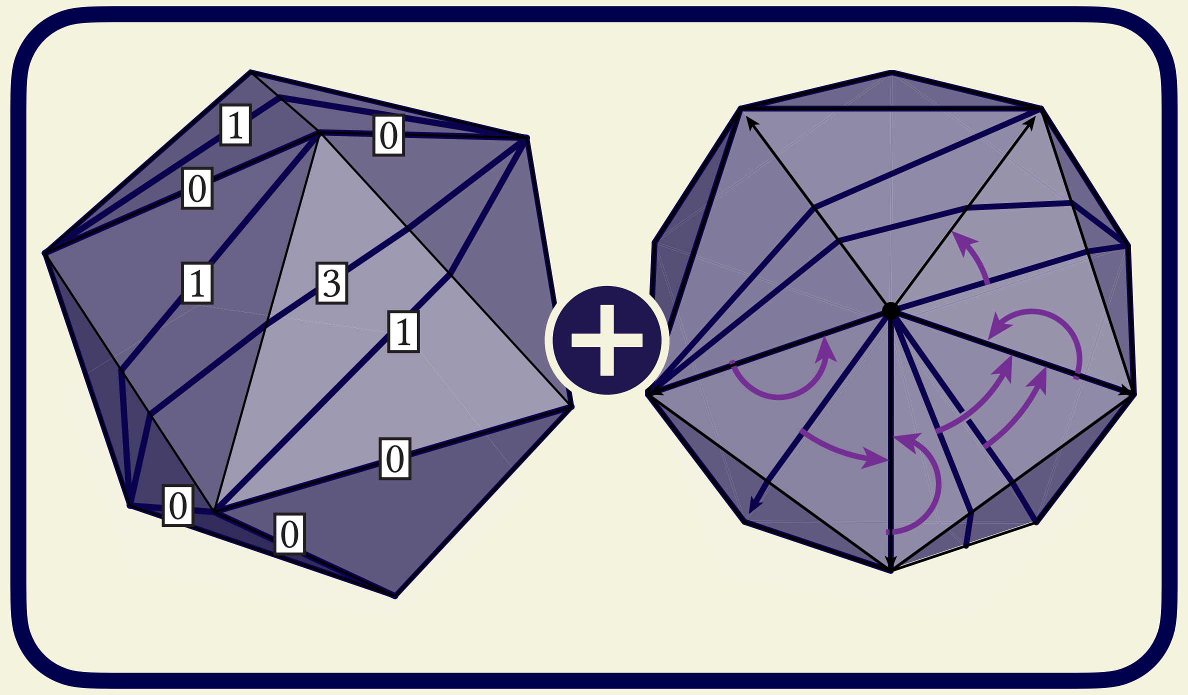
- Integer data structure for intrinsic triangulations
- Quality guarantees for intrinsic remeshing

Outline

I. Preliminaries



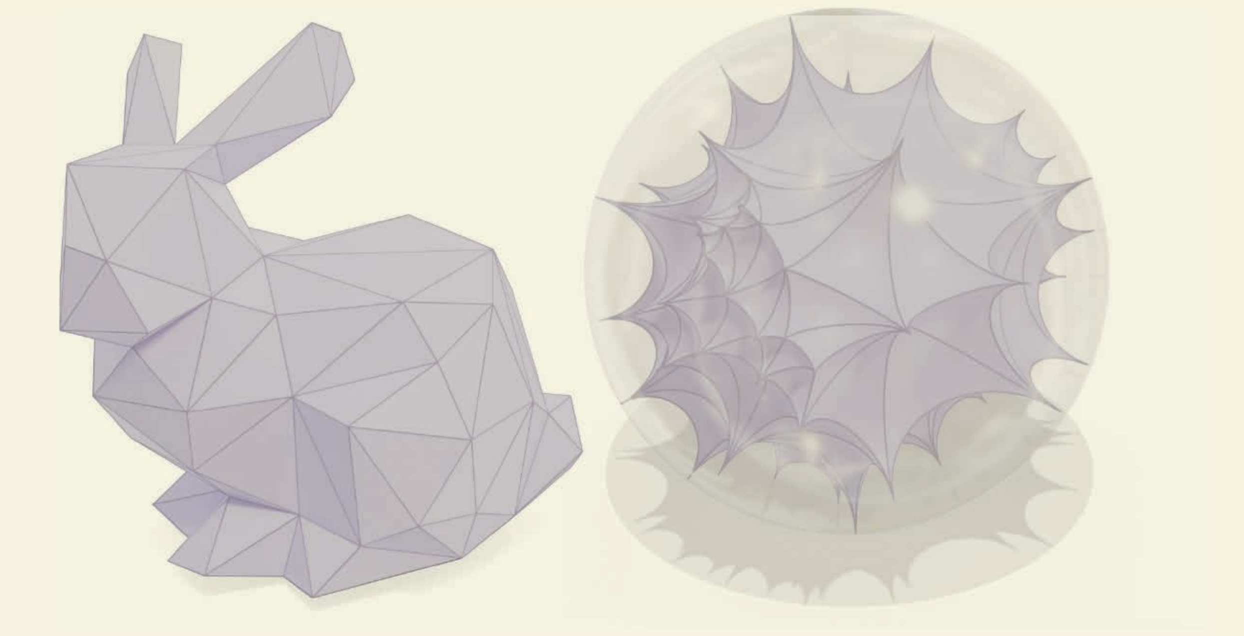
II. Data structures for intrinsic triangulations



[**G.**, Sharp, & Crane. 2021. Integer coordinates for intrinsic geometry processing. *ACM TOG*]

- Integer data structure for intrinsic triangulations
- Quality guarantees for intrinsic remeshing

III. Simplification of intrinsic triangulations

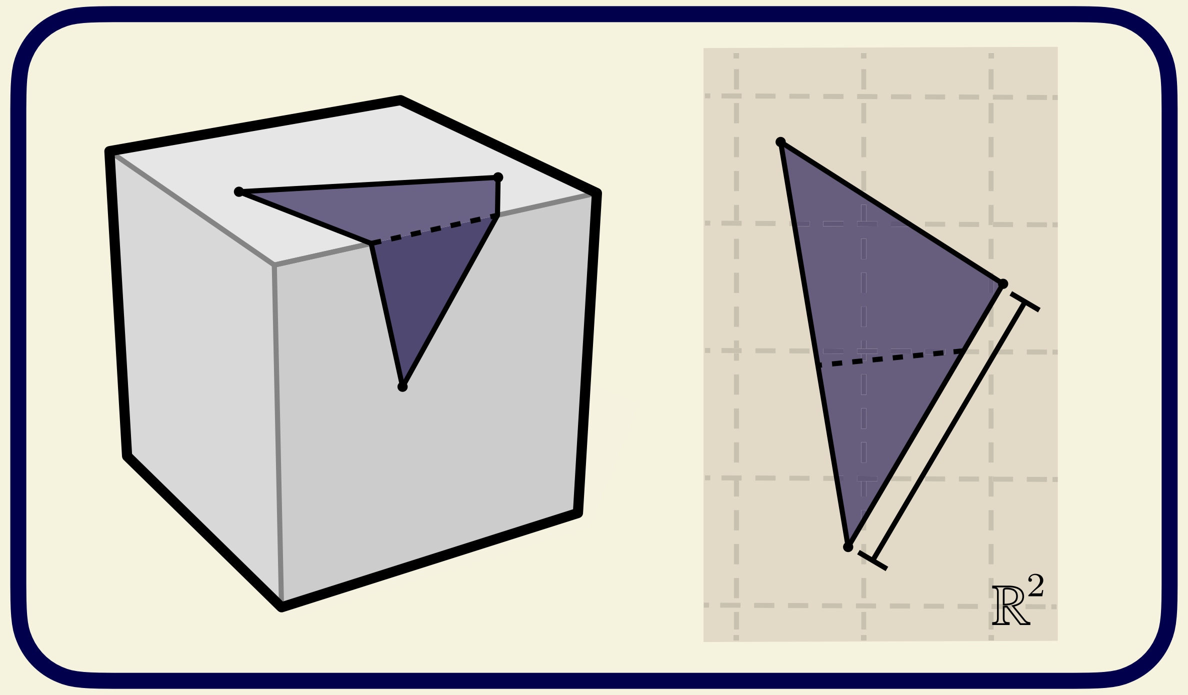


[Liu, **G.**, Chislett, Sharp, Jacobson & Crane. 2023. Surface Simplification using Intrinsic Error Metrics. *ACM TOG*]

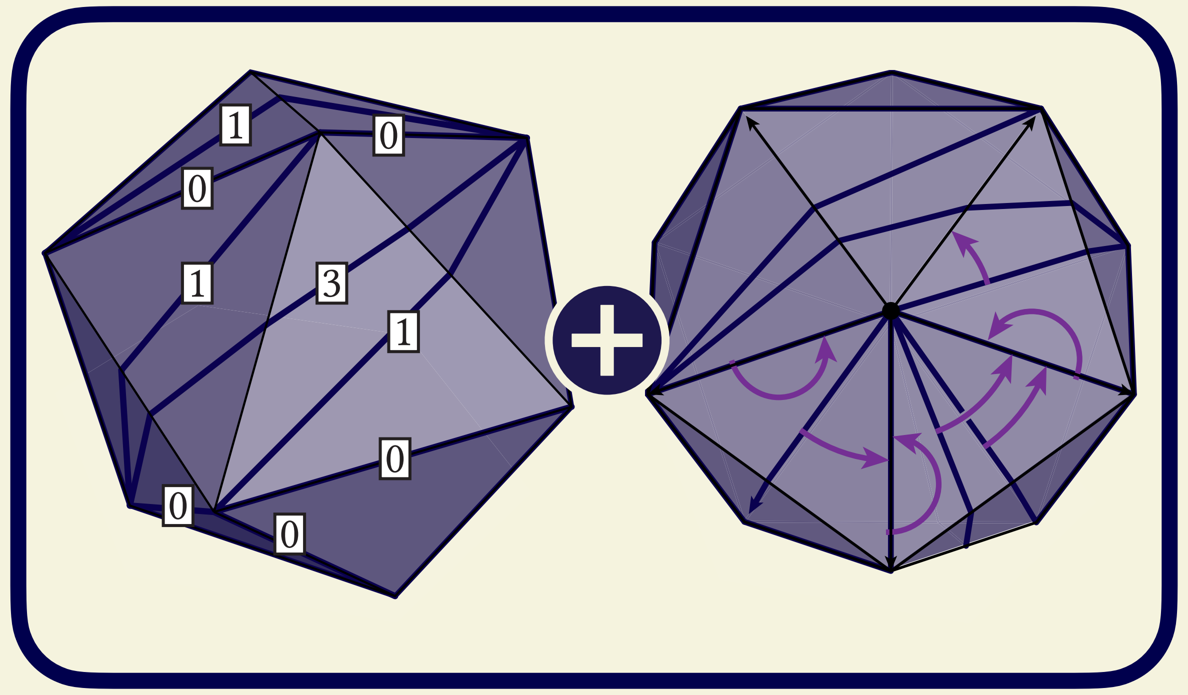
- First algorithm for intrinsic simplification
- New distortion measurement via *intrinsic curvature error*

Outline

I. Preliminaries



II. Data structures for intrinsic triangulations



[**G.**, Sharp, & Crane. 2021. Integer coordinates for intrinsic geometry processing. *ACM TOG*]

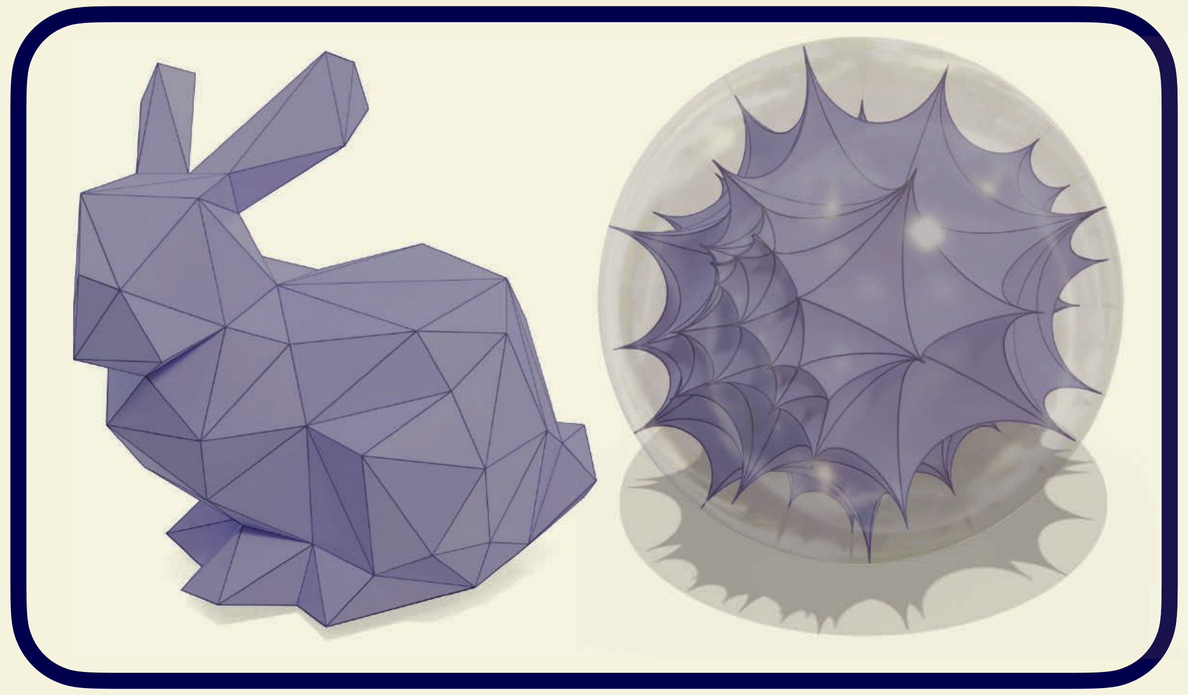
- Integer data structure for intrinsic triangulations
- Quality guarantees for intrinsic remeshing

III. Simplification of intrinsic triangulations

[Liu, **G.**, Chislett, Sharp, Jacobson & Crane. 2023. Surface Simplification using Intrinsic Error Metrics. *ACM TOG*]

- First algorithm for intrinsic simplification
- New distortion measurement via *intrinsic curvature error*

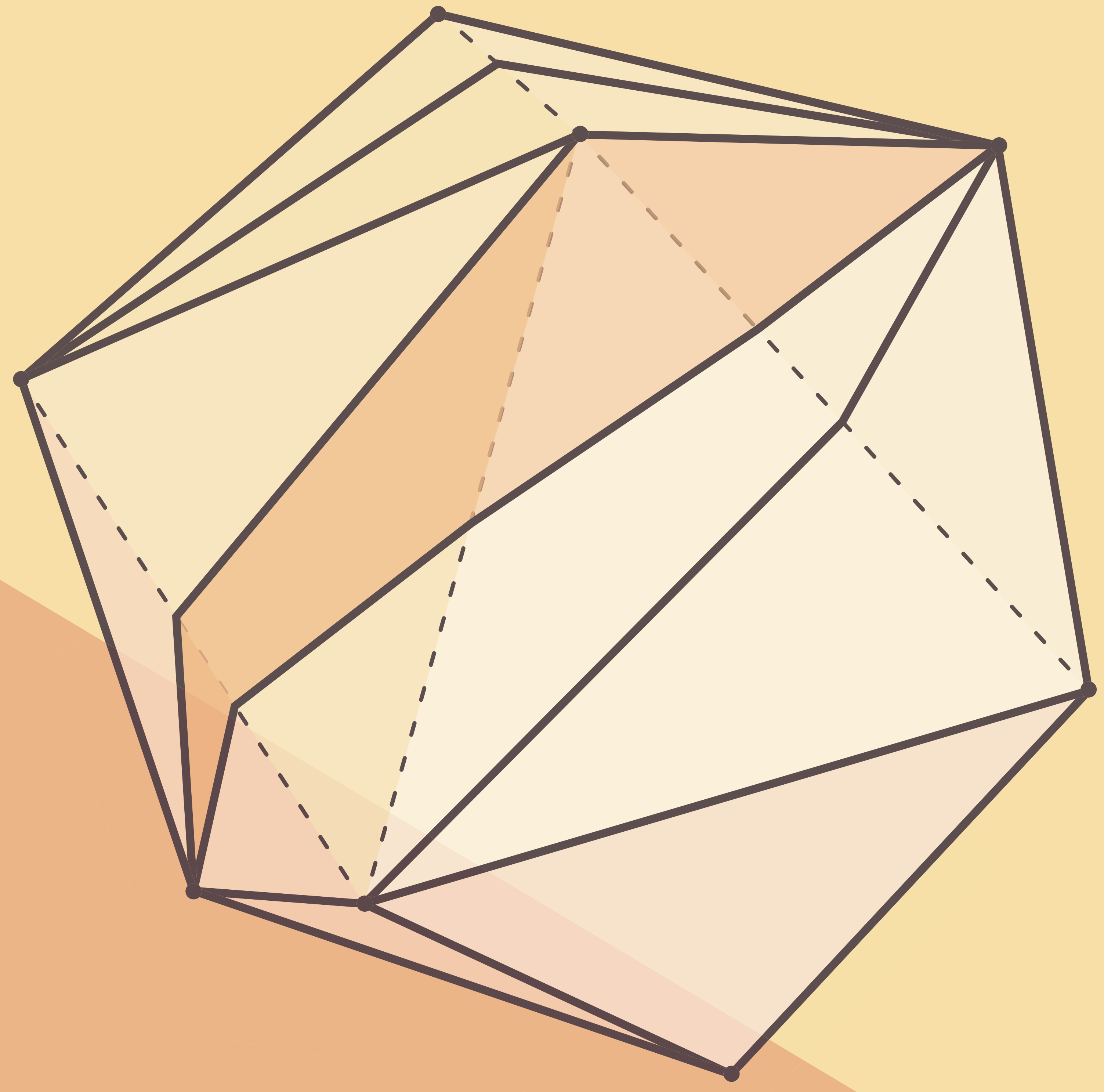
IV. Discrete conformal maps & uniformization



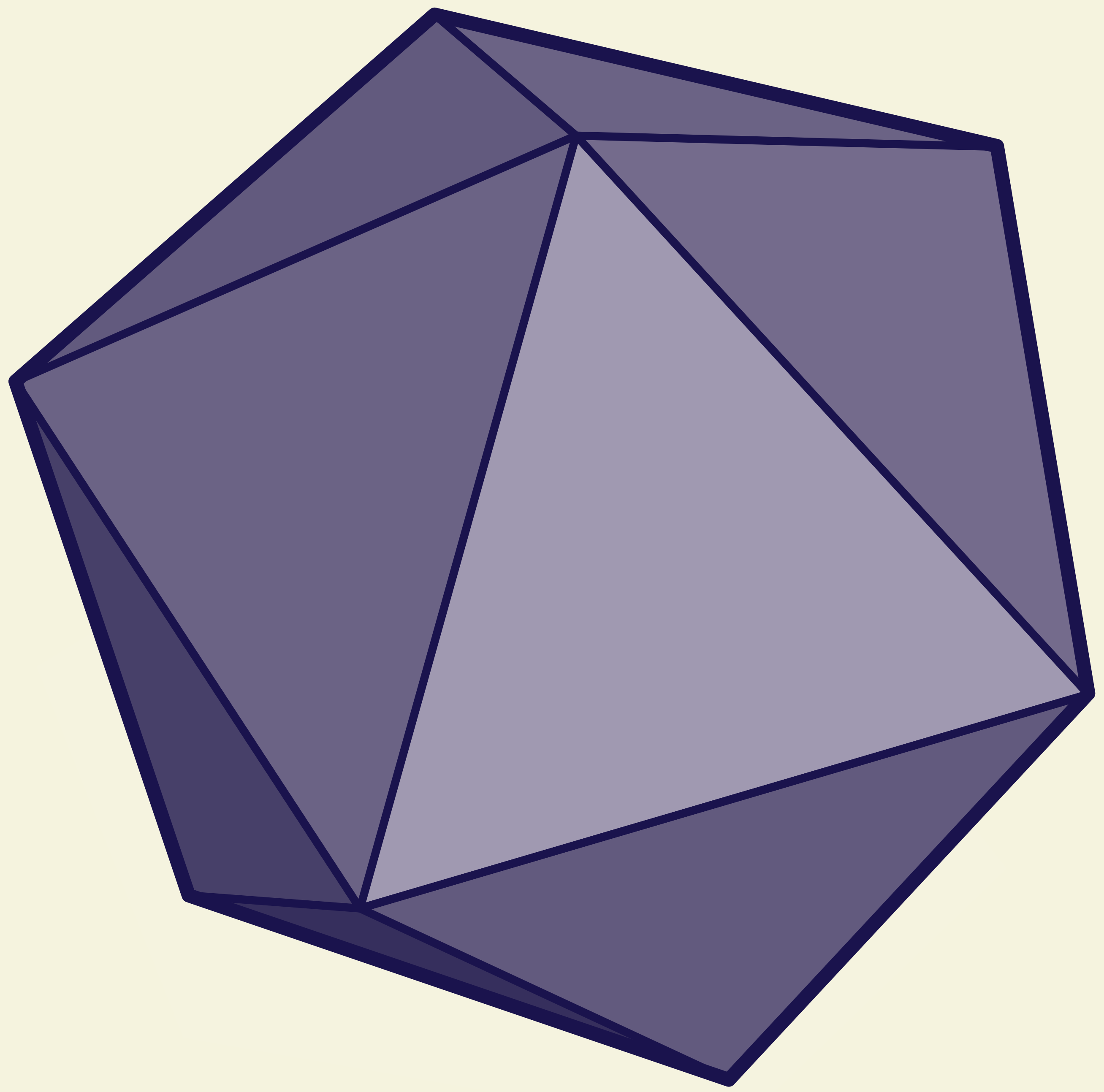
[**G.**, Springborn, & Crane. 2021. Discrete conformal equivalence of polyhedral surfaces. *ACM TOG*]

- Data structure for ideal hyperbolic polyhedra
- Interpolation for hyperbolic isometries
- Careful treatment of numerics

I. Preliminaries



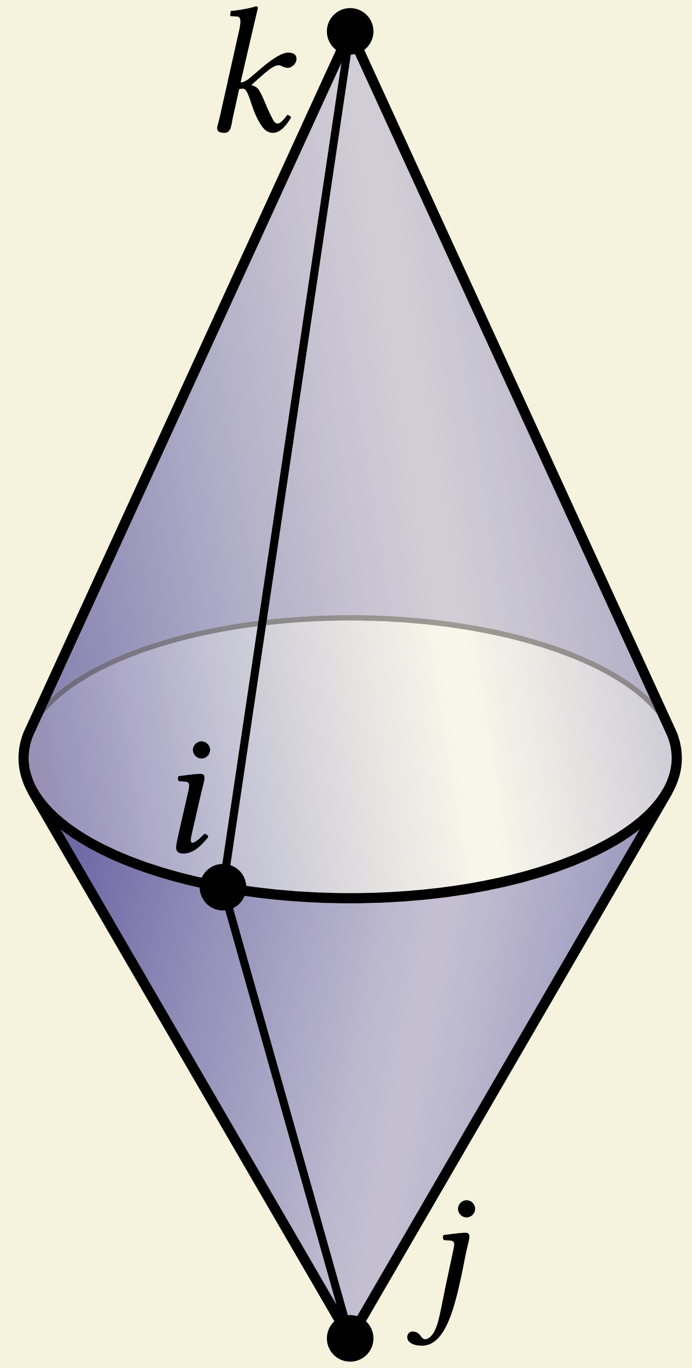
Triangulations



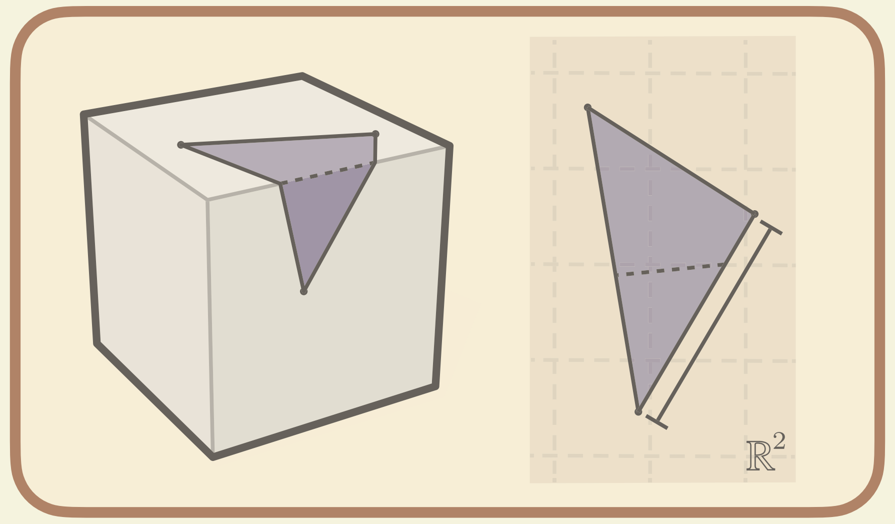
Definition

A *(surface) triangulation* is a manifold 2-dimensional cell complex $T = (V, E, F)$ whose faces are all triangles

- May be *irregular* (*e.g.*, two edges of a face may be glued together)



Intrinsic and extrinsic triangulations

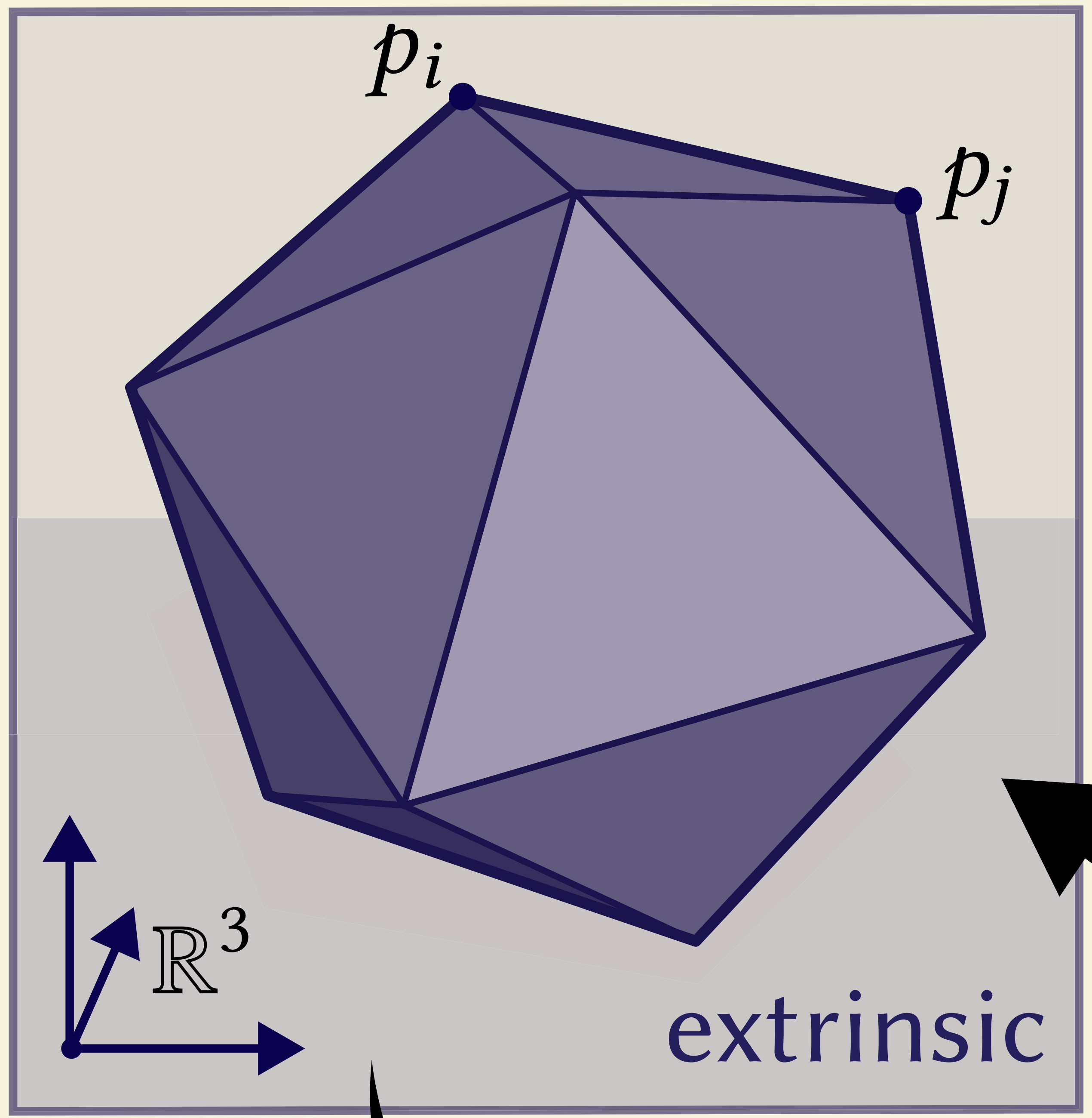


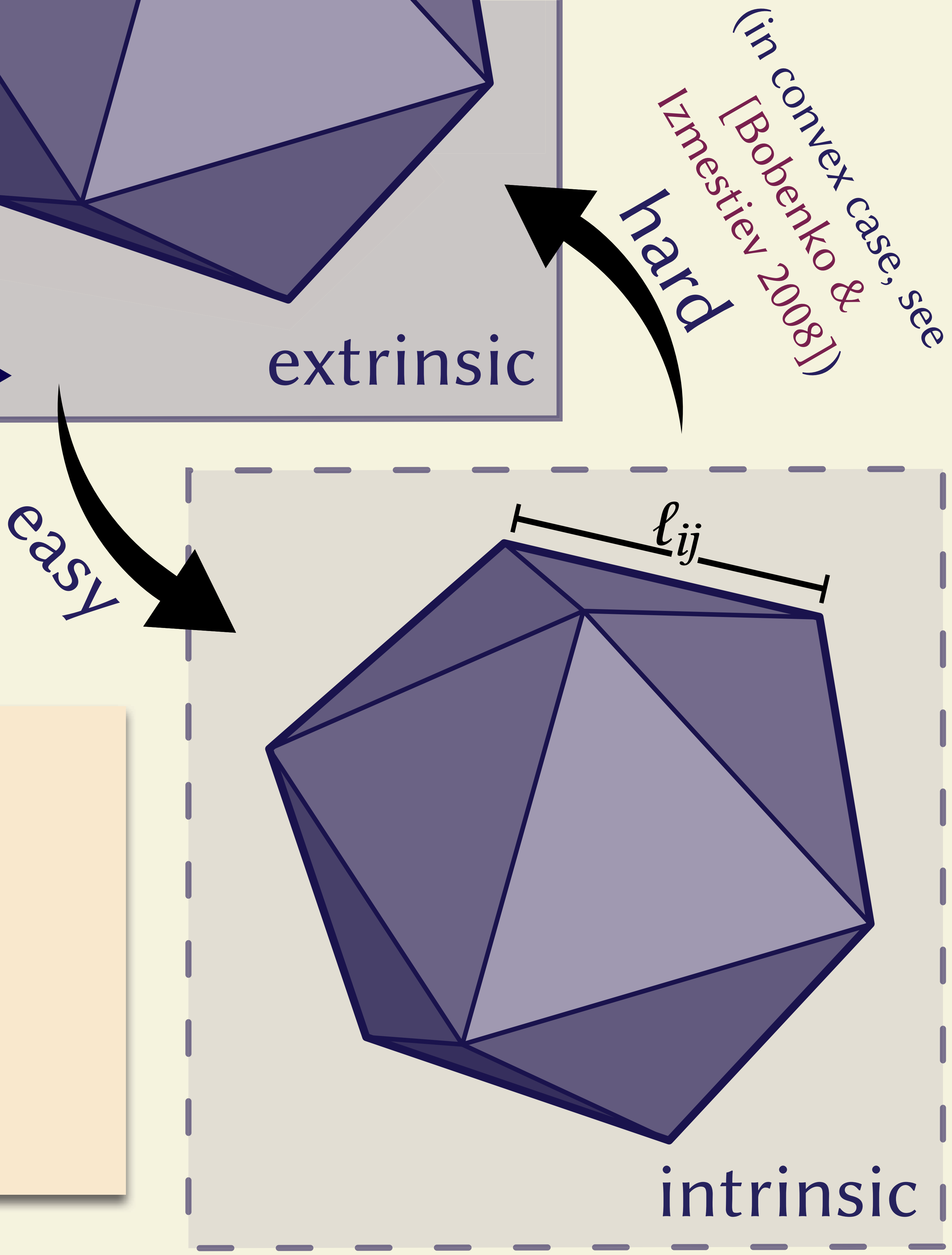
Definition

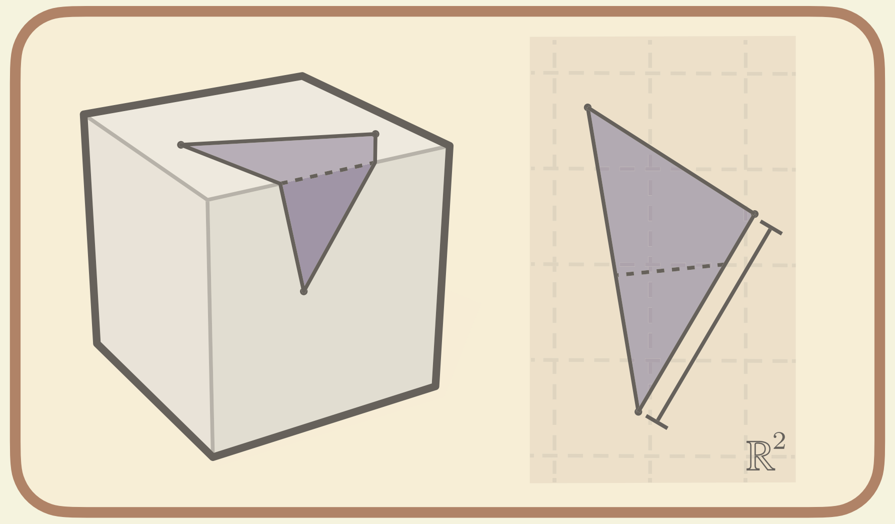
An *extrinsic triangulation* is a triangulation equipped with a piecewise-linear embedding into $\mathbb{R}^3$, *i.e.*, vertex positions $p : V \to \mathbb{R}^3$

Definition

An *intrinsic triangulation* is a triangulation equipped with positive edge lengths $\ell : E \to \mathbb{R}_{>0}$ satisfying the triangle inequality within each face

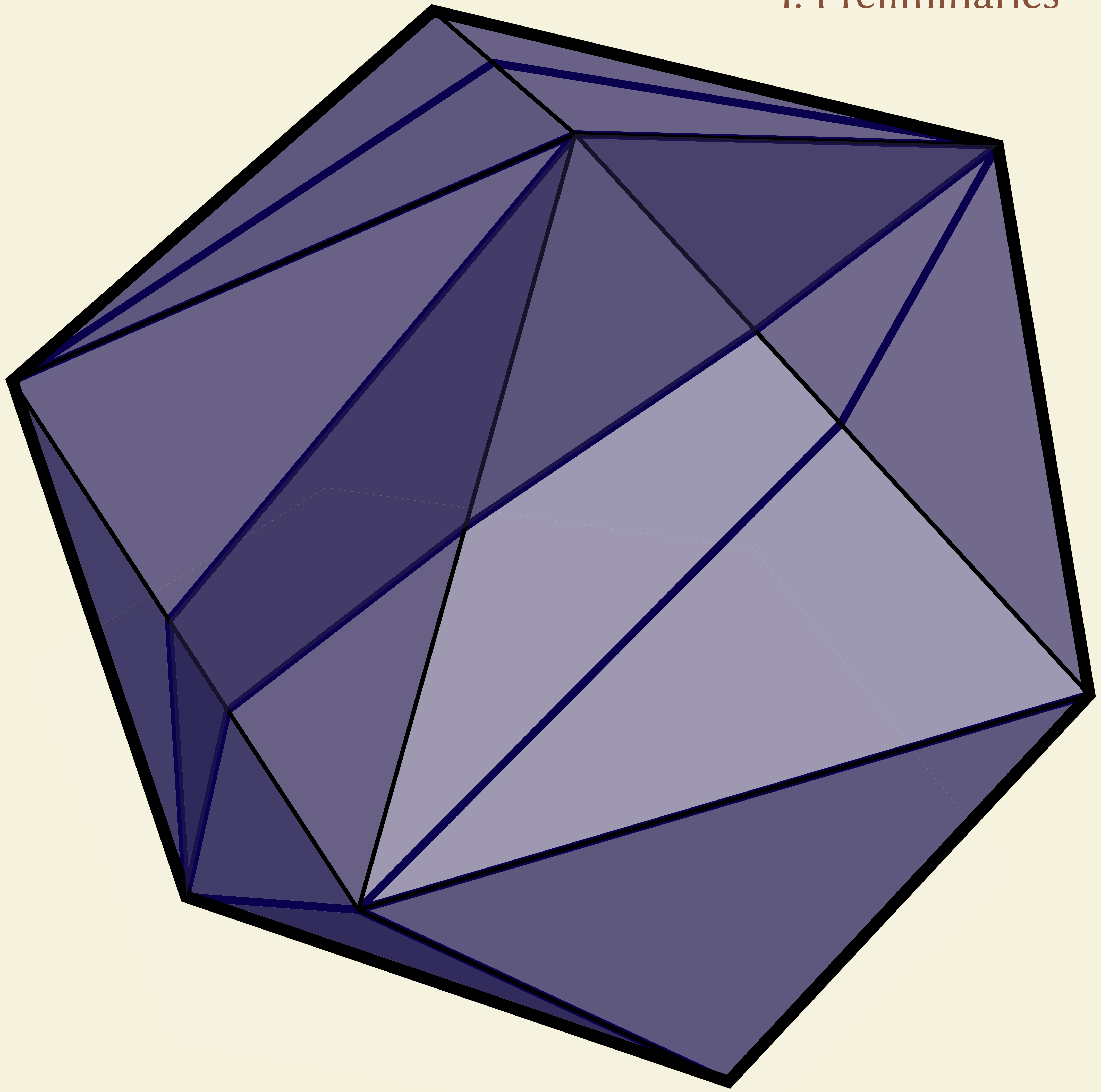


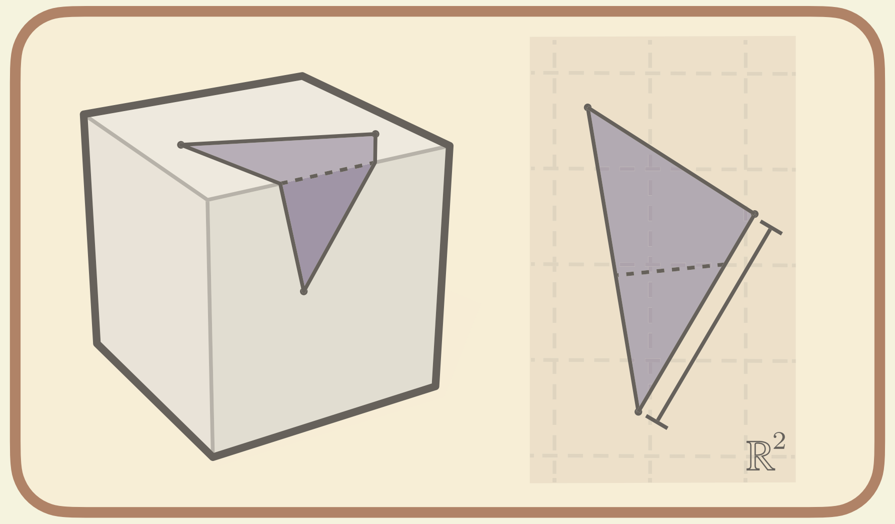




Correspondence

- Common case: intrinsic triangulation on top of extrinsic triangulation
    - *i.e.* isometric or at least homeomorphic to extrinsic triangulation
    - The *correspondence* is the homeomorphism mapping between them



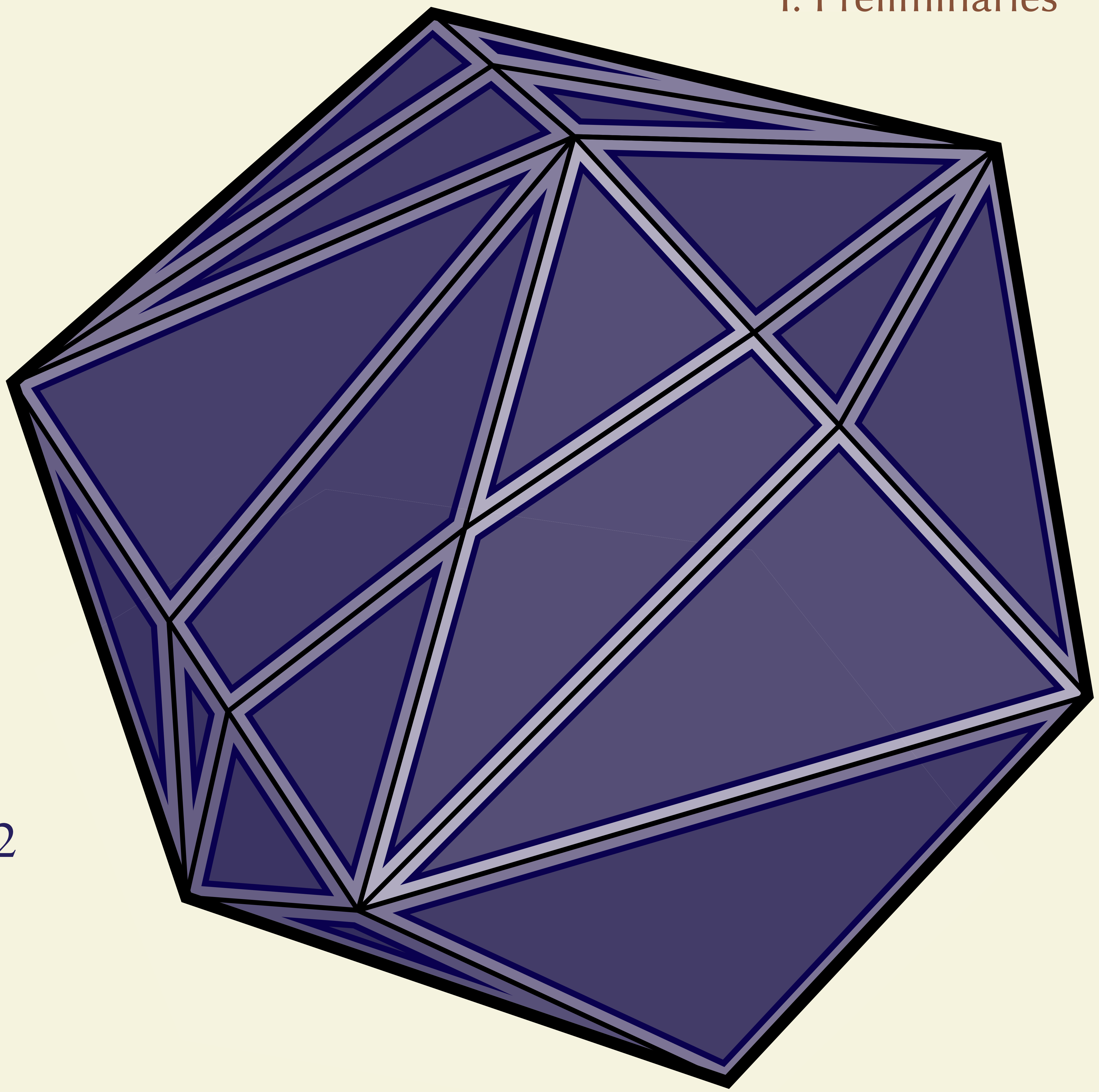


Common subdivision

Definition

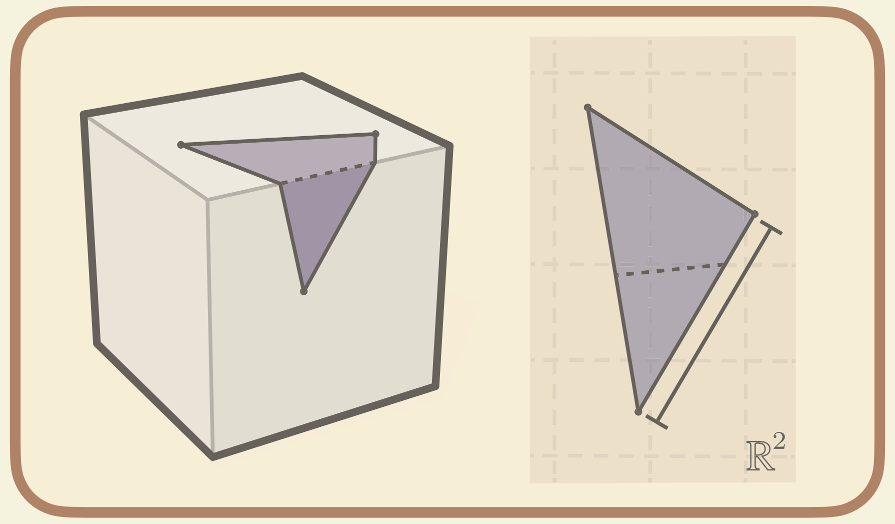
The *common subdivision* of two triangulations T_1, T_2 is the coarsest polygonal complex $\mathcal{S}$ such that all faces of T_1 or T_2 are unions of faces of $\mathcal{S}$

Intuitively, the result of cutting T_1 along edges of T_2

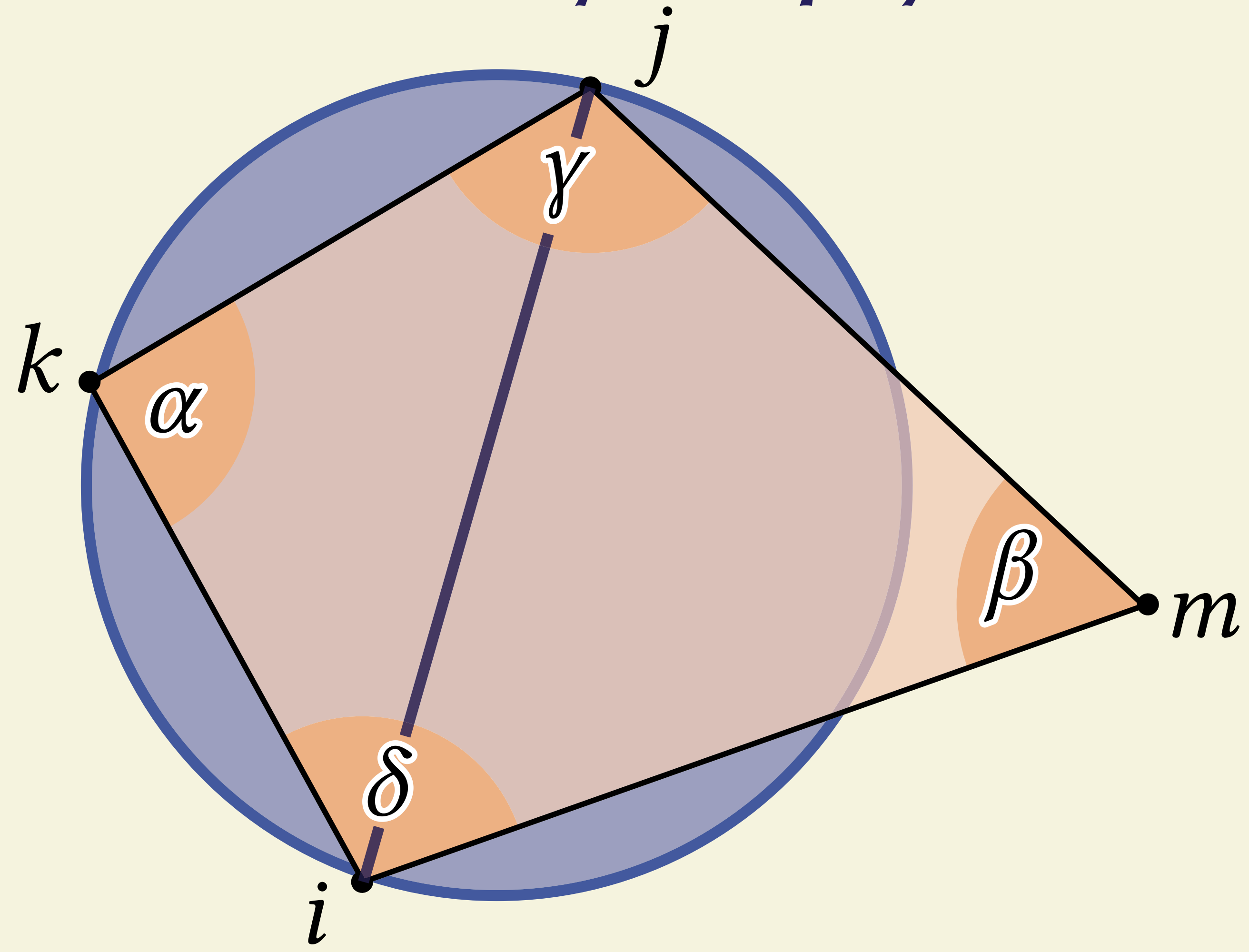


The space of intrinsic triangulations is large

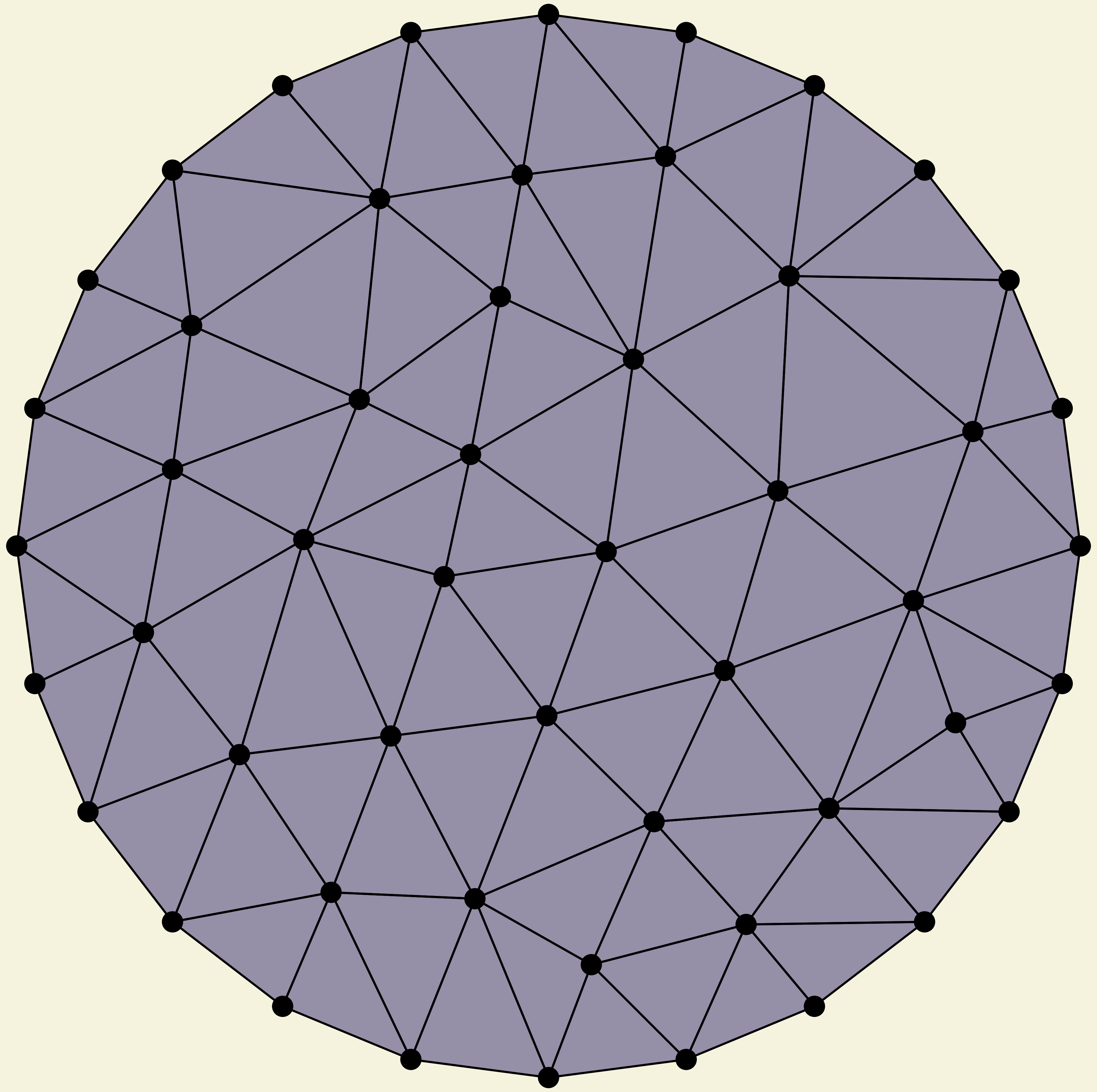
Delaunay triangulations



- Planar Delaunay triangulations have many nice properties:
  - Essentially unique, maximize angles lexicographically, minimize spectrum lexicographically, smoothest interpolation, positive cotan weights…
- Characterized by *empty circumcircle condition*

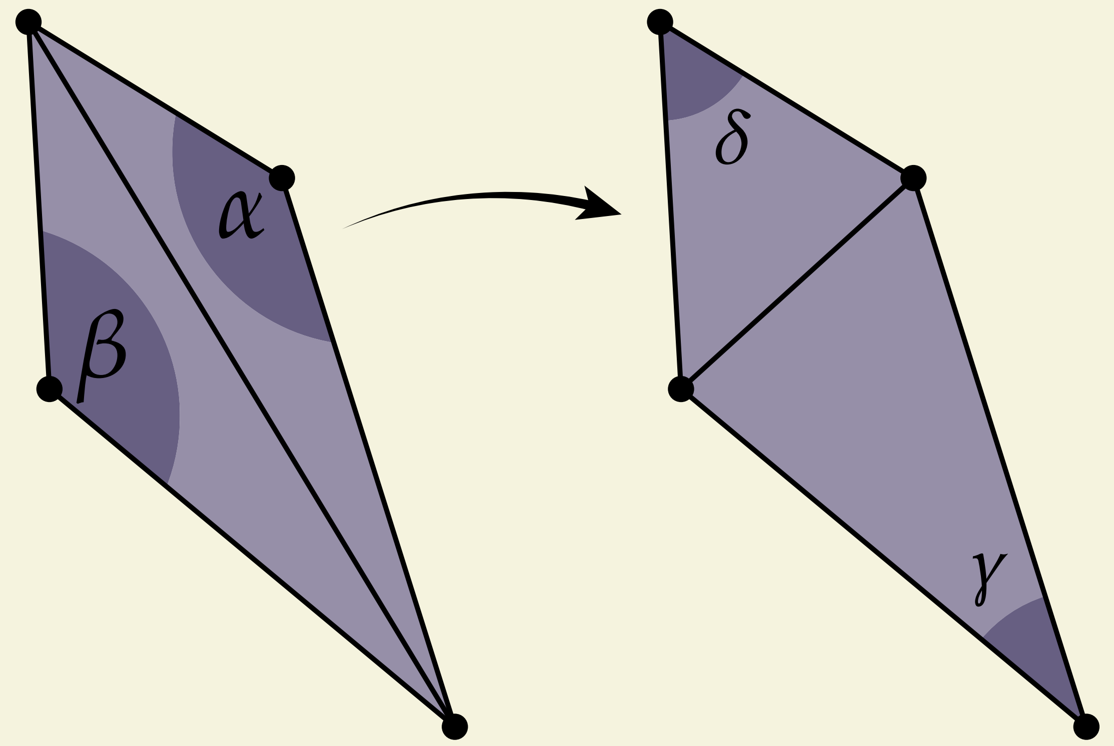


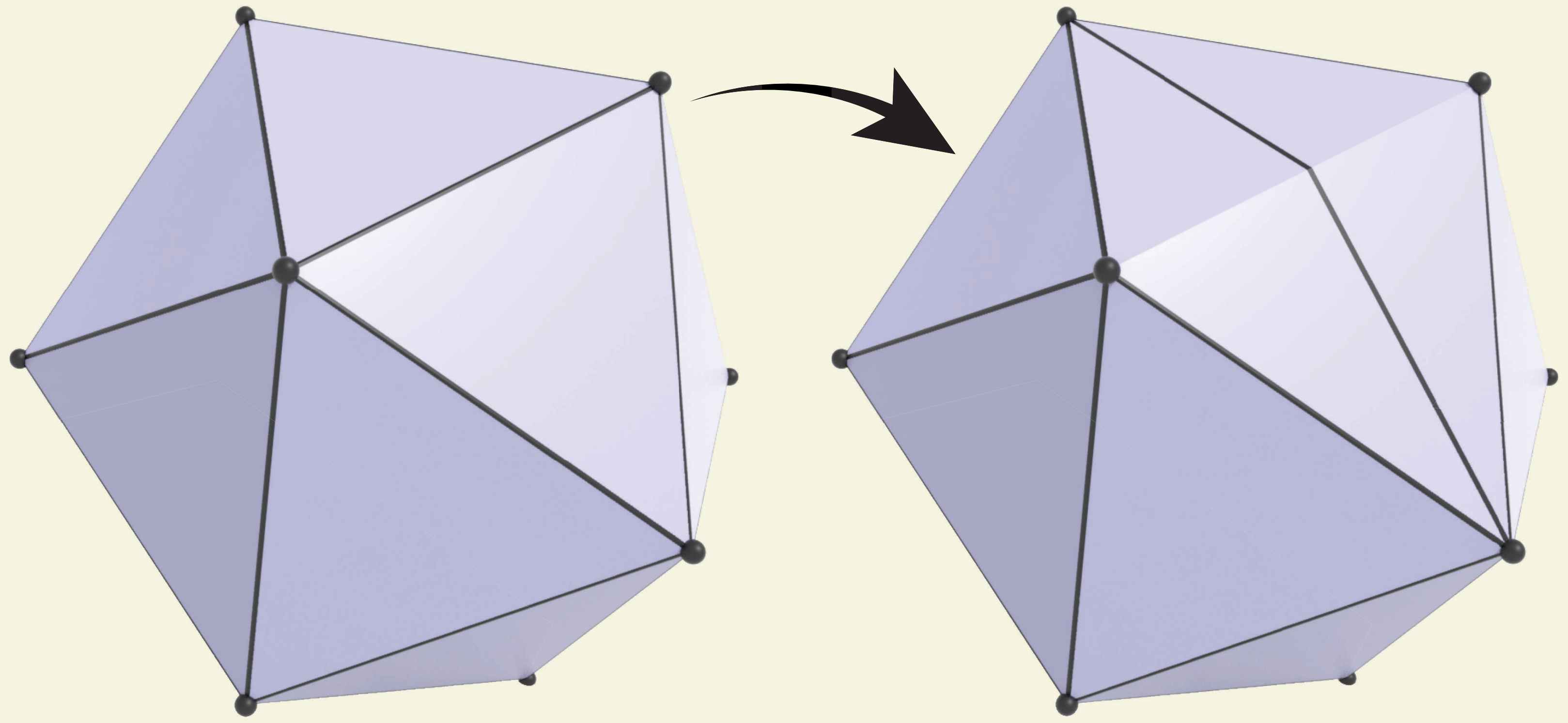
$$\alpha + \beta \leq \gamma + \delta$$

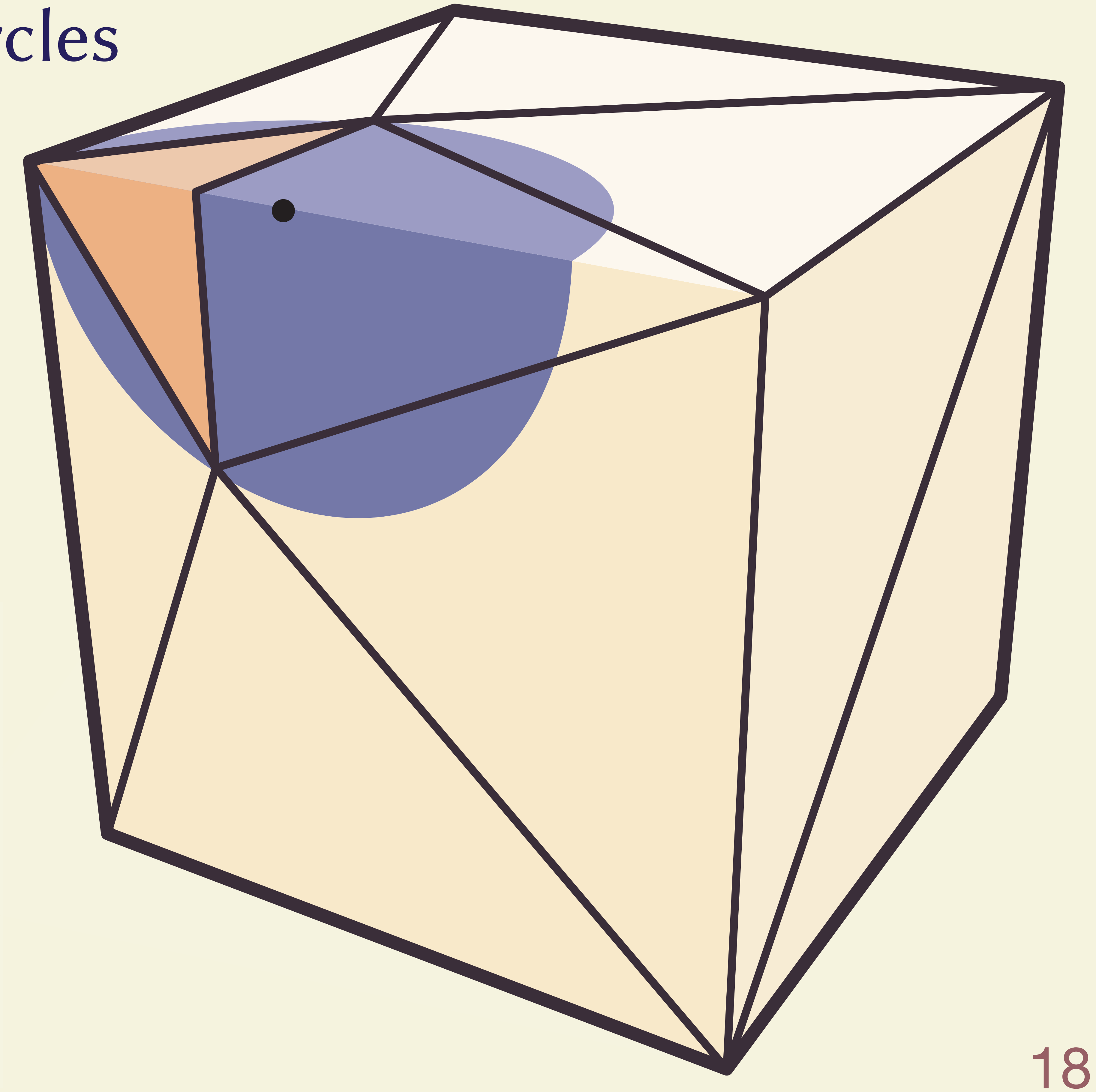


Intrinsic Delaunay triangulations

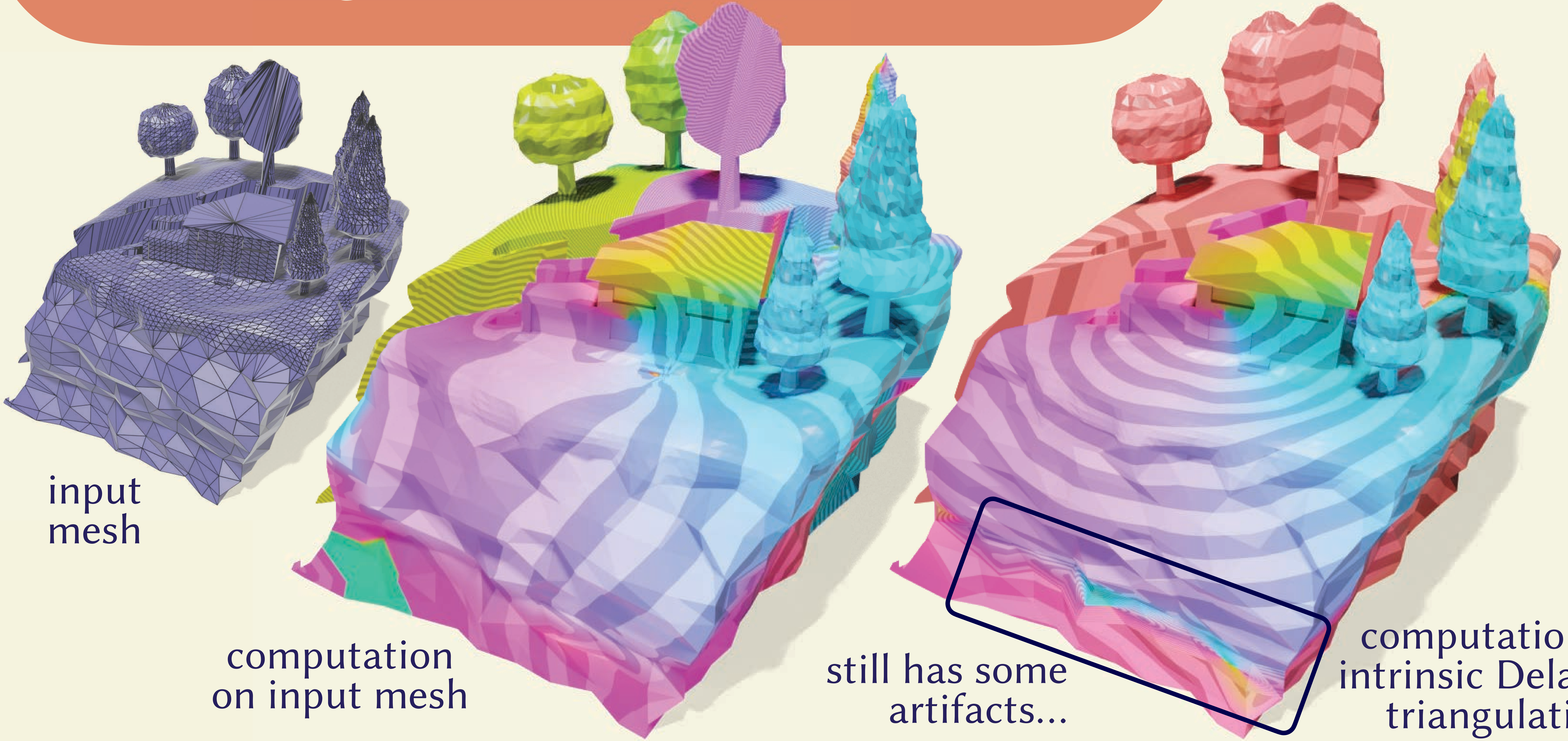
- [Indermitte, Liebling, Troyanov & Clemençon 2001, Bobenko & Springborn 2007]: empty intrinsic circumcircles
  - Maintain most nice properties. [Sharp, **G.** & Crane 2021; §4.1.1]
- Compute by a simple algorithm:
  - Flip any non-Delaunay edge until none remain



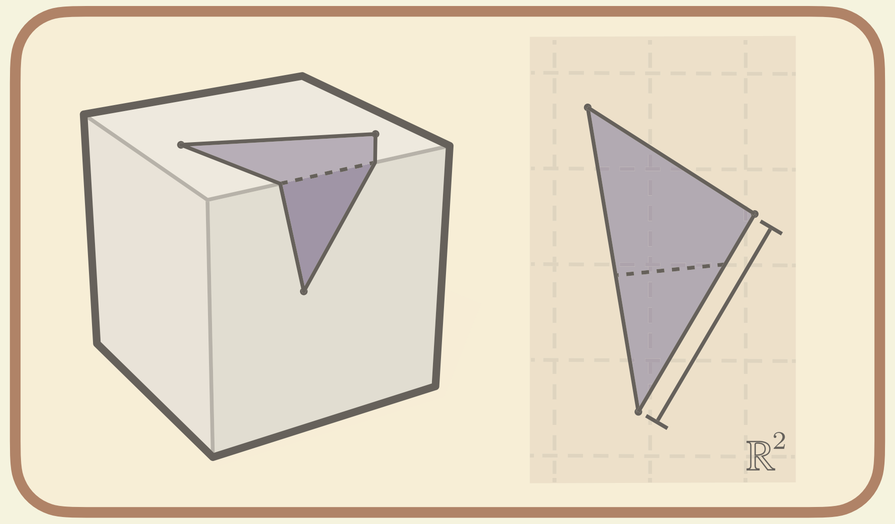


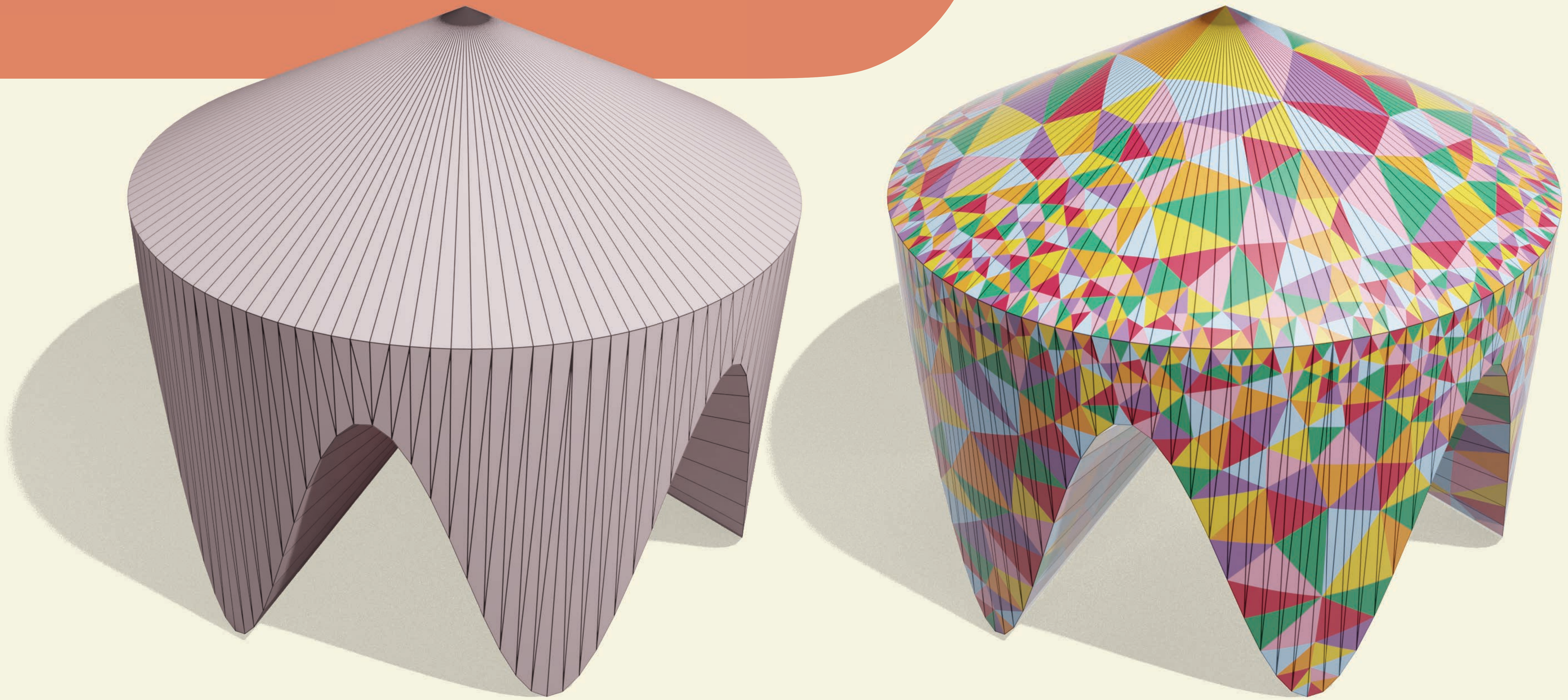


Intrinsic Delaunay triangulations provide good function spaces



A brief history of intrinsic triangulations

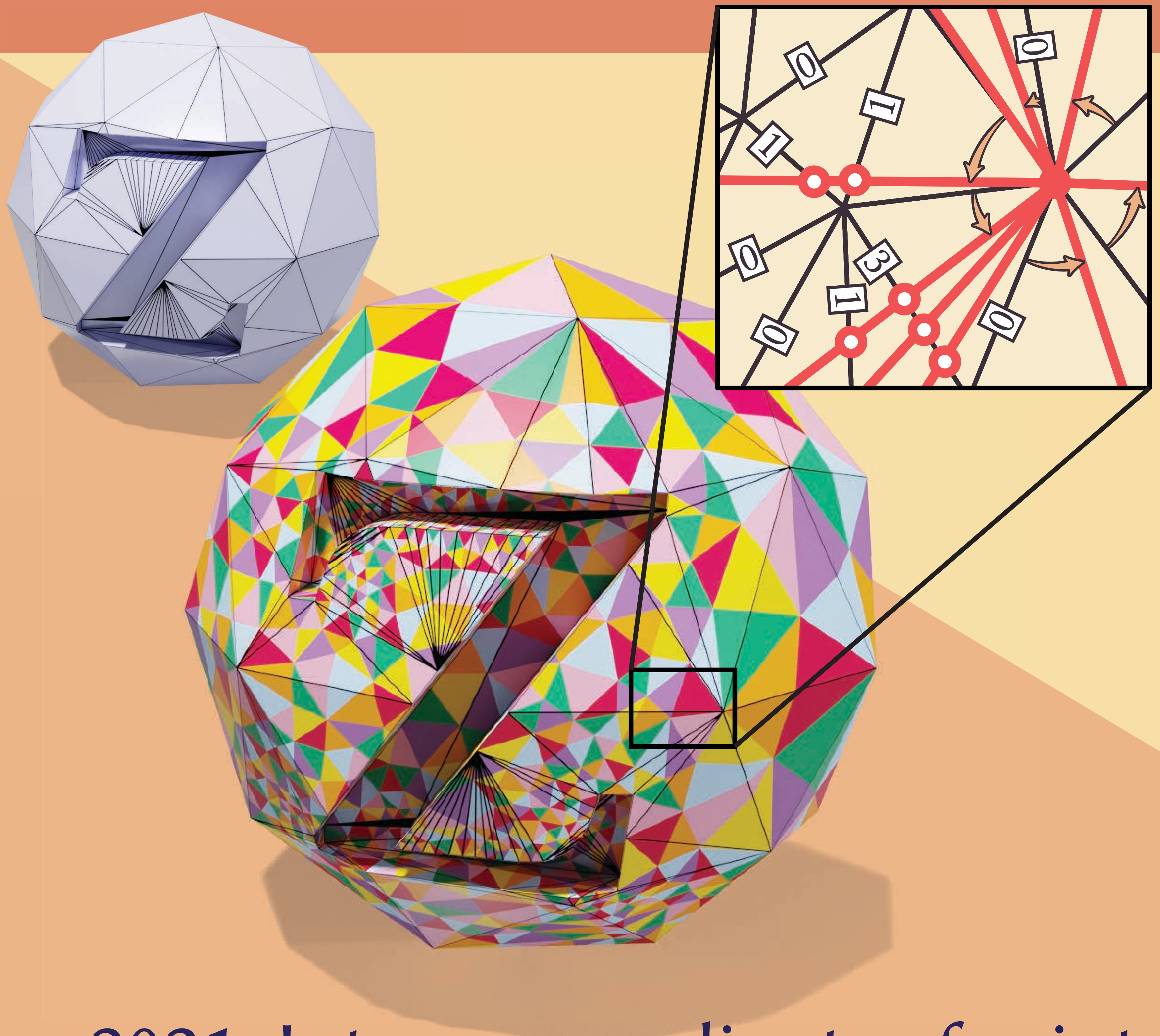




Foundations: [Alexandrov 1948; Regge 1961]

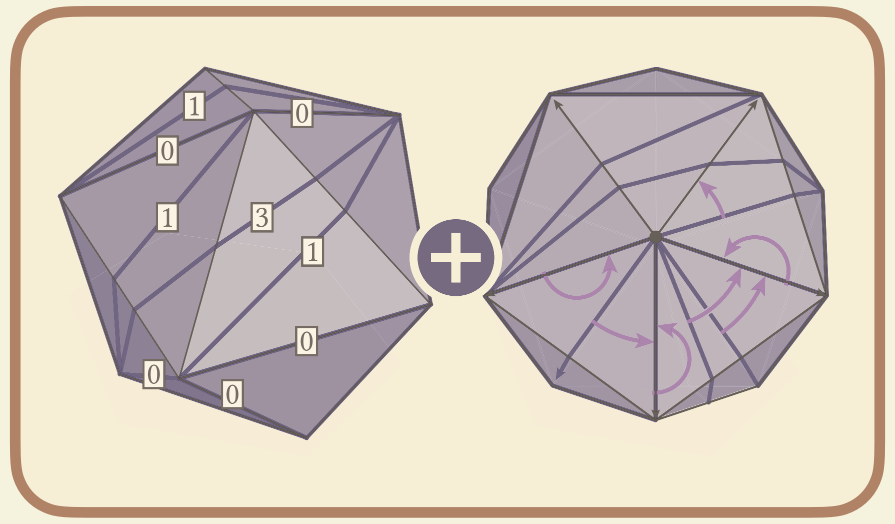
Geometry Processing: [Fisher, Springborn, Bobenko & Schröder 2006; Bobenko & Springborn 2007, Bobenko & Izmestiev 2008; Sun, Wu, Gu & Luo 2015; Sharp, Soliman & Crane 2019; Fumero, Möller & Rodolà 2020; Gillespie, Springborn & Crane 2021; Finnendahl, Schwartz & Alexa 2023]

II. Integer Coordinates for Intrinsic Triangulations



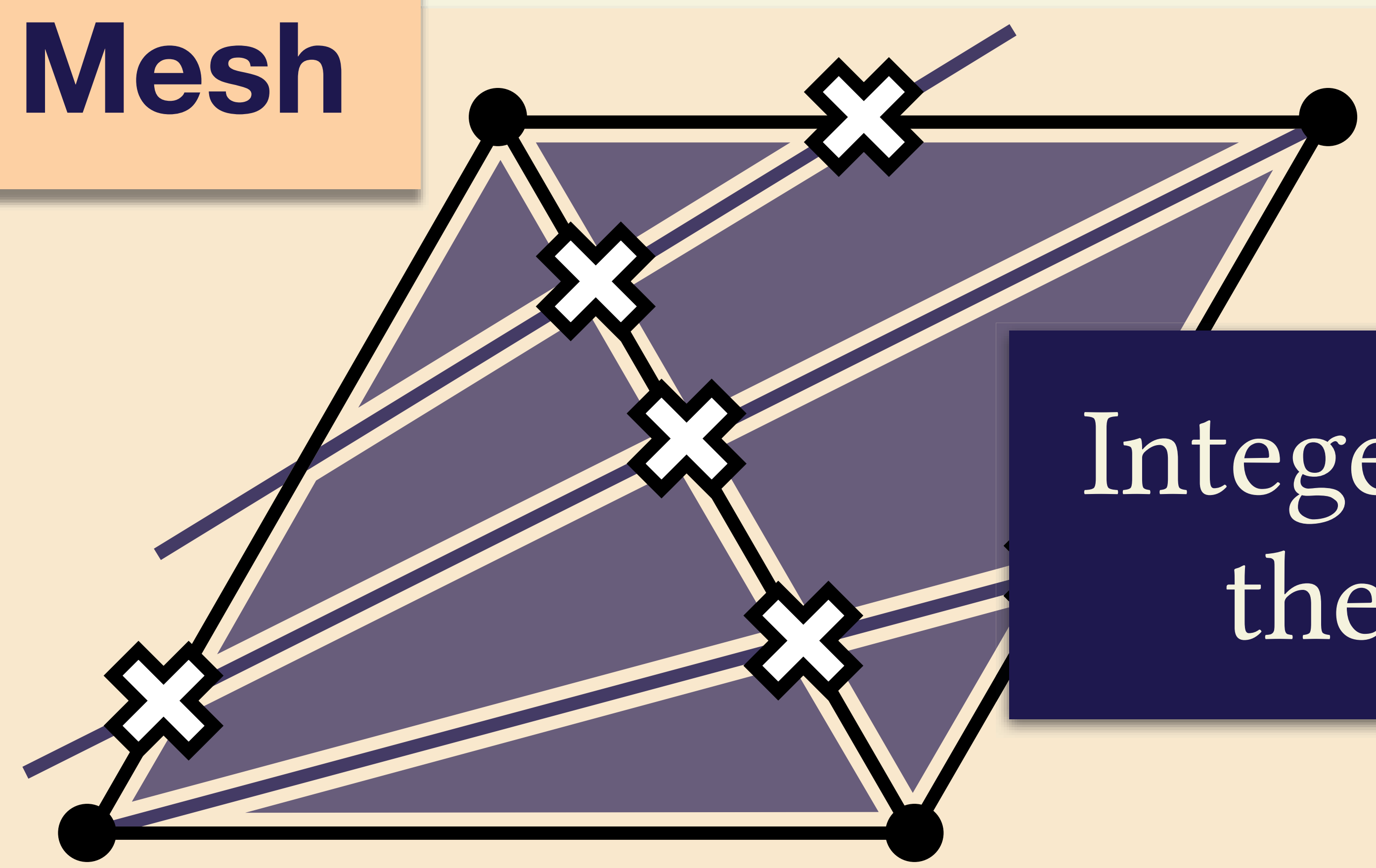
[G., Sharp, & Crane. 2021. Integer coordinates for intrinsic geometry processing. *ACM Transactions on Graphics*]

Correspondence data structures



II. Integer coordinates for intrinsic triangulations

Overlay Mesh

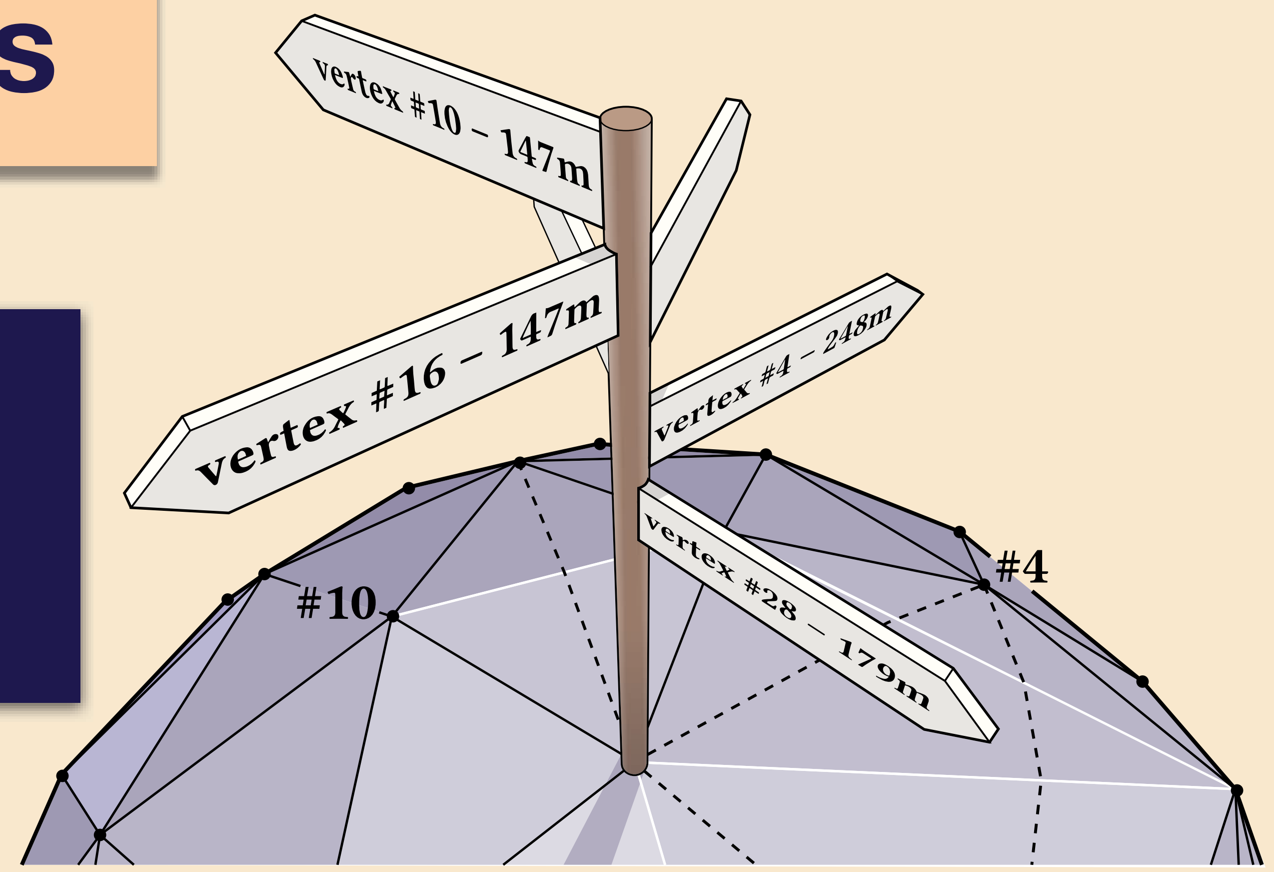


[Fisher, Springborn, Bobenko & Schröder 2006]

- Explicit mesh of common subdivision
- Edge flips nonlocal & expensive
    - **No further operations**

Integer coordinates combine the best of both worlds

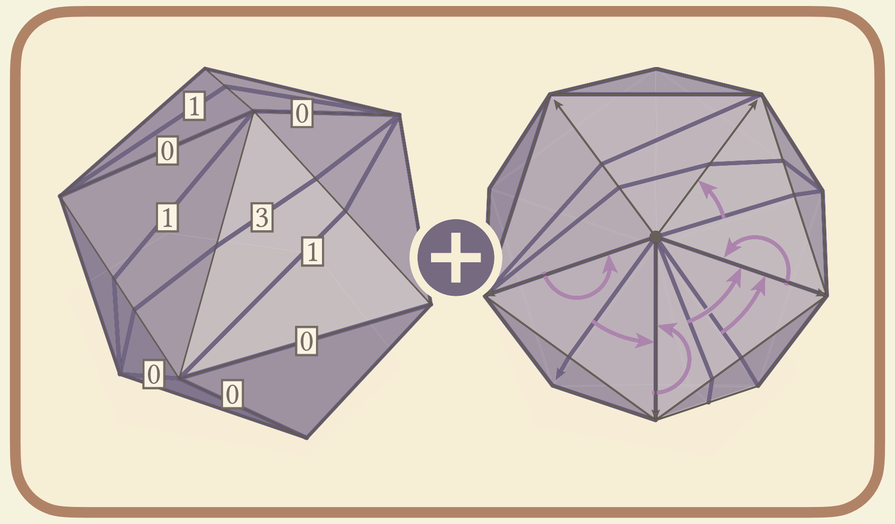
Signposts



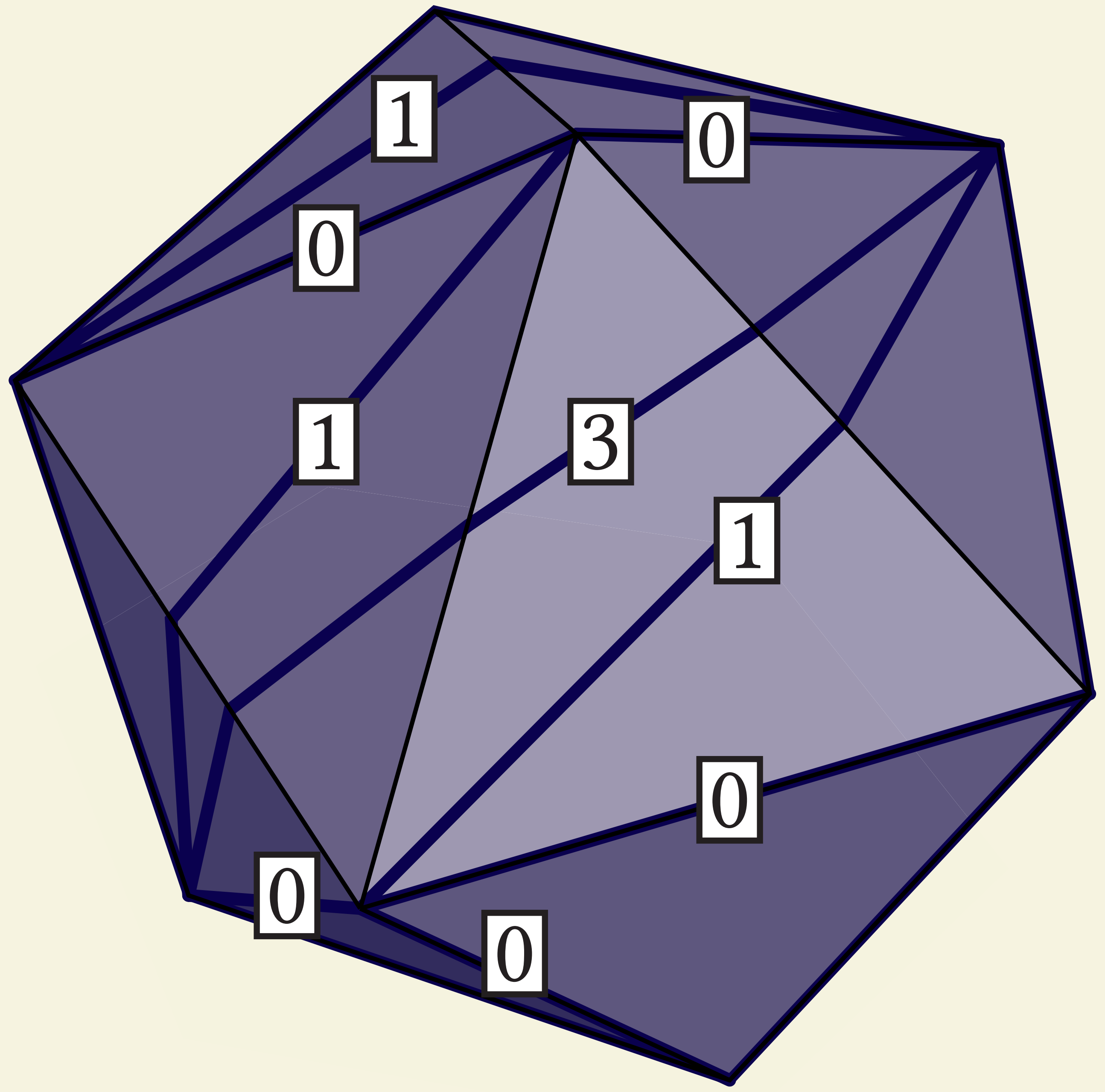
[Sharp, Soliman & Crane 2019]

- Floating point *signpost* vectors at vertices
- **Supports many local mesh operations**
- Common subdivision connectivity may be invalid

The integer coordinates data structure

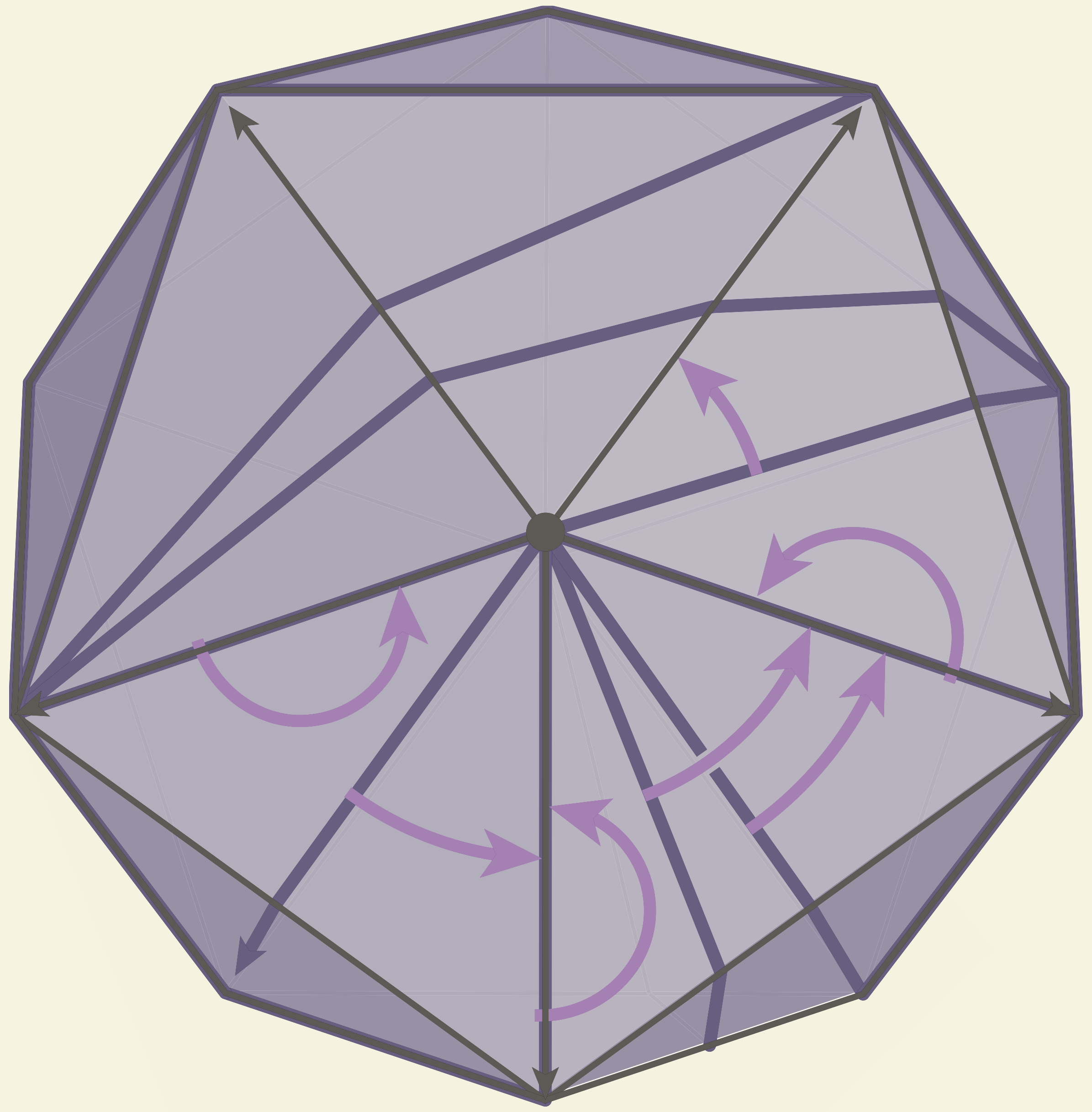


II. Integer coordinates for intrinsic triangulations



normal coordinates

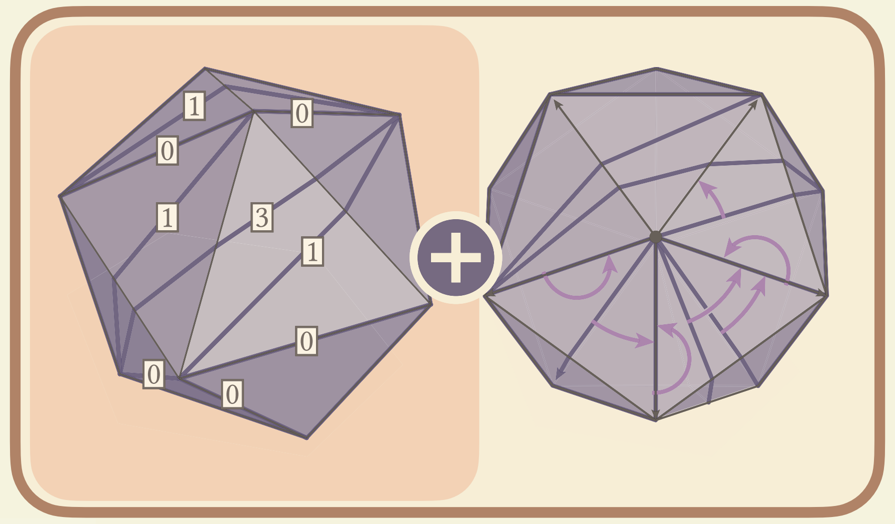
+

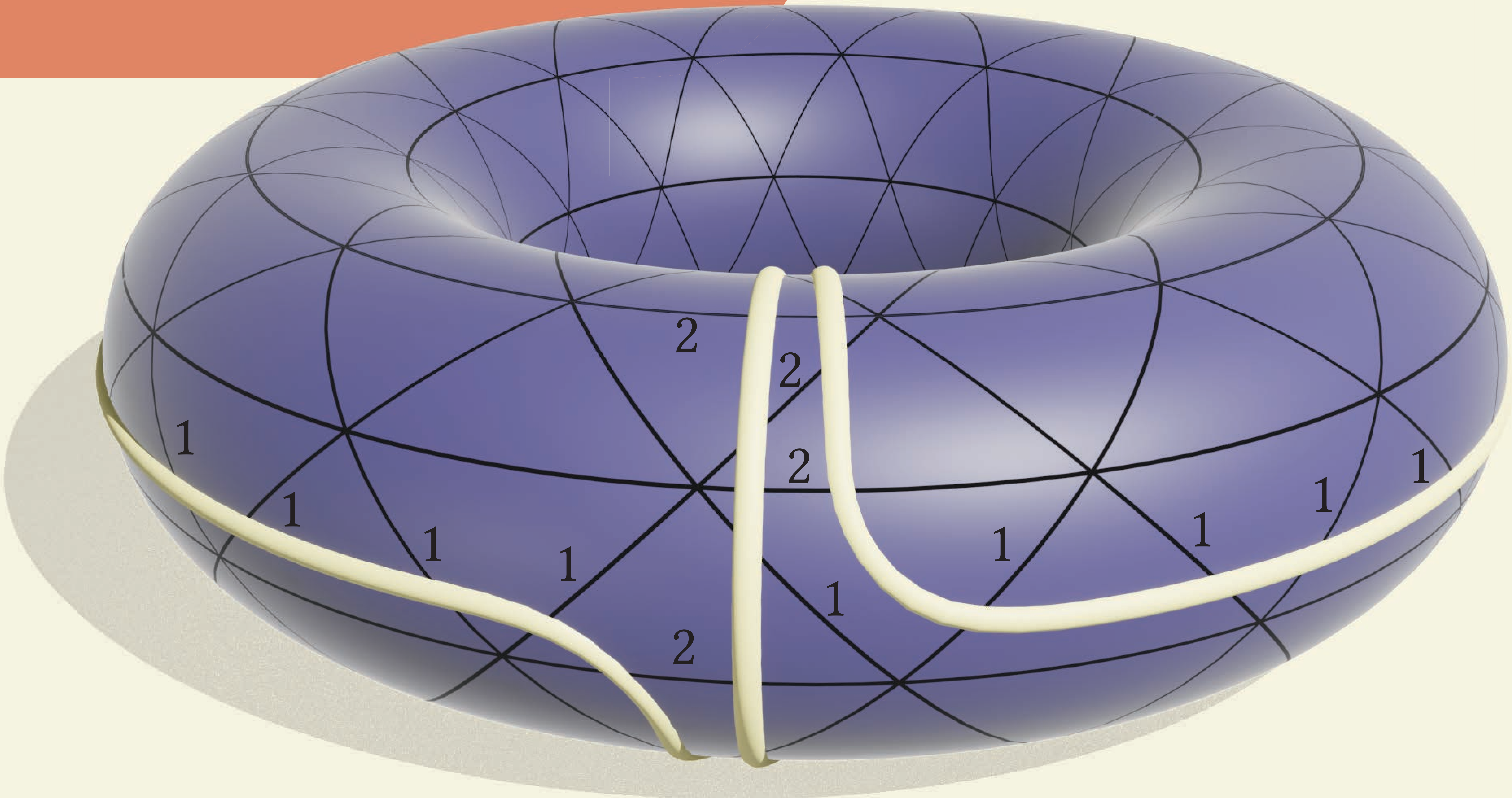


roundabouts

(concretely, just 3 integers per mesh edge)

Normal coordinates





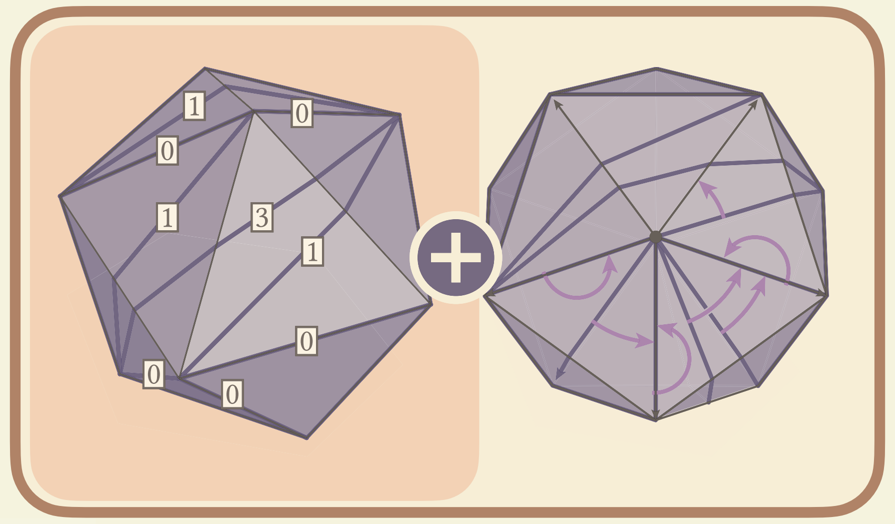
II. Integer coordinates for intrinsic triangulations

Foundations: [Kneser 1929; Haken 1961]

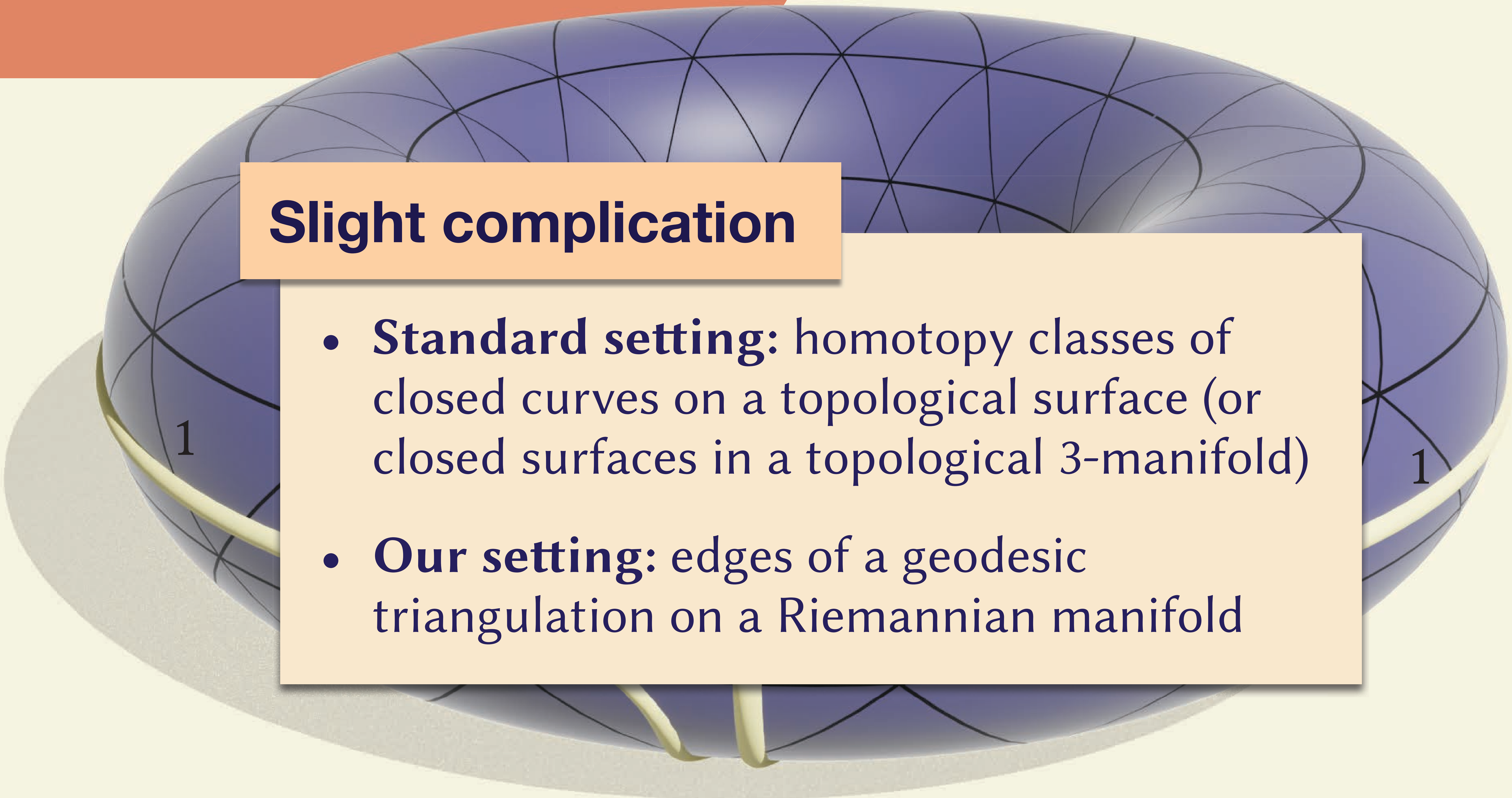
Computational Topology: [Schaefer+ 2008; Erickson & Nayyeri 2013]

Geometry Processing: [Hass & Trnkova 2020]

Normal coordinates



II. Integer coordinates for intrinsic triangulations



Slight complication

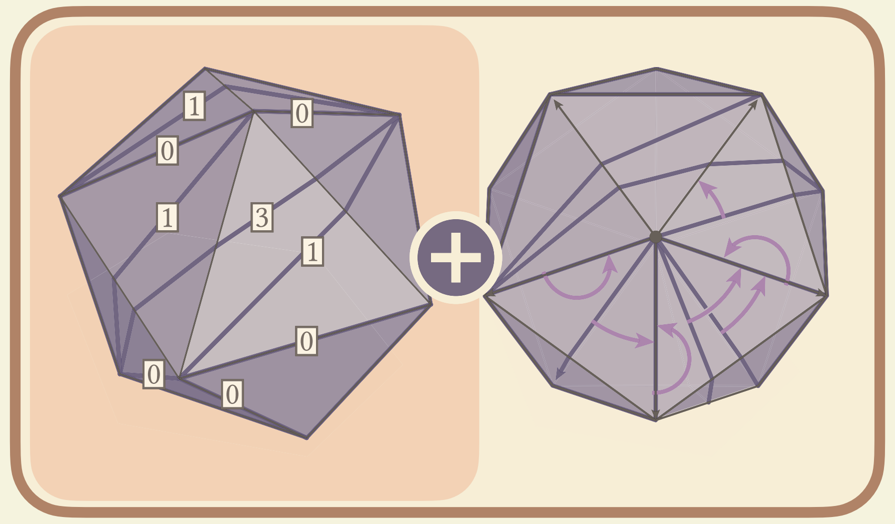
- **Standard setting:** homotopy classes of closed curves on a topological surface (or closed surfaces in a topological 3-manifold)
- **Our setting:** edges of a geodesic triangulation on a Riemannian manifold

Foundations: [Kneser 1929; Haken 1961]

Computational Topology: [Schaefer+ 2008; Erickson & Nayyeri 2013]

Geometry Processing: [Hass & Trnkova 2020]

Encoding a curve with normal coordinates

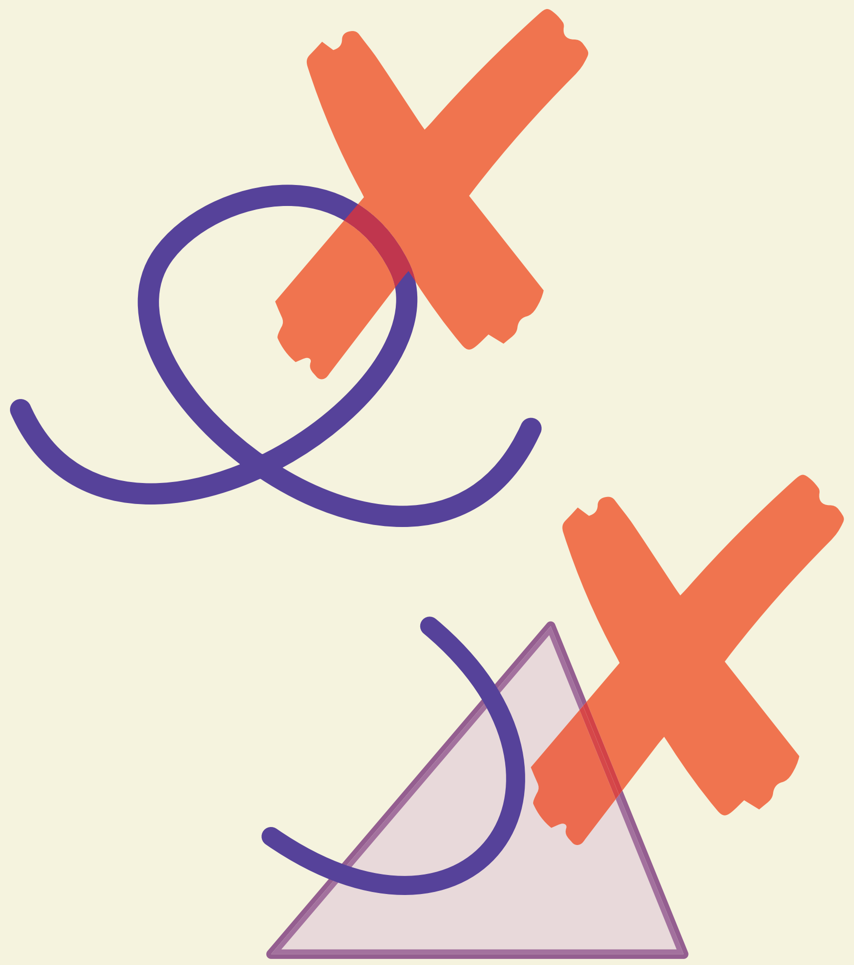


II. Integer coordinates for intrinsic triangulations

- Just count intersections

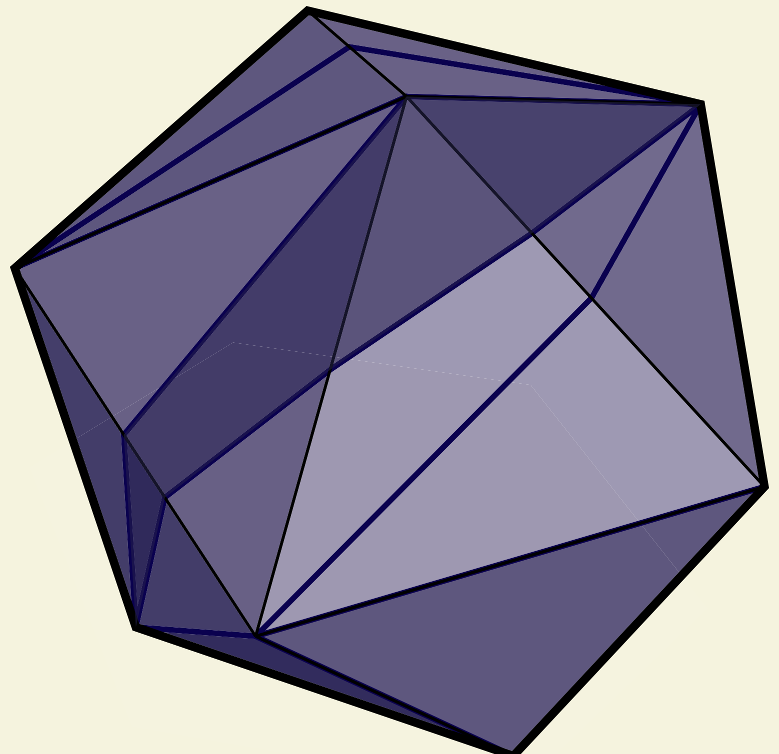
Rules

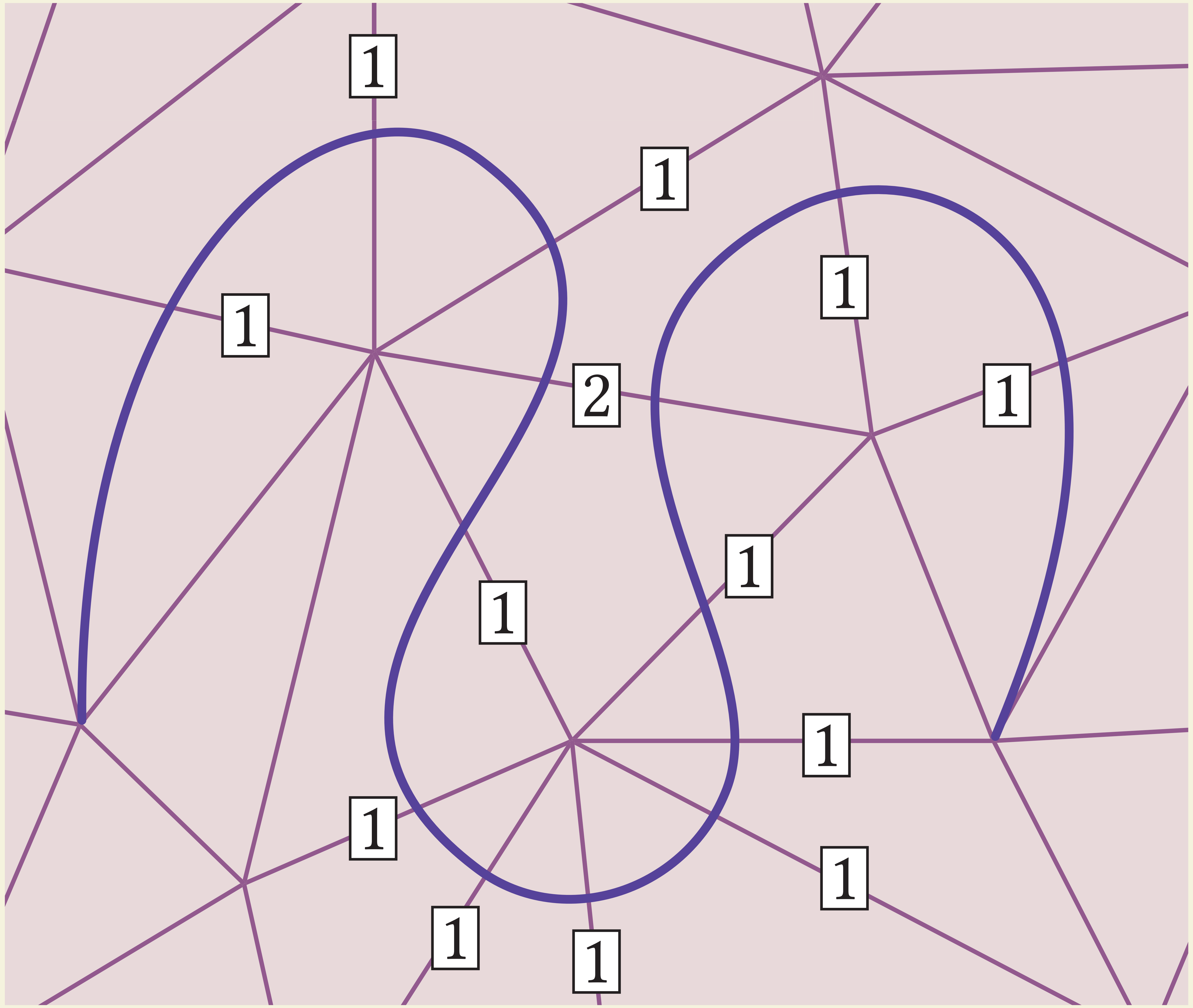
1. No self-crossings
2. No U-turns



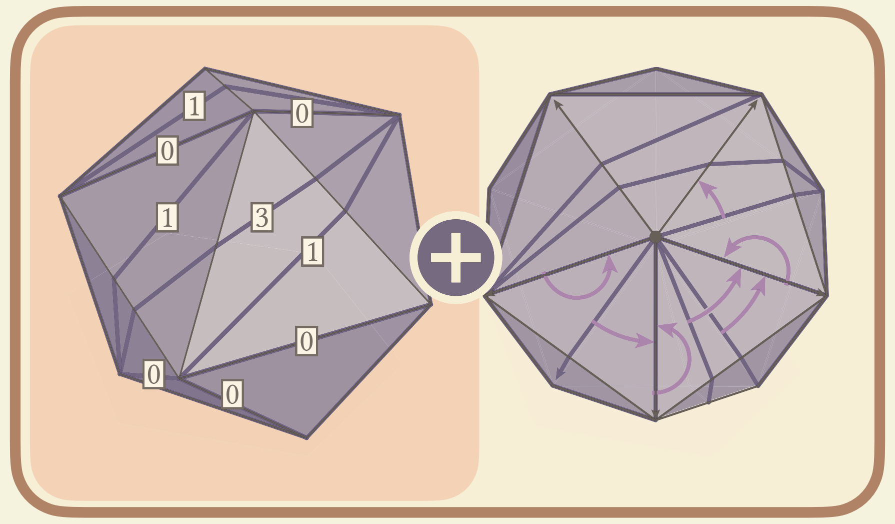
(also curves may only start or end at vertices of the triangulation)

automatically satisfied for geodesic triangulations





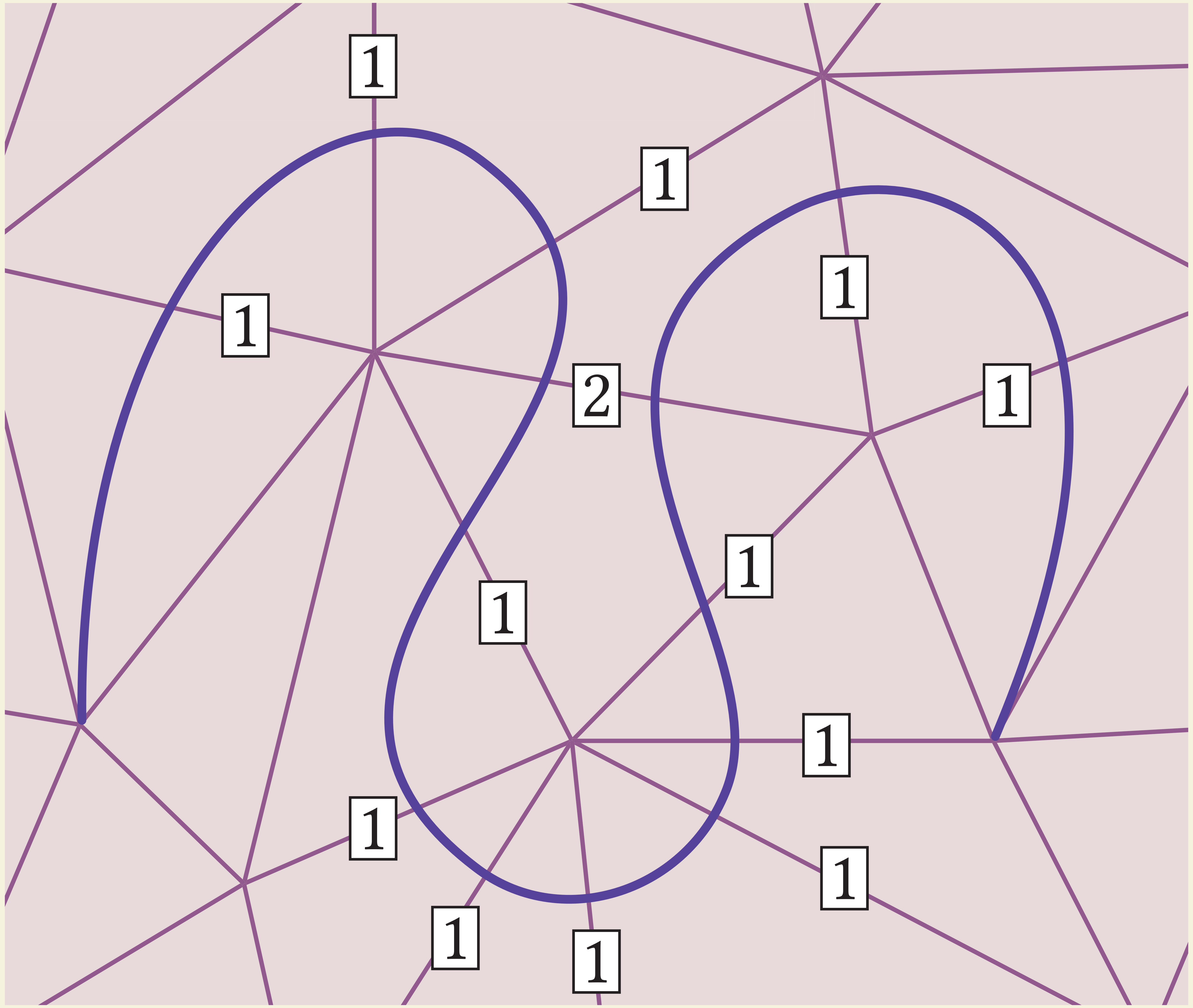
How much do normal coordinates tell us?



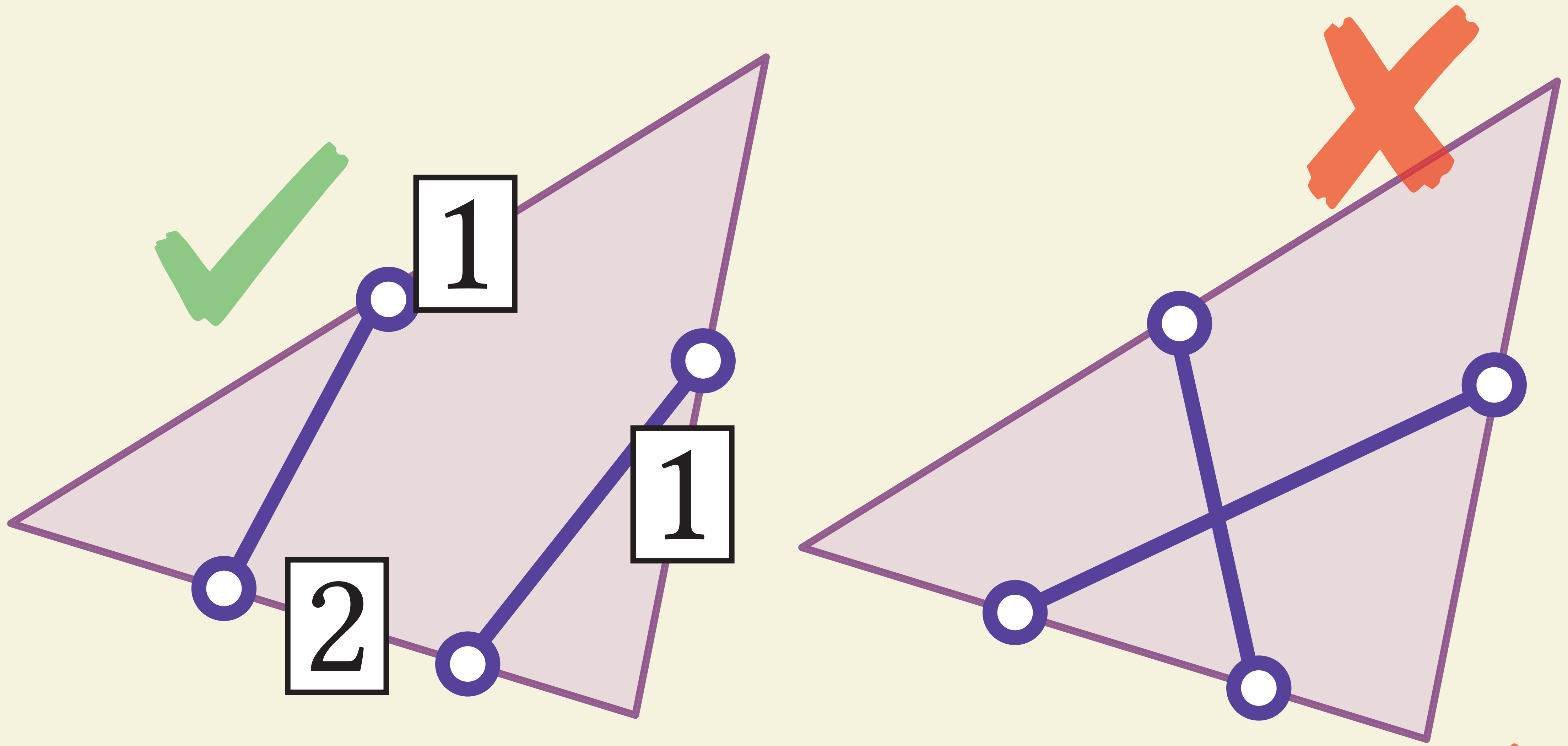
II. Integer coordinates for intrinsic triangulations

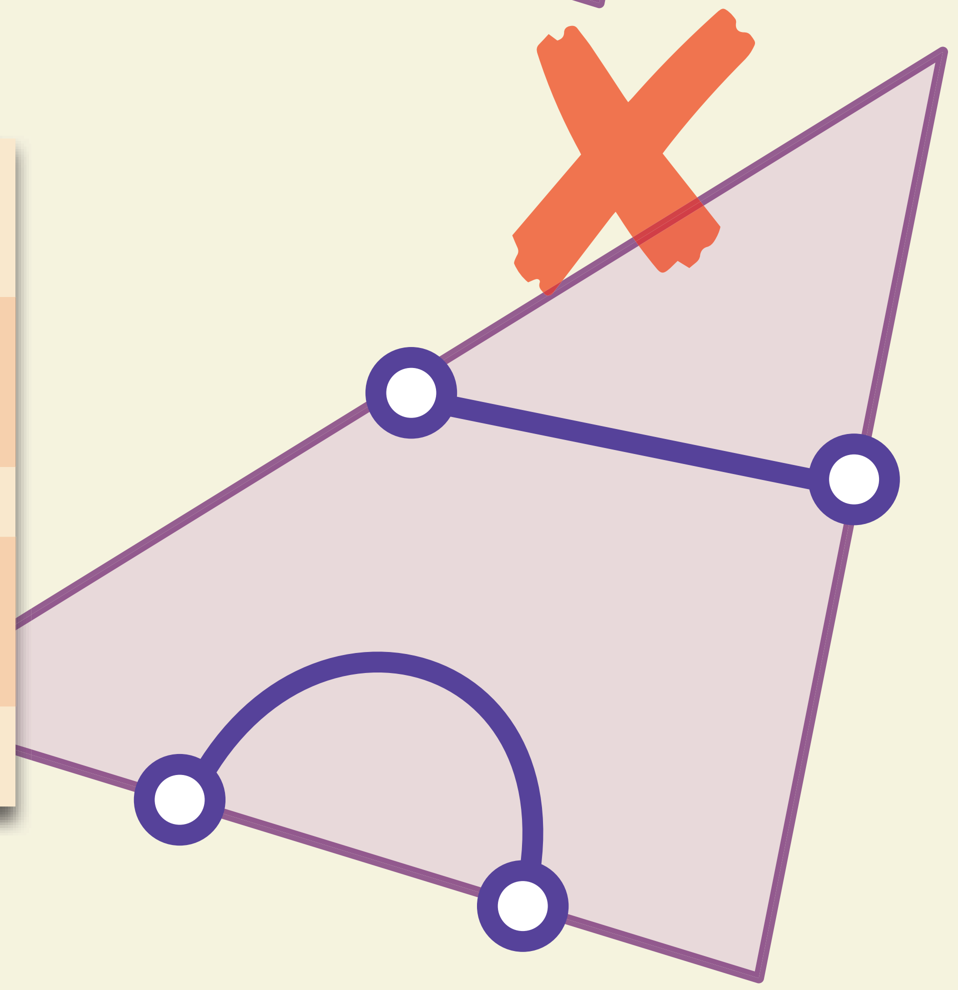
- Represents curve up to homotopy
  (on the surface punctured at vertices)

- Equivalently, encodes a sequence of triangles



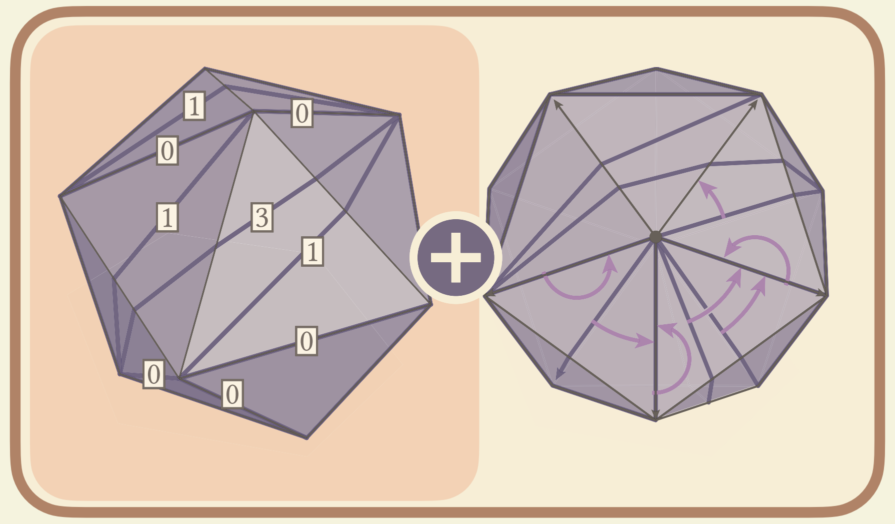
Reconstructing the curve



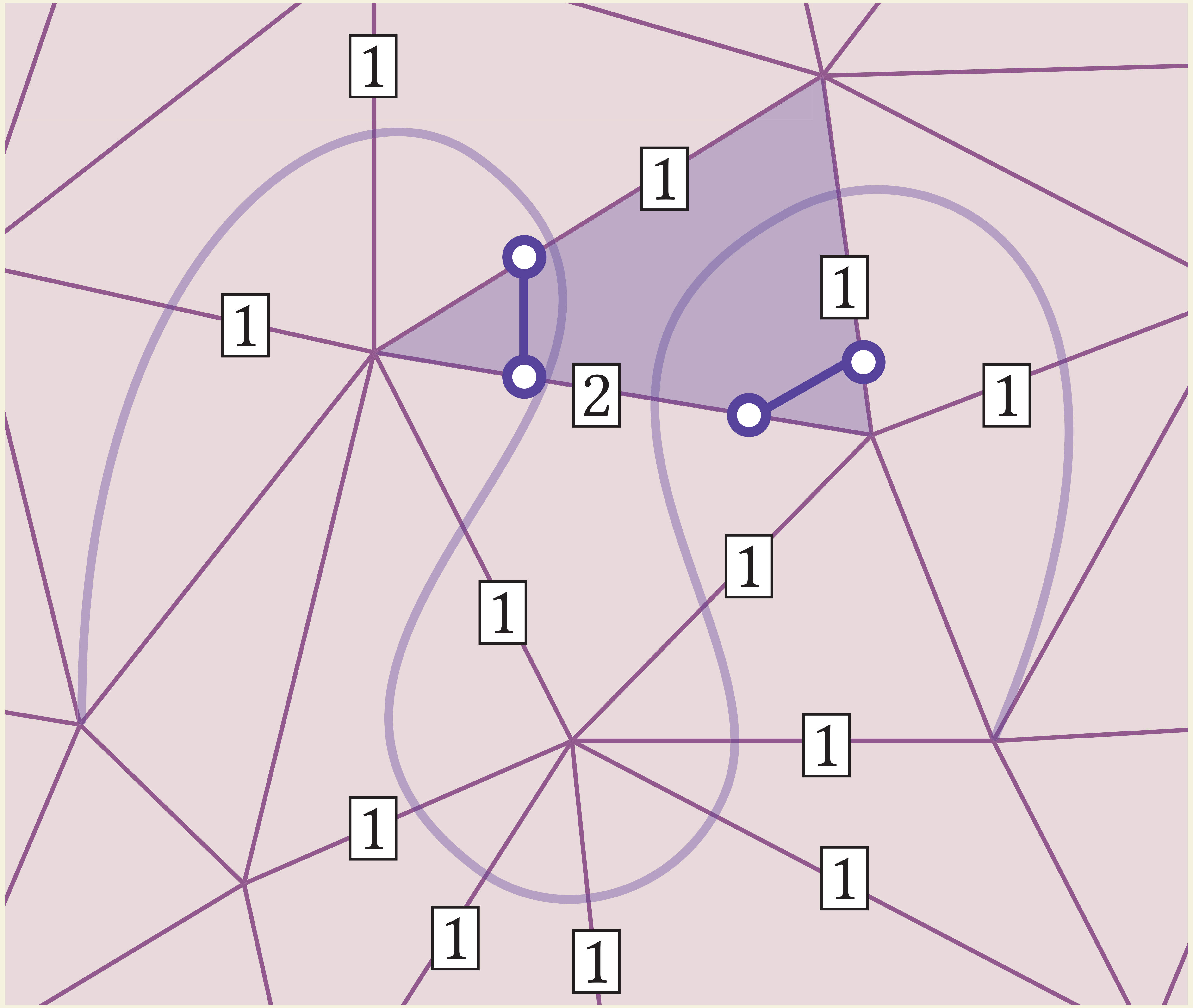


Rules

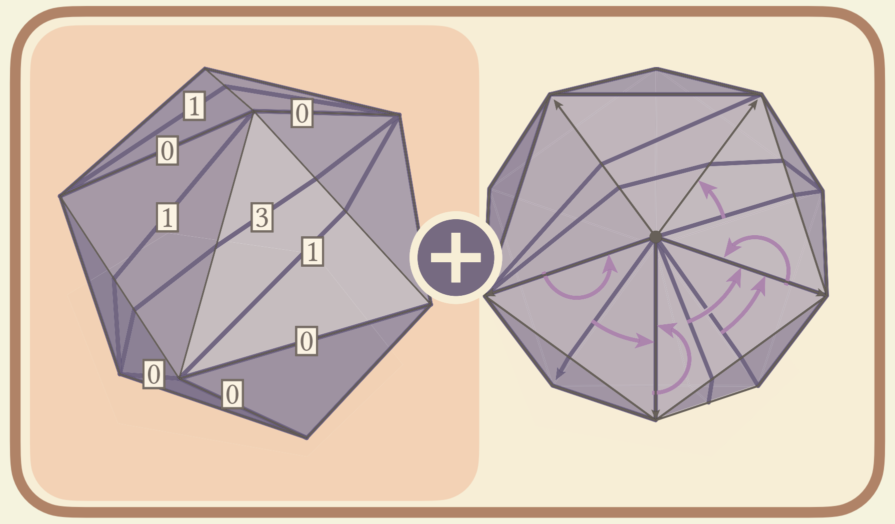
1. No self-crossings
2. No U-turns



II. Integer coordinates for intrinsic triangulations

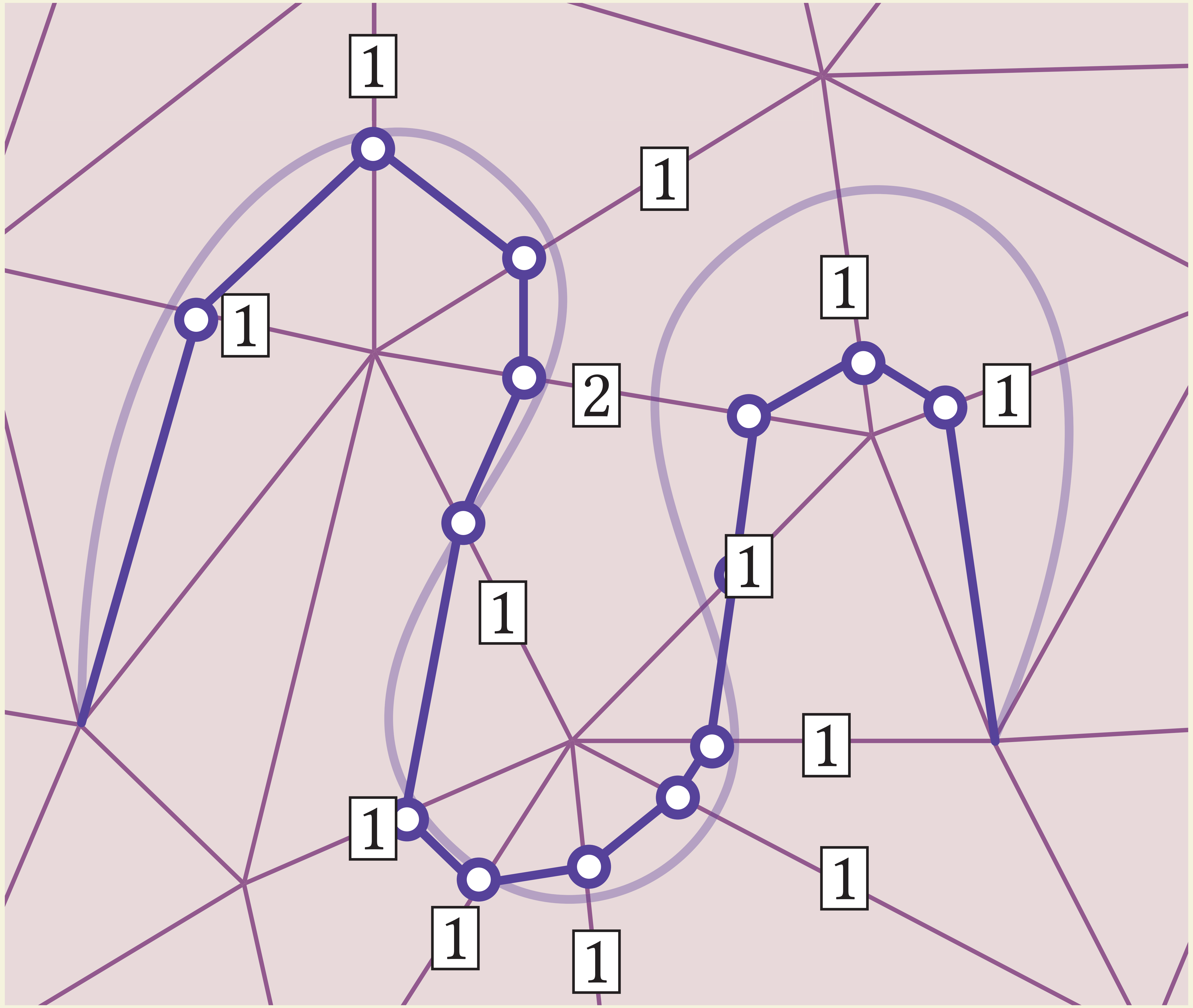


Reconstructing the curve

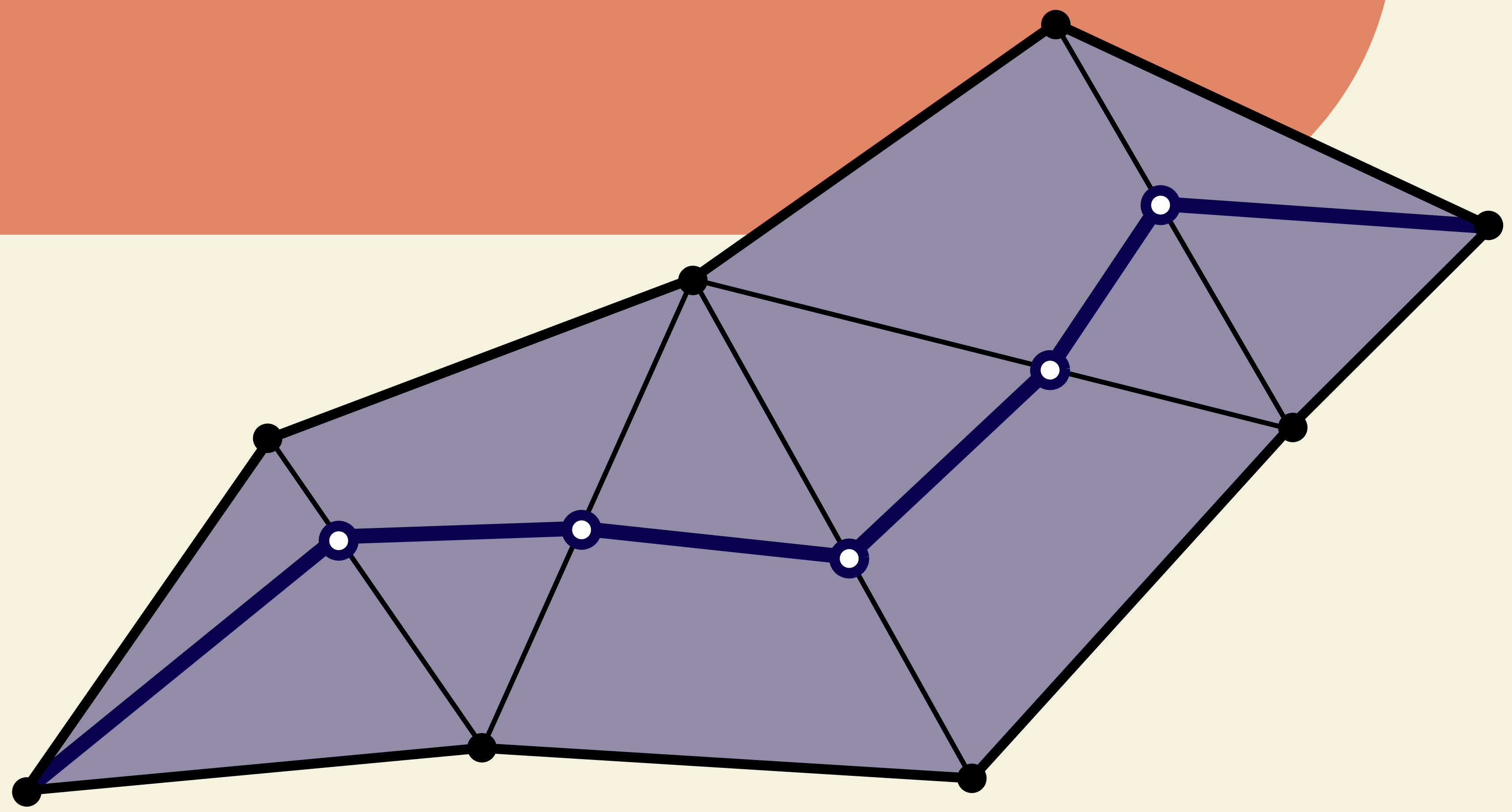


II. Integer coordinates for intrinsic triangulations

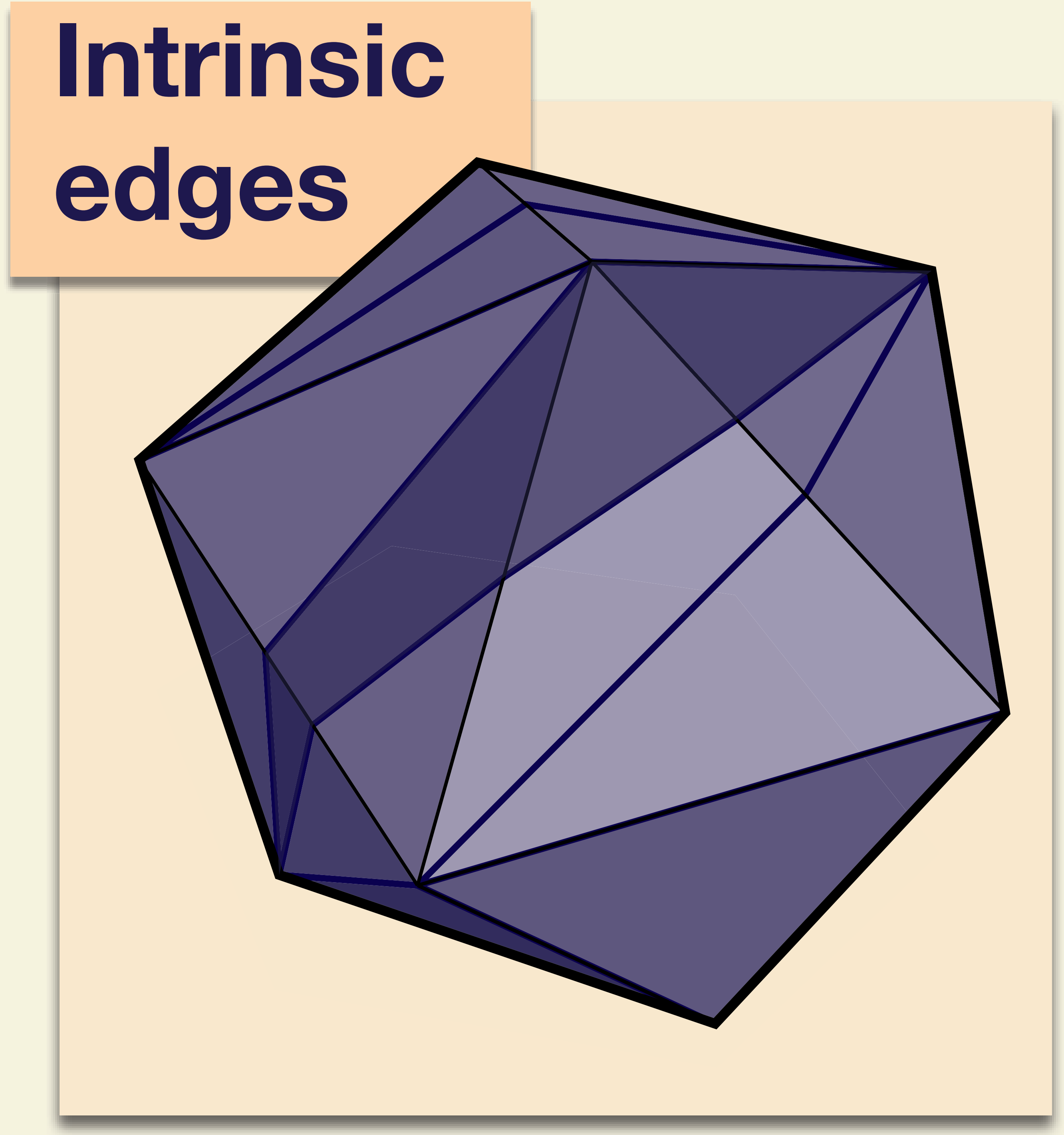
- Normality conditions determine curves within each triangle

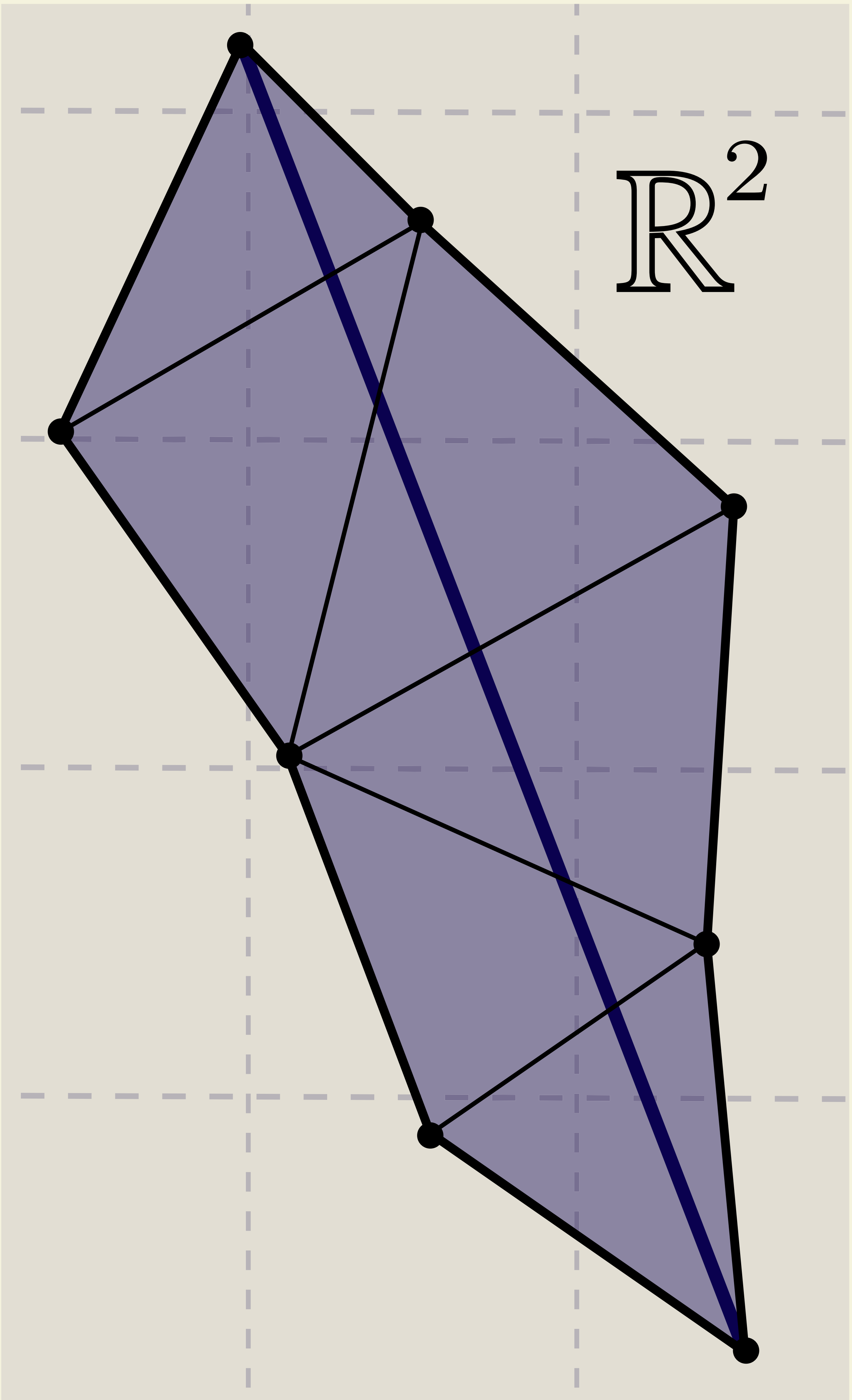


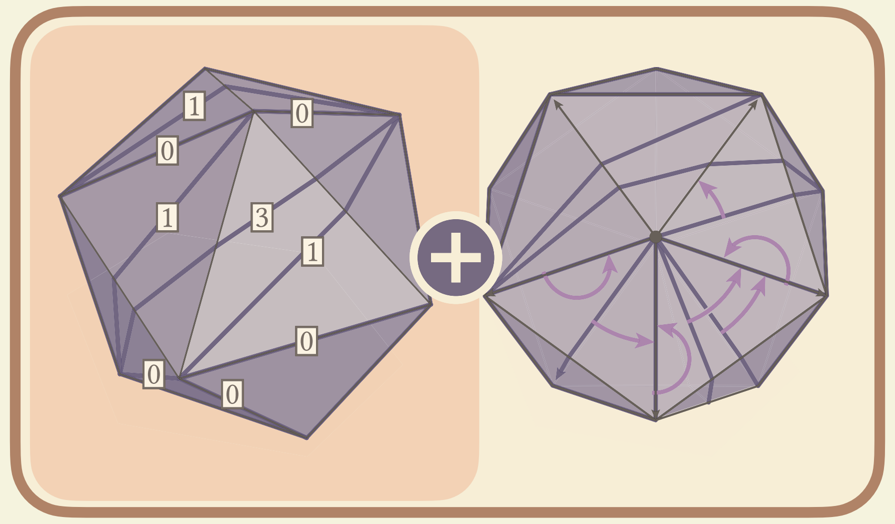
Finding the exact curve geometry



- So far: triangle strip
- True curve is *geodesic*
  - Lay out in plane to find exact curve
- Normal coordinates determine edges exactly

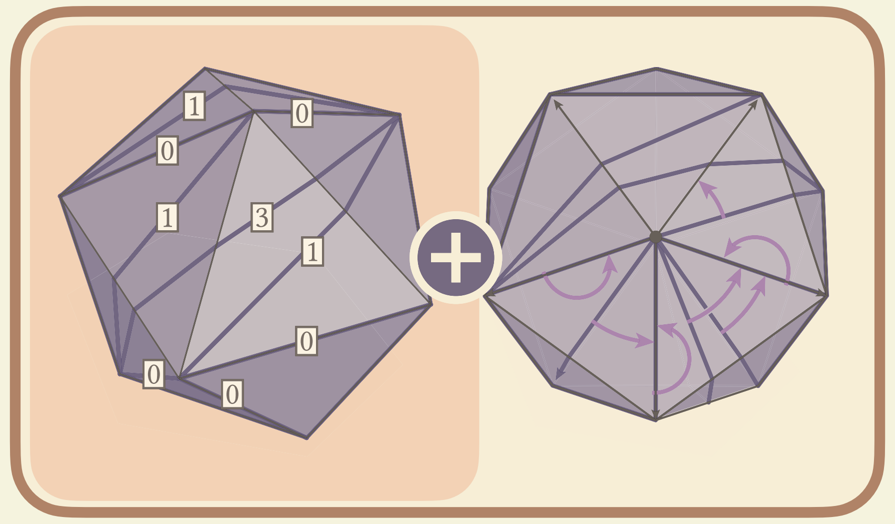






II. Integer coordinates for intrinsic triangulations

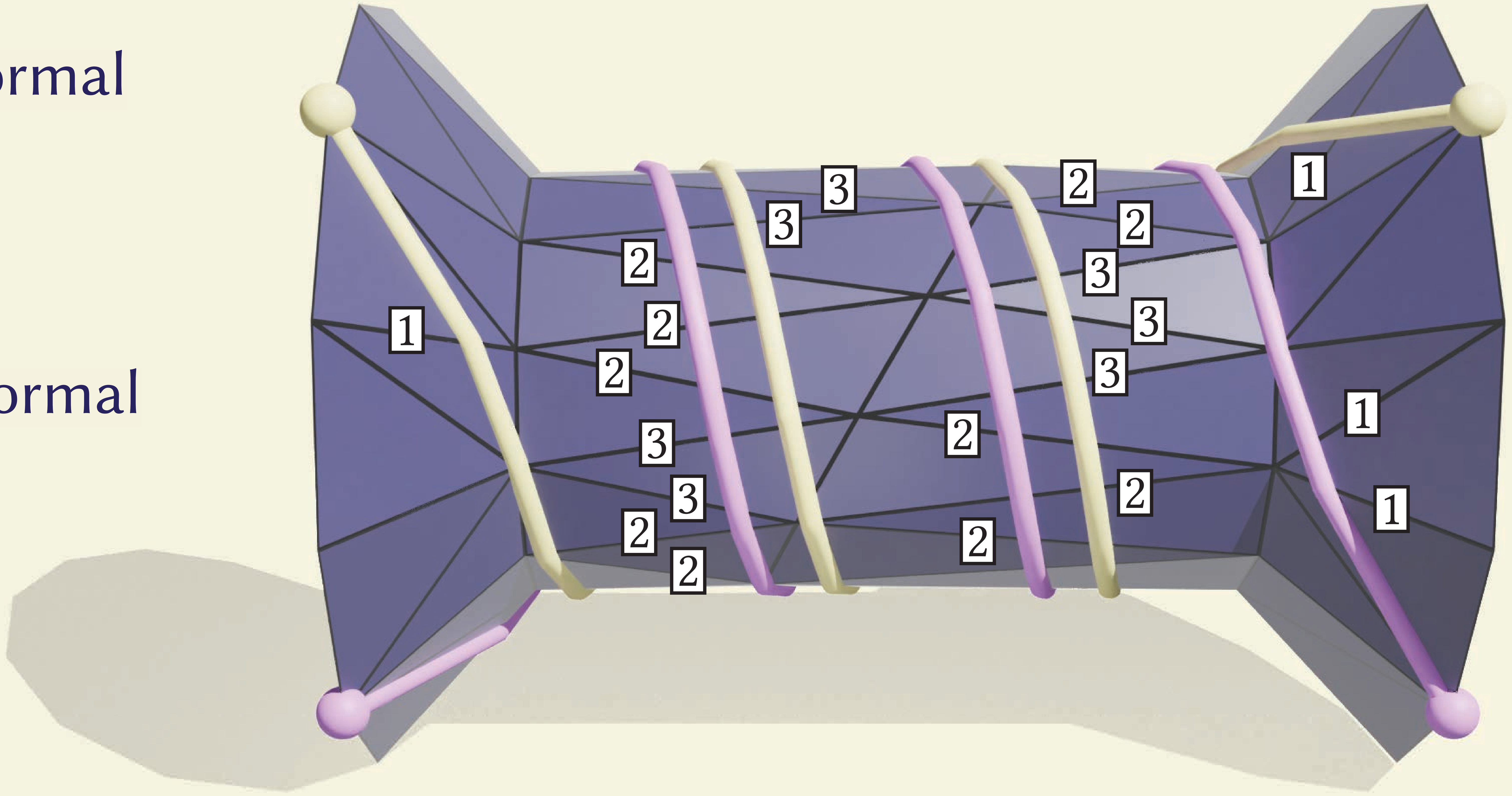
Collections of Curves



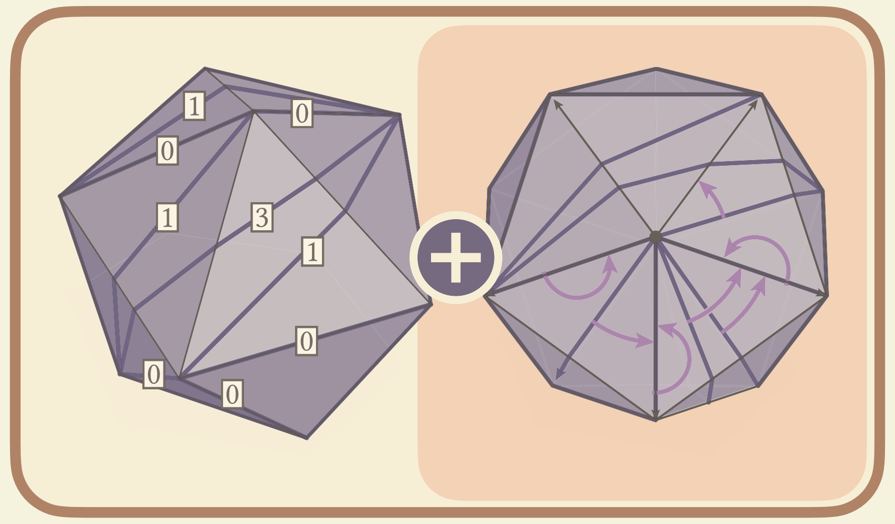
II. Integer coordinates for intrinsic triangulations

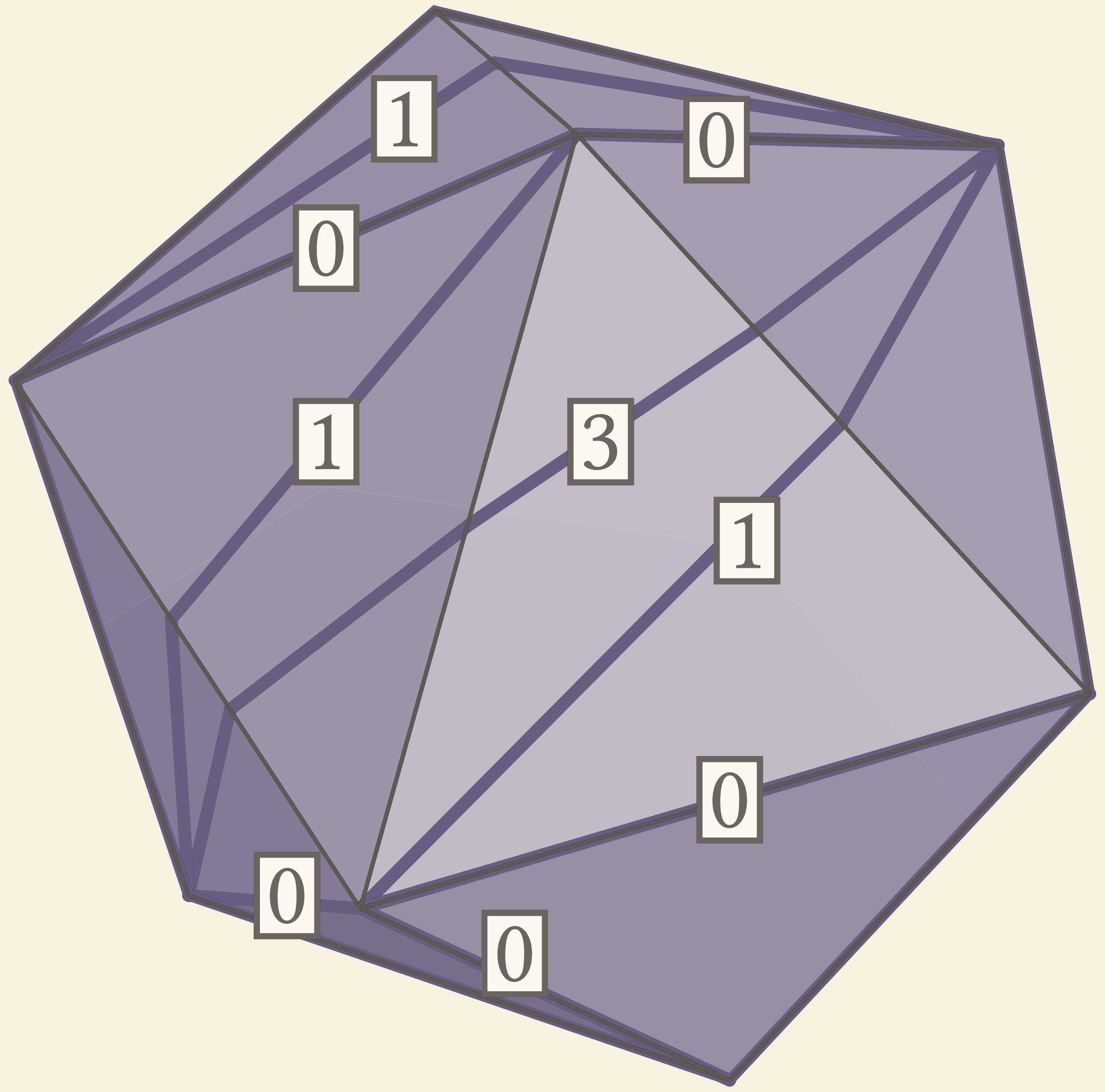
- *e.g.* edges of a triangulation
- Could store multiple sets of normal coordinates
  - Expensive ✗
- Instead, just store one set of normal coordinates

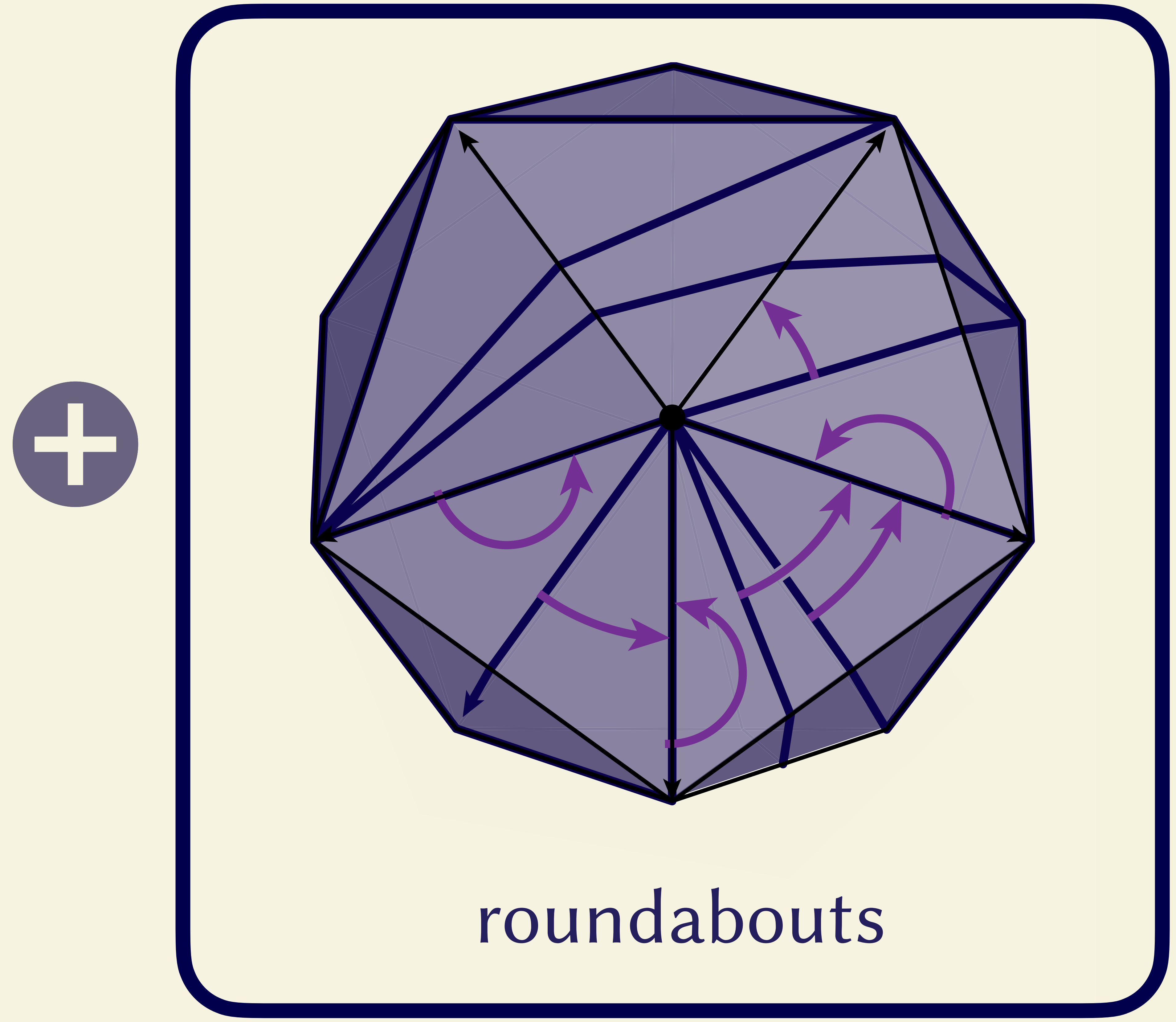
Store entire triangulation using one integer per edge



The integer coordinates data structure

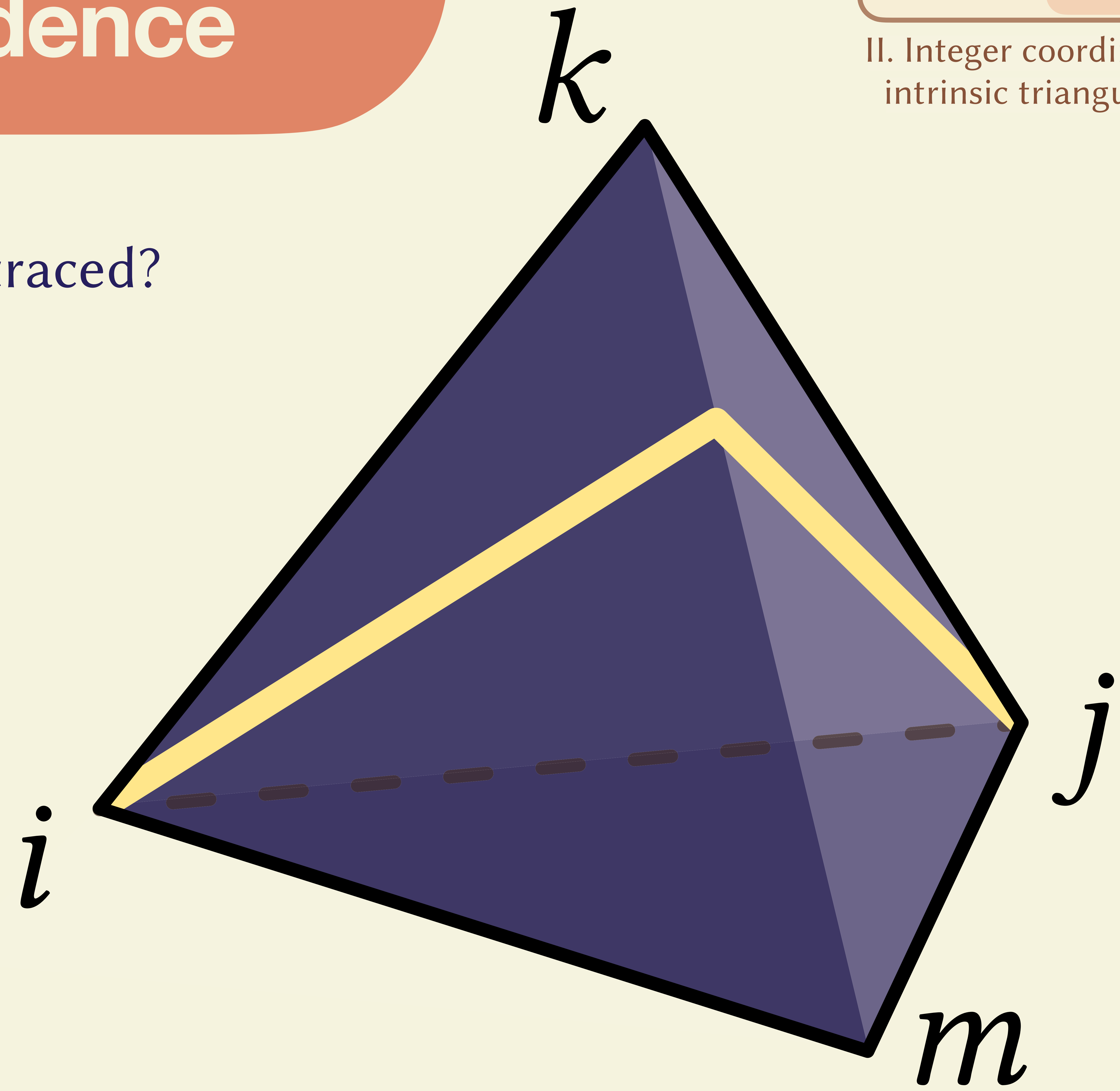


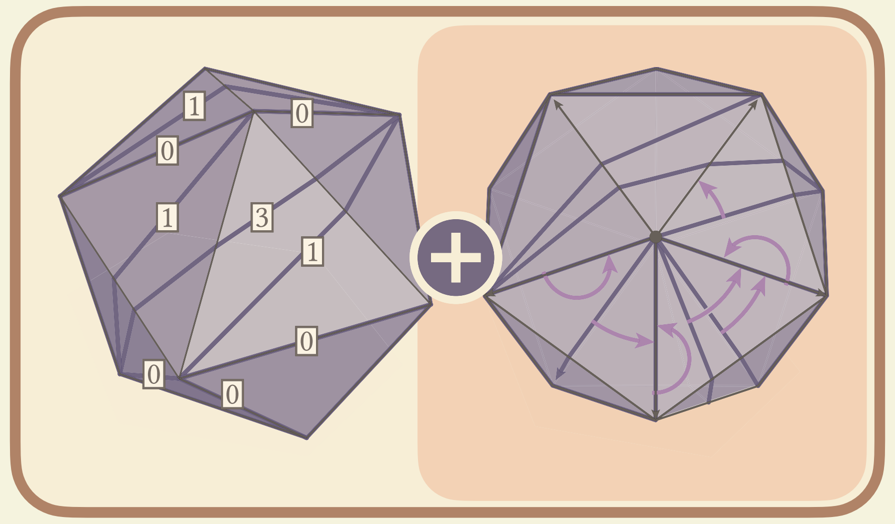




Normal coordinates are not enough for correspondence

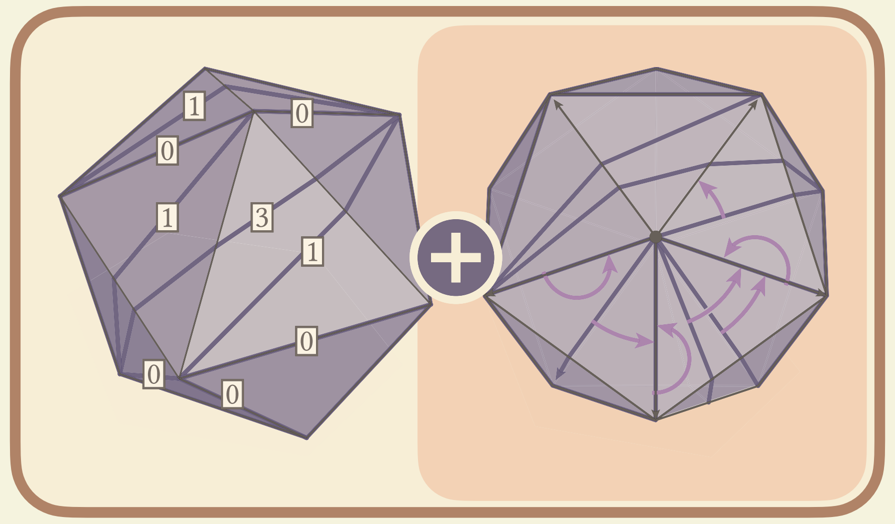
- How do you tell what edge you have traced?





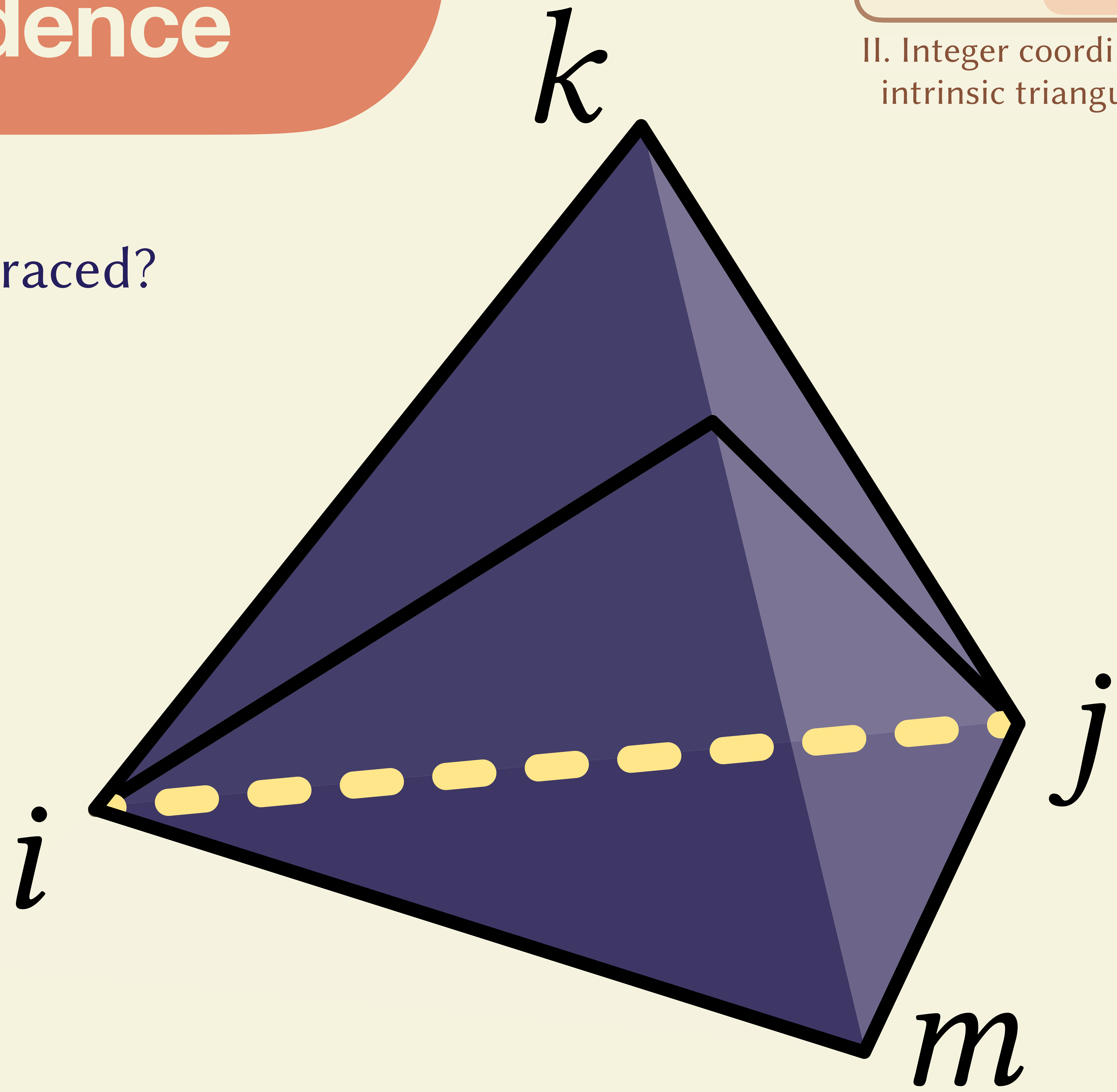
II. Integer coordinates for intrinsic triangulations

Normal coordinates are not enough for correspondence



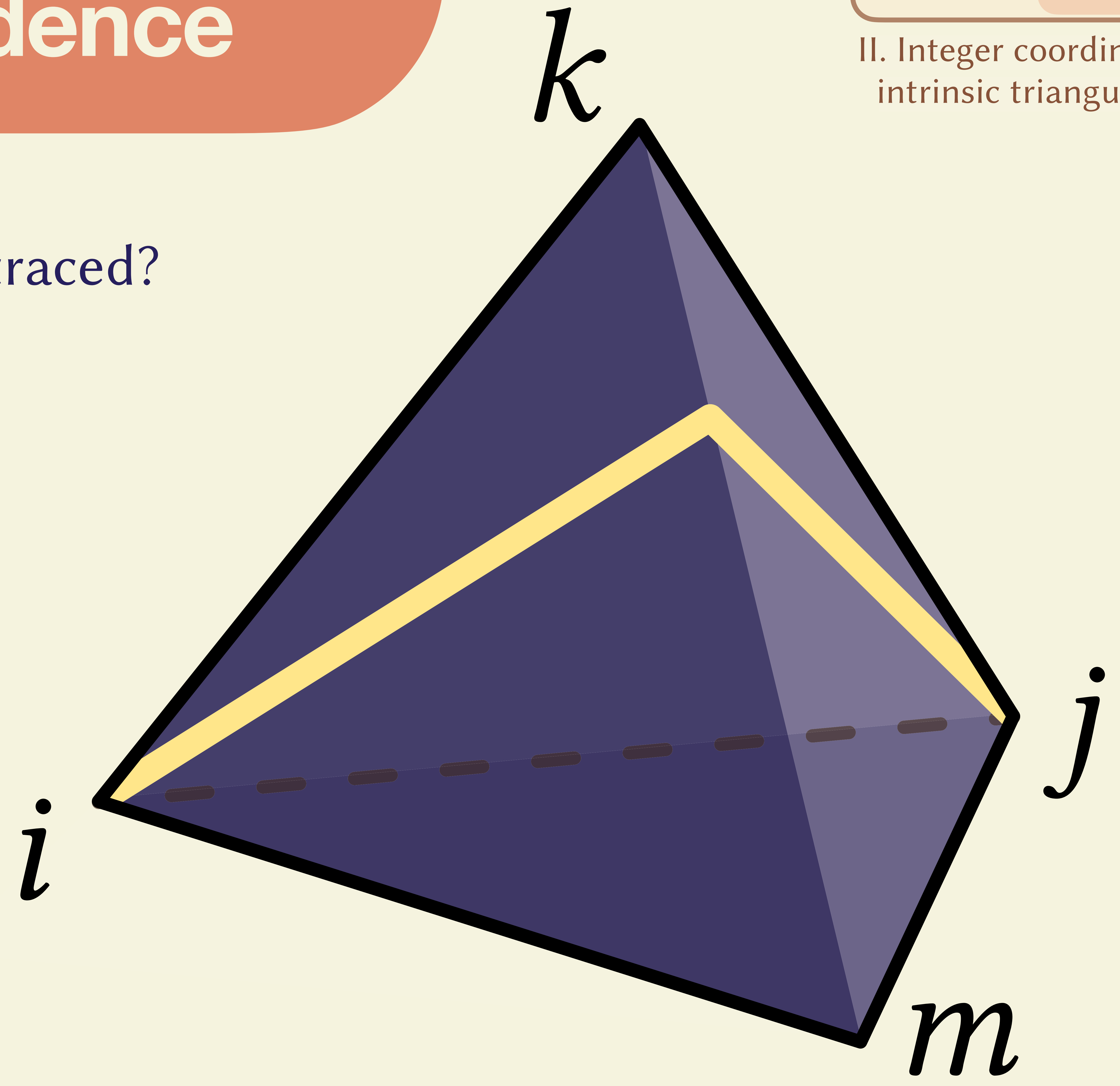
II. Integer coordinates for intrinsic triangulations

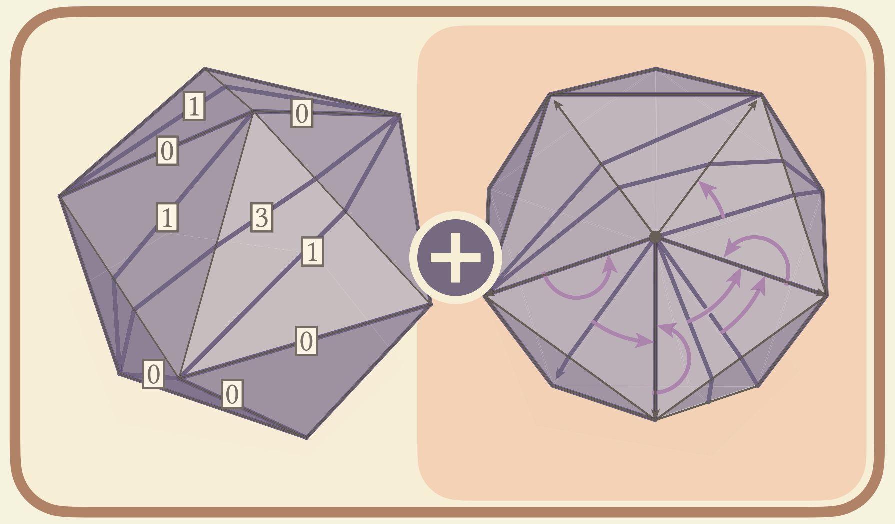
- How do you tell what edge you have traced?



Normal coordinates are not enough for correspondence

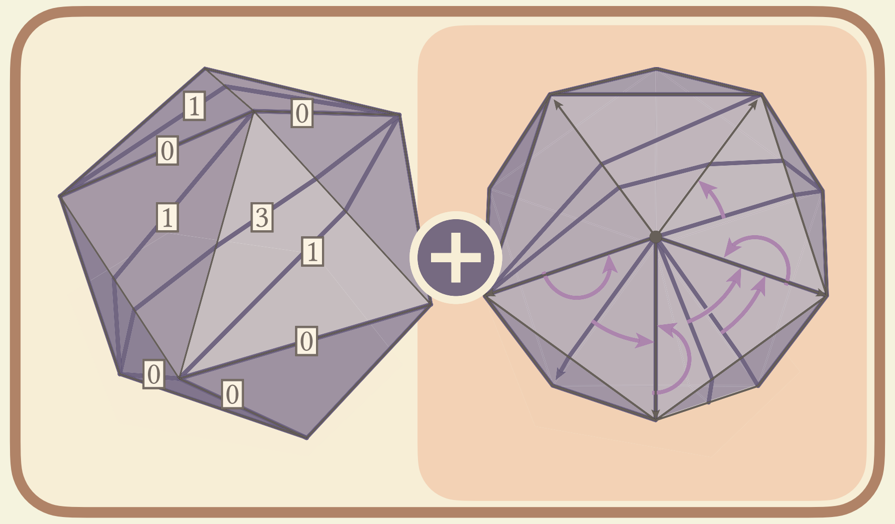
- How do you tell what edge you have traced?
  - Disambiguate with roundabouts





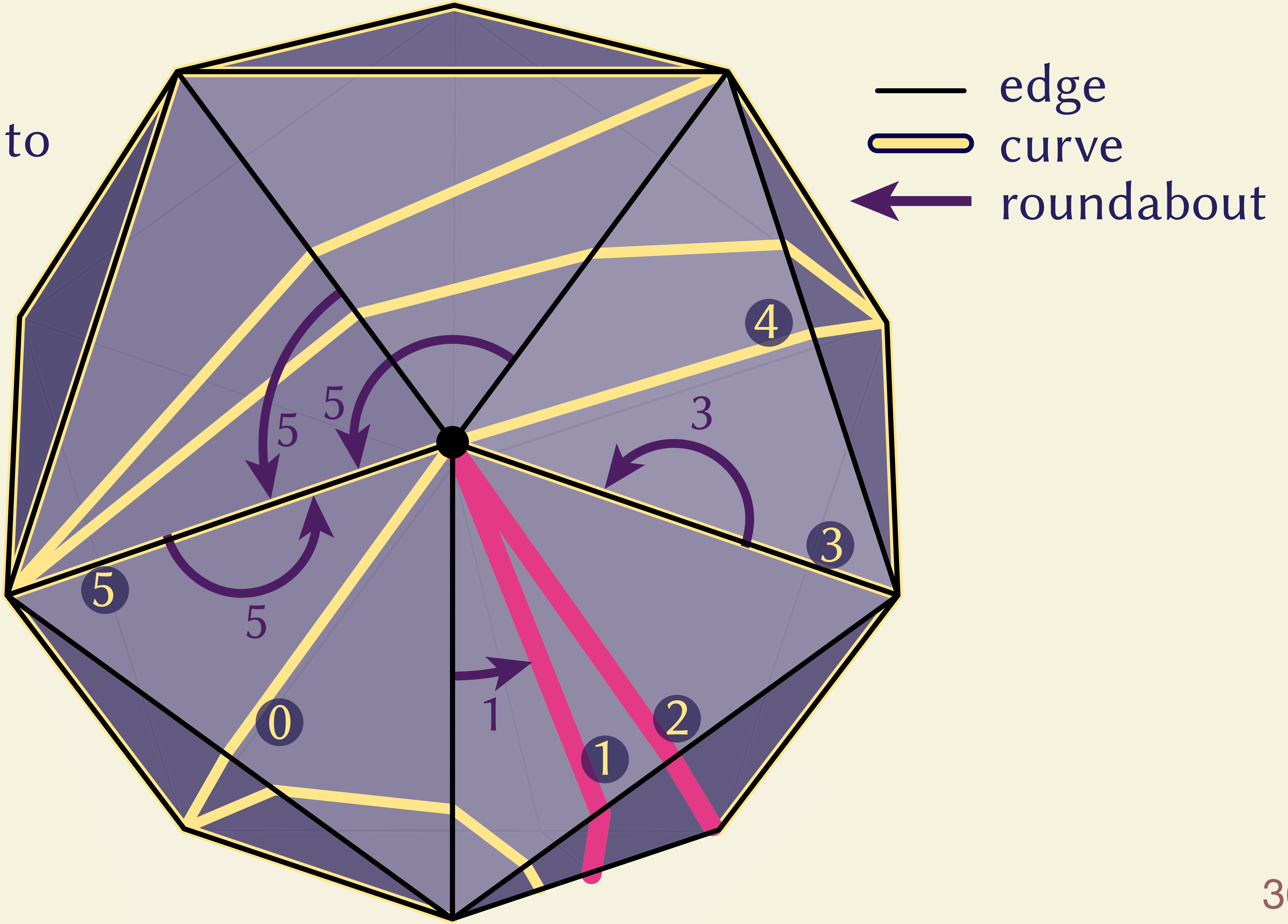
II. Integer coordinates for intrinsic triangulations

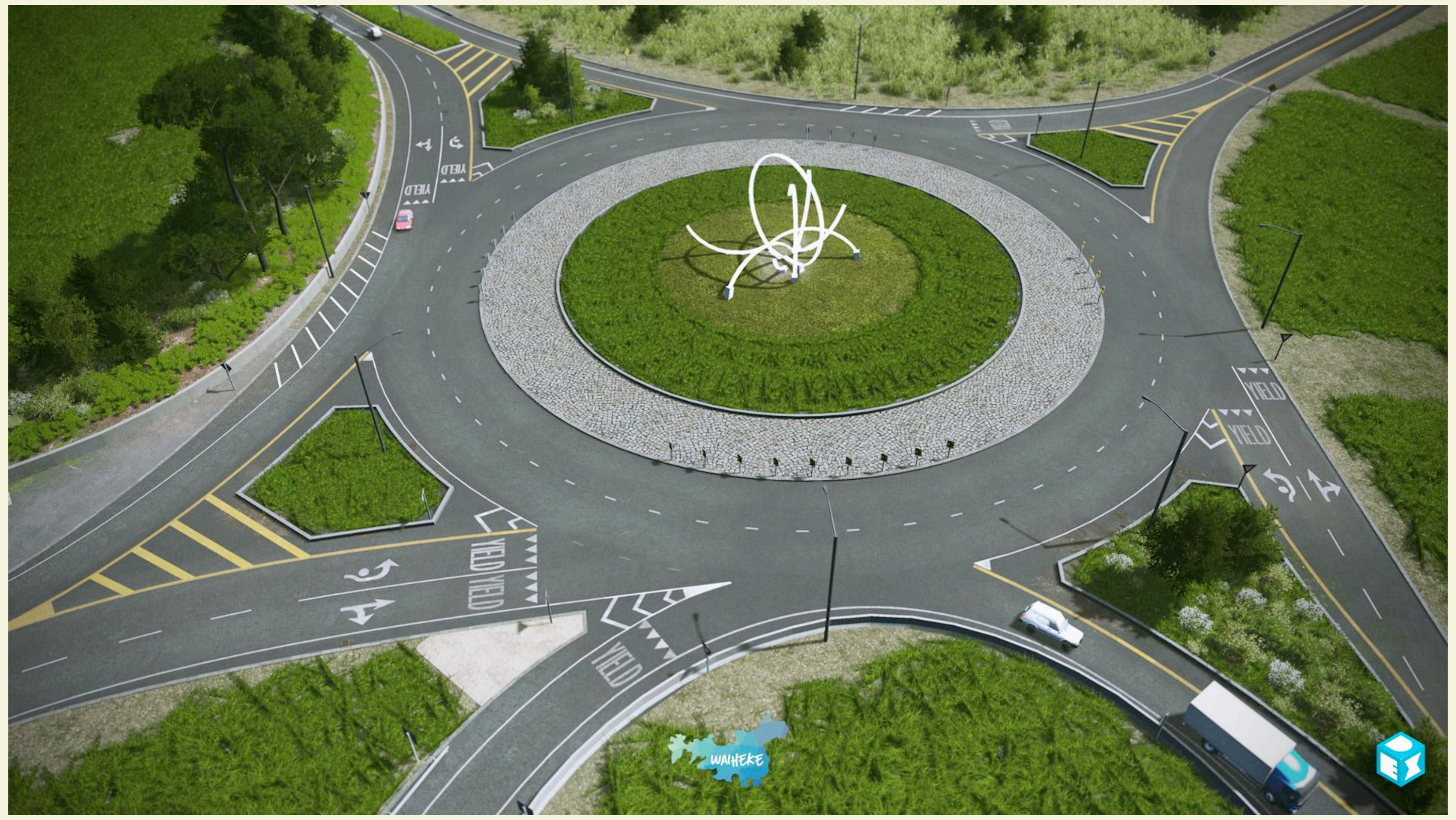
Roundabouts



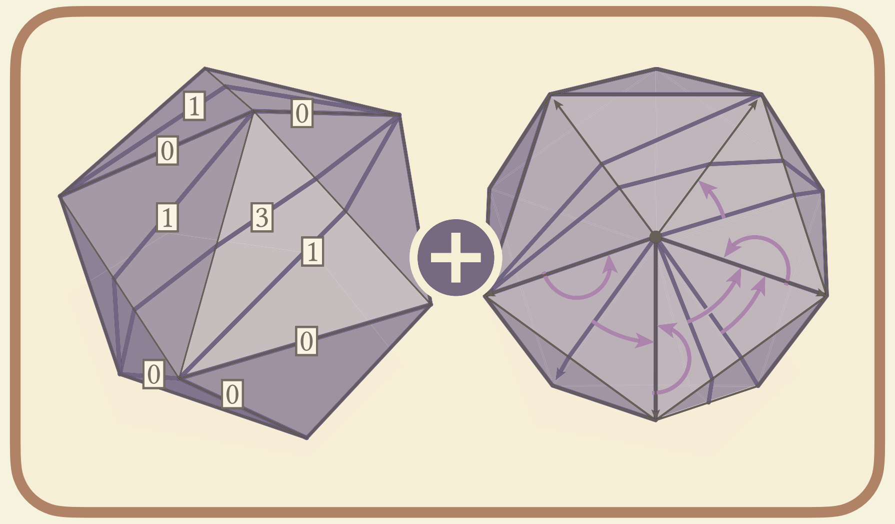
- Each black edge stores a pointer to the next yellow curve

- Resolves all ambiguity

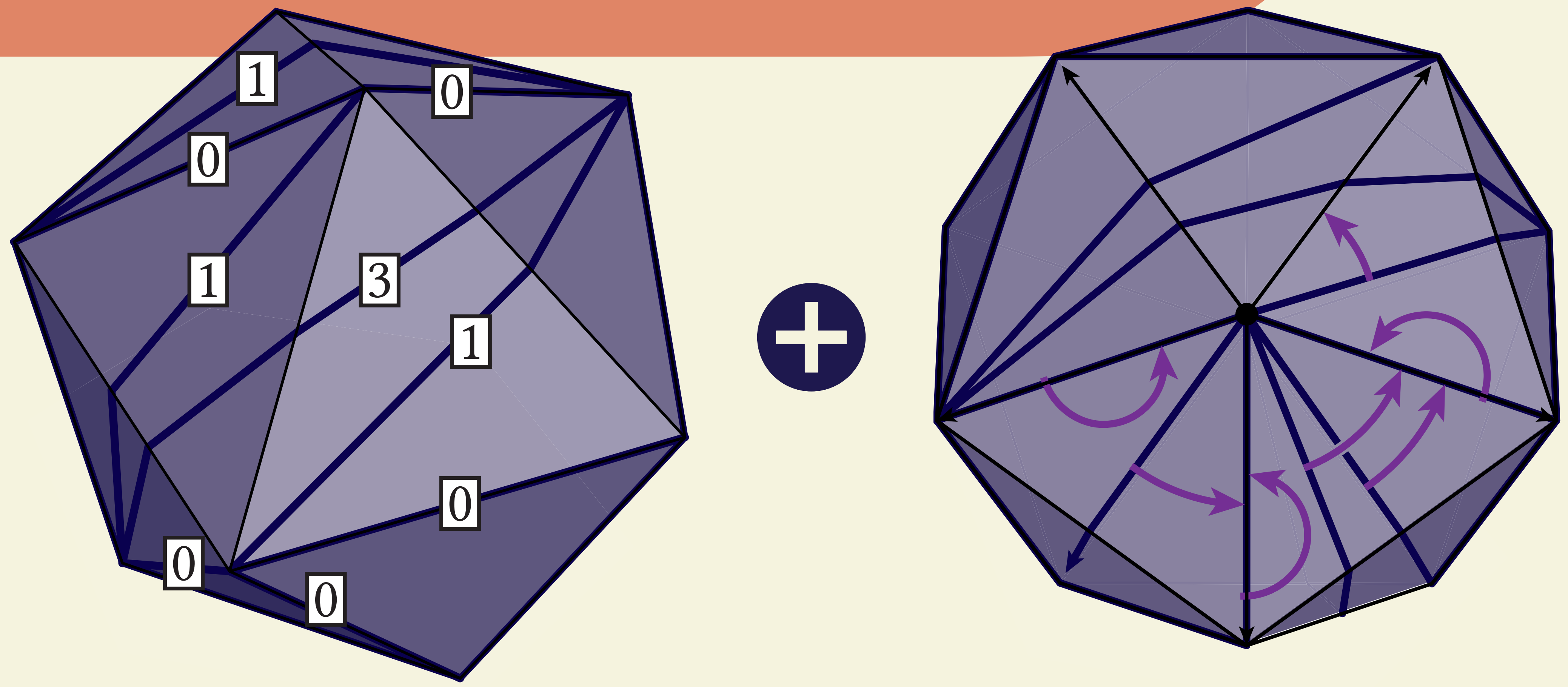




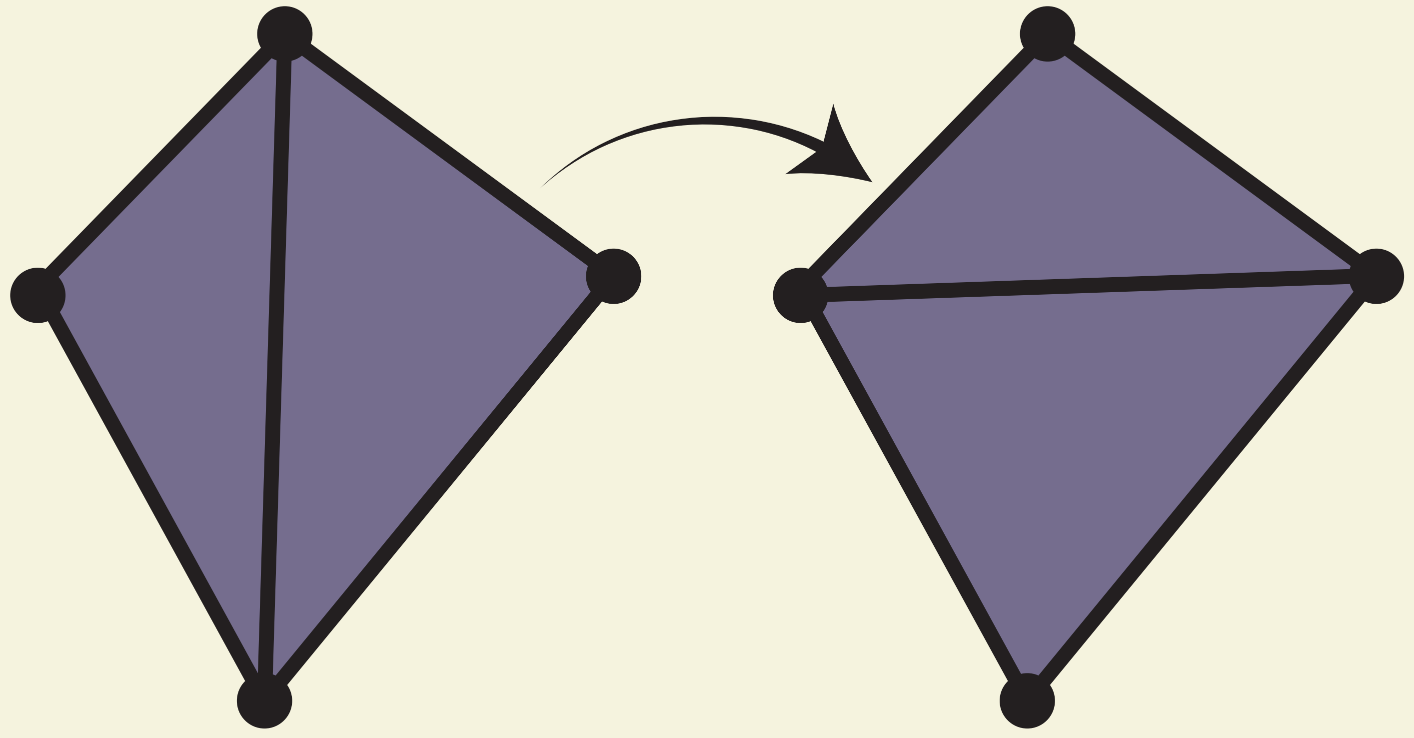
Data structure operations



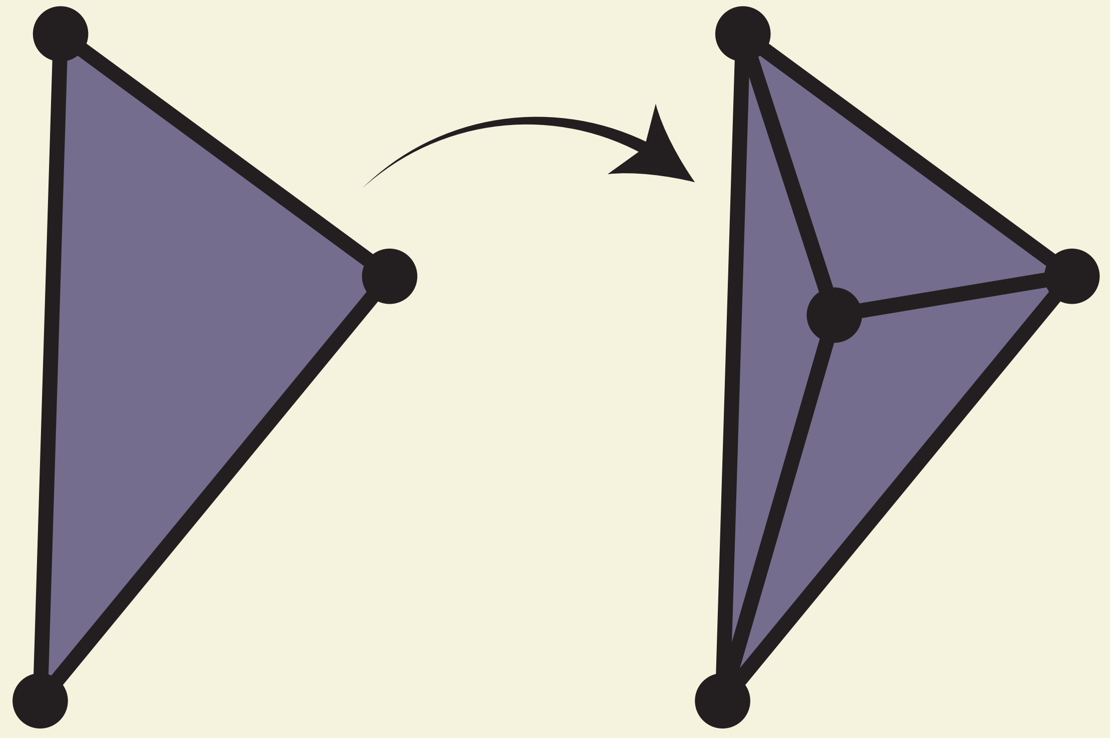
II. Integer coordinates for intrinsic triangulations
▸ *connectivity changes*



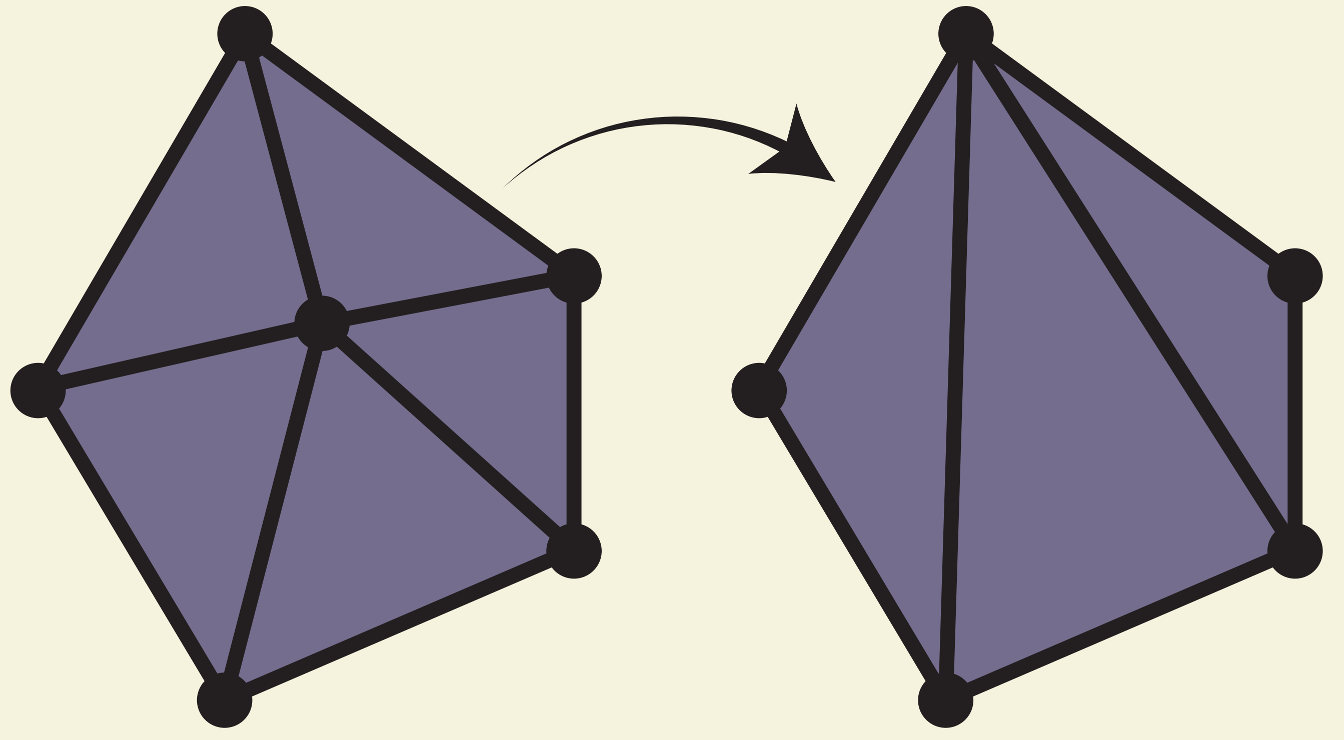
- Supports a variety of connectivity changes:



edge flips

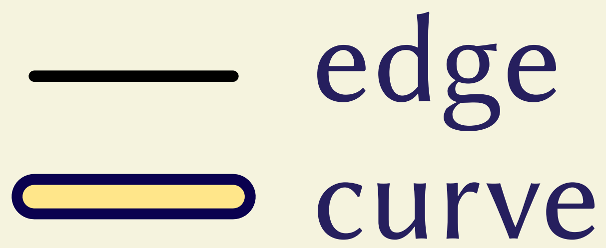


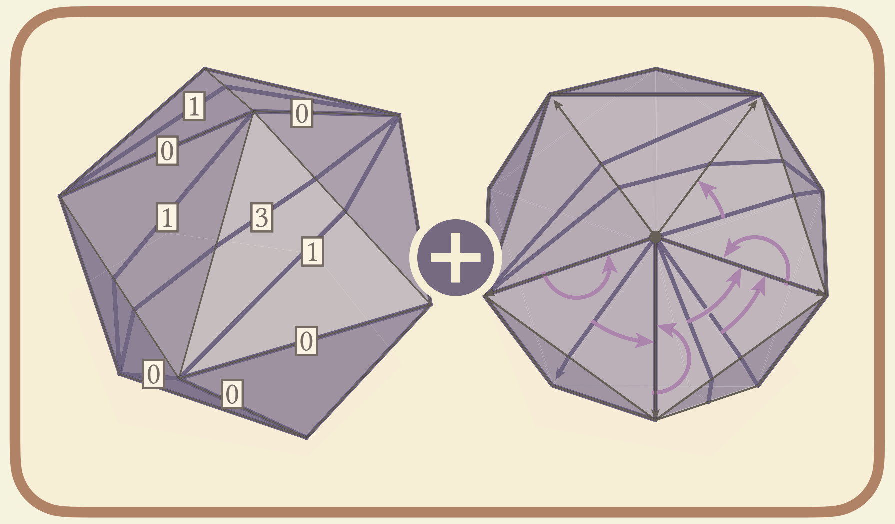
vertex insertion



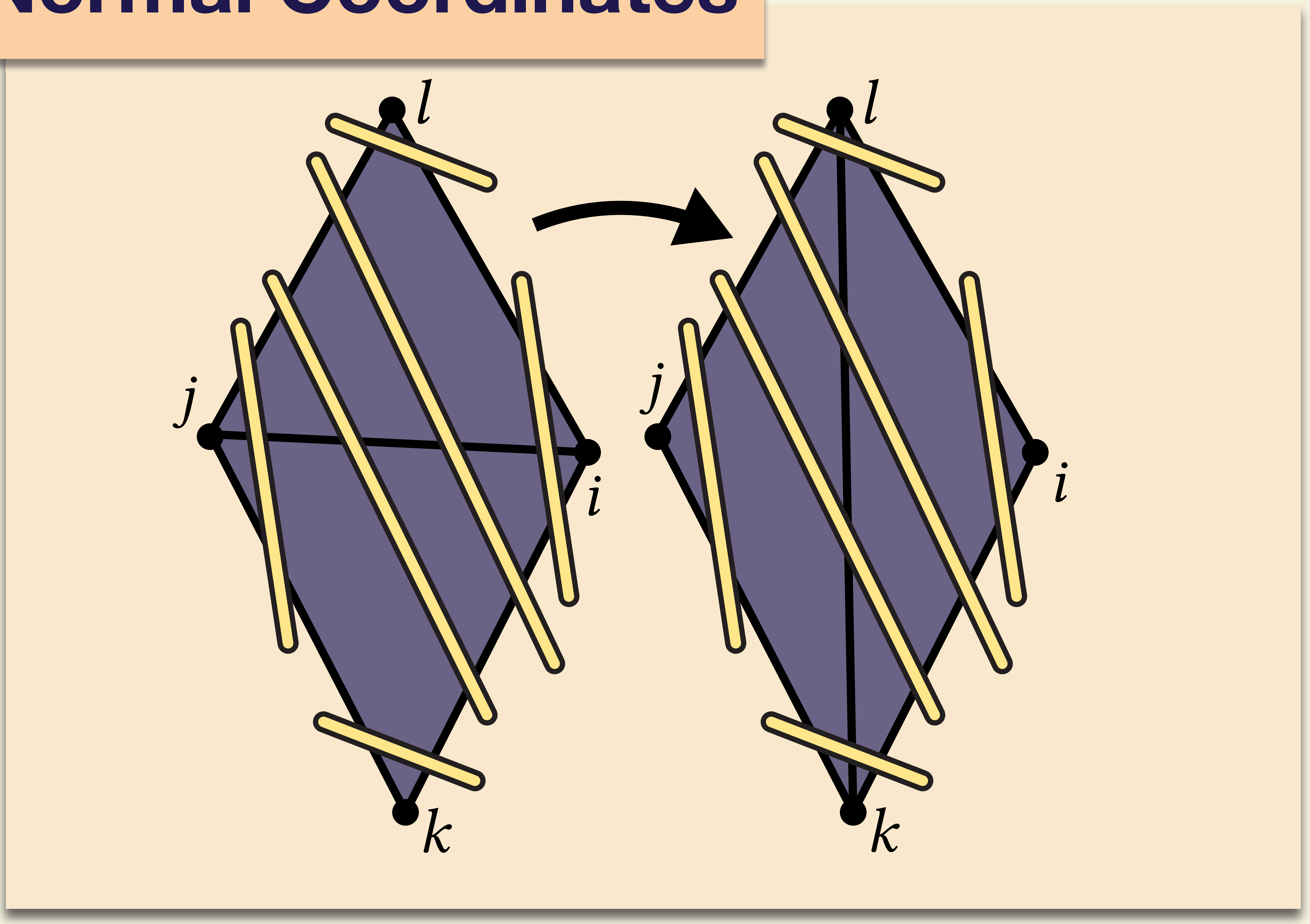
flat vertex removal

Edge flips





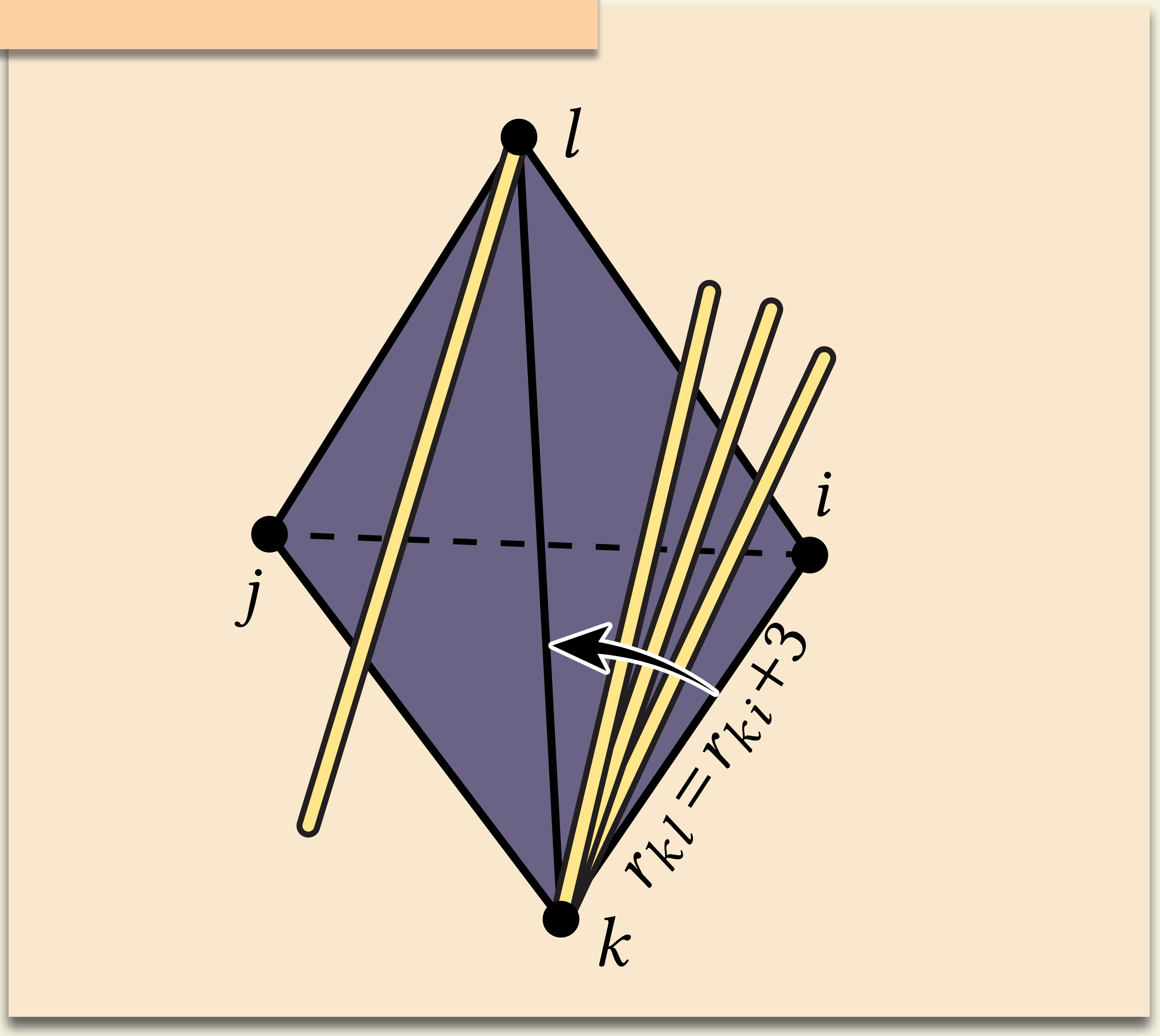
Normal Coordinates



$$n_{kl} = \max(n_{ki} + n_{lj}, n_{jk} + n_{li}) - n_{ij}$$

(if there are no endpoints)

Roundabouts

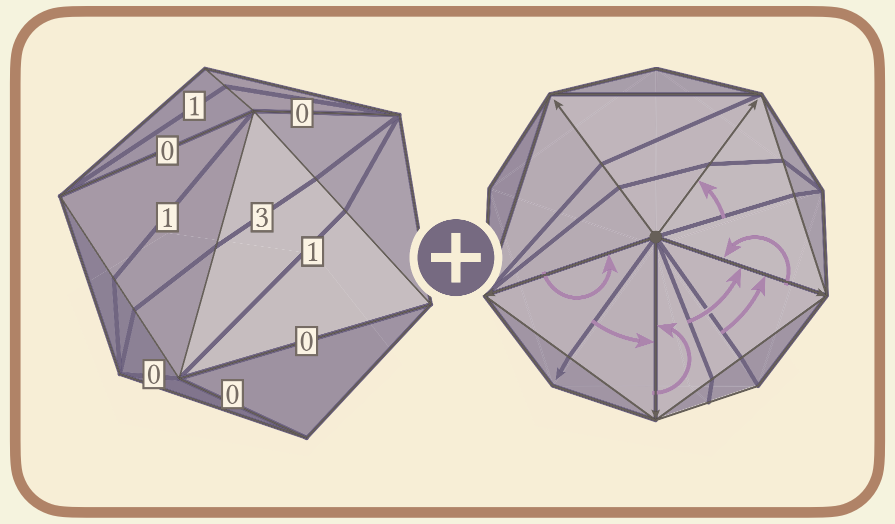


$$r_{kl} = r_{ki} + n_{ki}^{-} + \max(0, n_{il}^{+} - n_{lk}^{+} - n_{ki}^{+})$$

Edge flips

edge

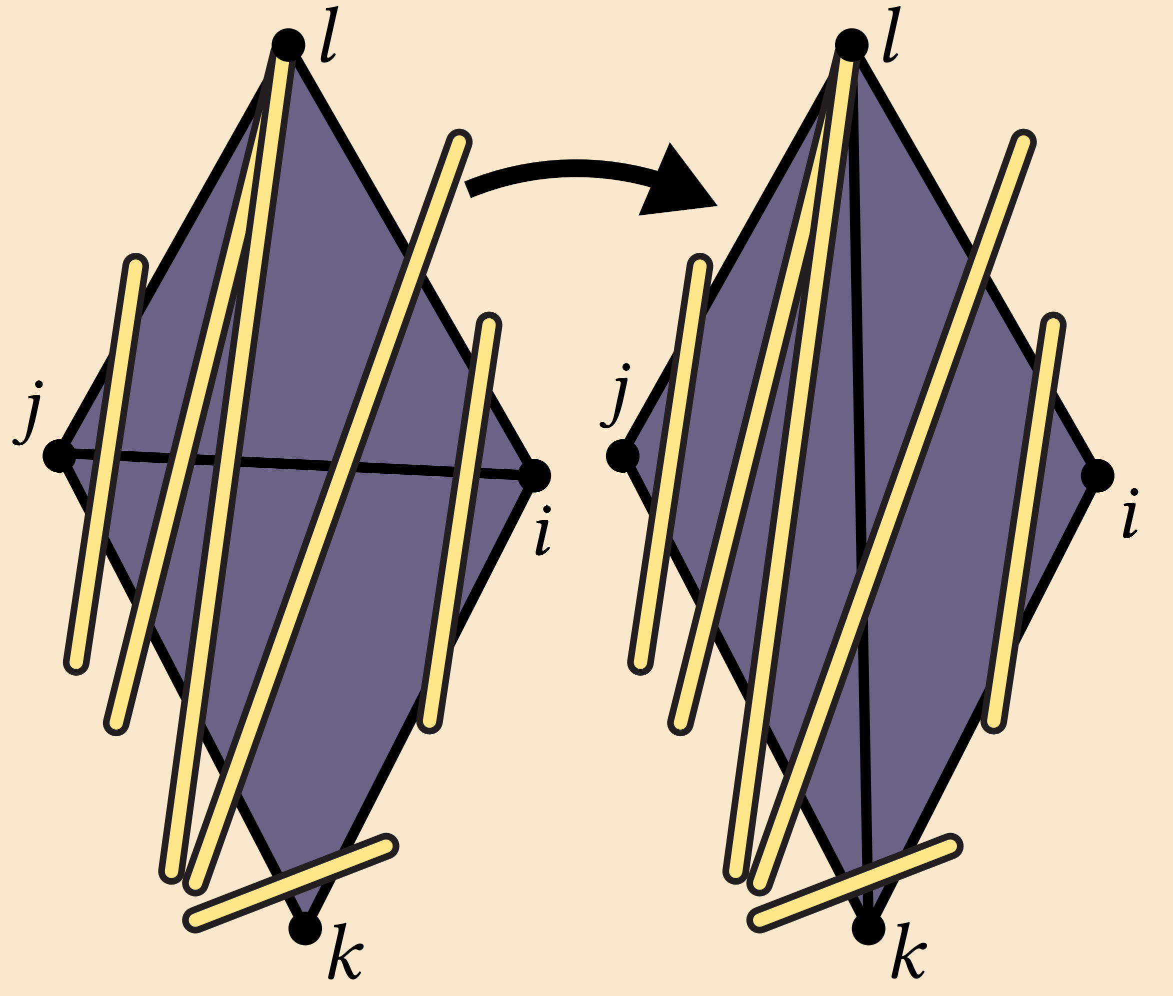
curve



II. Integer coordinates for intrinsic triangulations

▸ *connectivity changes*

Normal Coordinates



$$n_{kl} = c_l^{jk} + c_k^{ij} + \frac{1}{2}\left|c_j^{il} - c_j^{ki}\right| + \frac{1}{2}\left|c_i^{lj} - c_i^{jk}\right| - \frac{1}{2}e_l^{ji} - \frac{1}{2}e_k^{ij} + e_i^{lj} + e_i^{jk} + e_j^{il} + e_j^{ki} + n_{ij}^-$$

(general case)

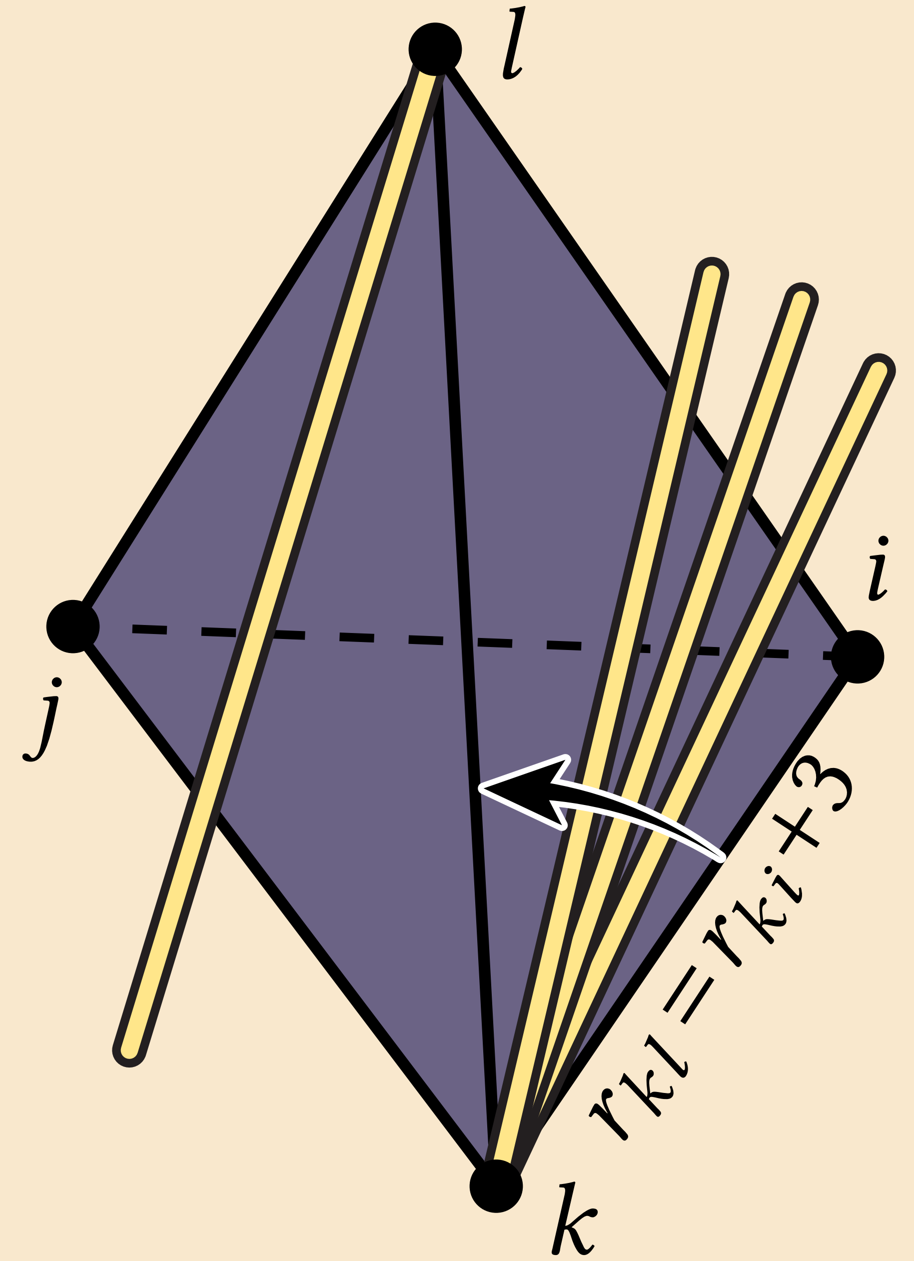
where

$$2c_k^{ij} := \max(0,\, n_{jk}^+ + n_{ki}^+ - n_{ij}^+) - e_i^{jk} - e_j^{ki}$$

$$e_k^{ij} := \max(0,\, n_{ij}^+ - n_{jk}^+ - n_{ki}^+)$$

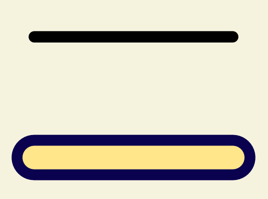
Key takeaway:
closed form flip formulas,
independent of geometry

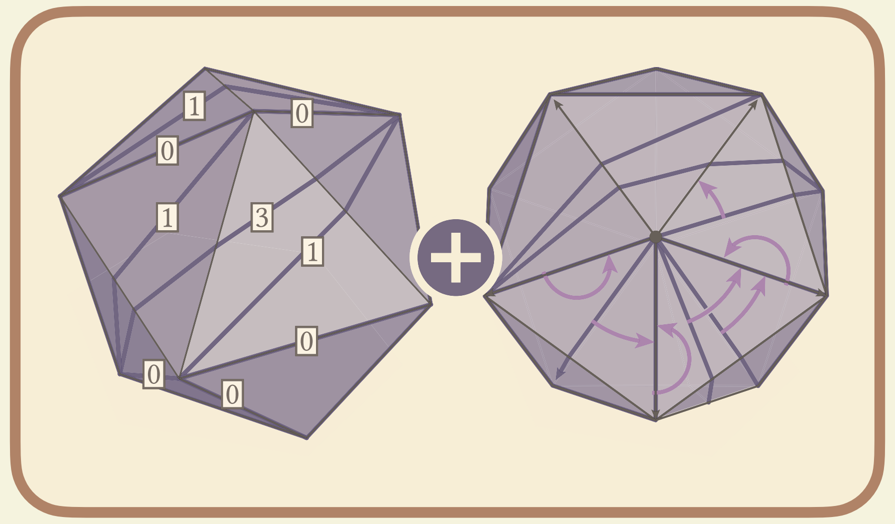
Roundabouts



$$r_{kl} = r_{ki} + n_{ki}^- + \max(0,\, n_{il}^+ - n_{lk}^+ - n_{ki}^+)$$

Vertex insertion

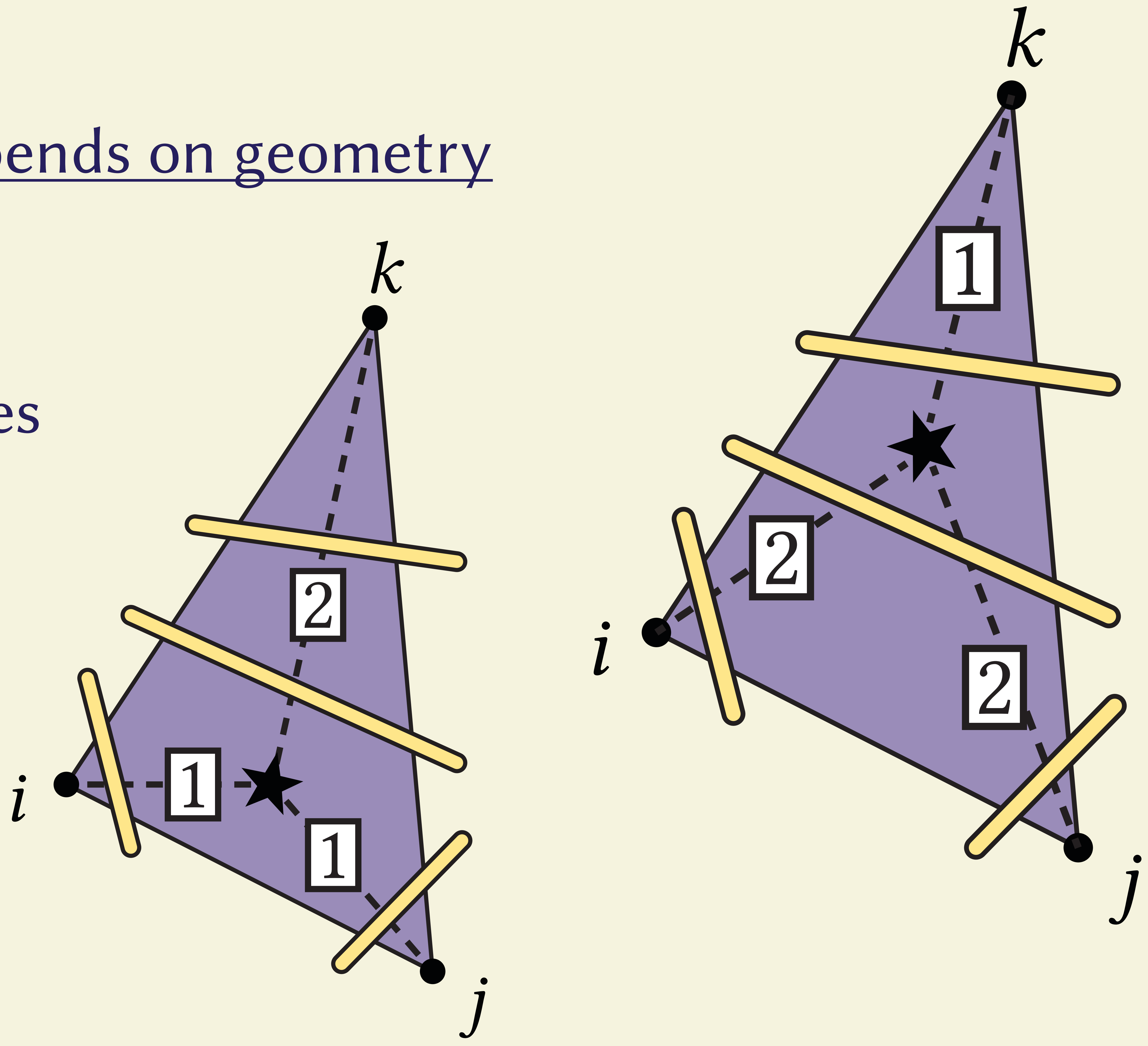




II. Integer coordinates for intrinsic triangulations

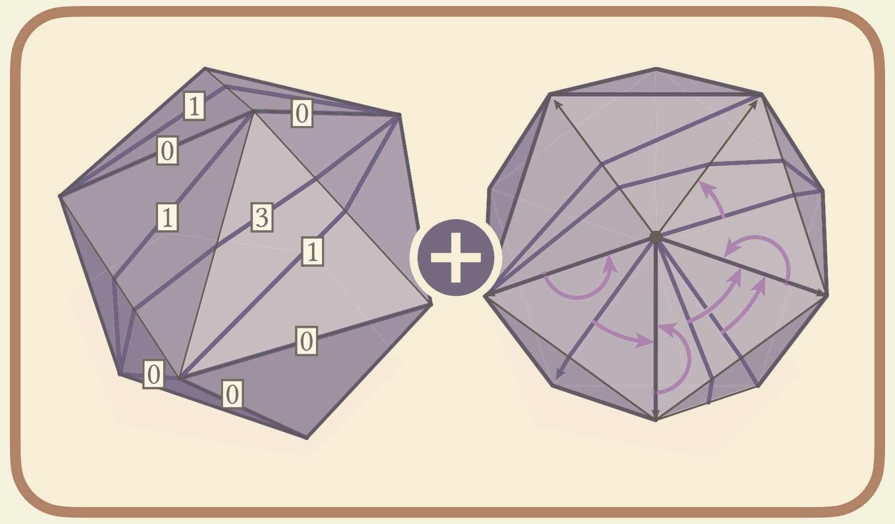
▸ *connectivity changes*

- *Unlike* classic normal coordinates, depends on geometry
- Not a computational challenge:
  1. Locate curves via normal coordinates
  2. Count intersections
  3. Update roundabouts



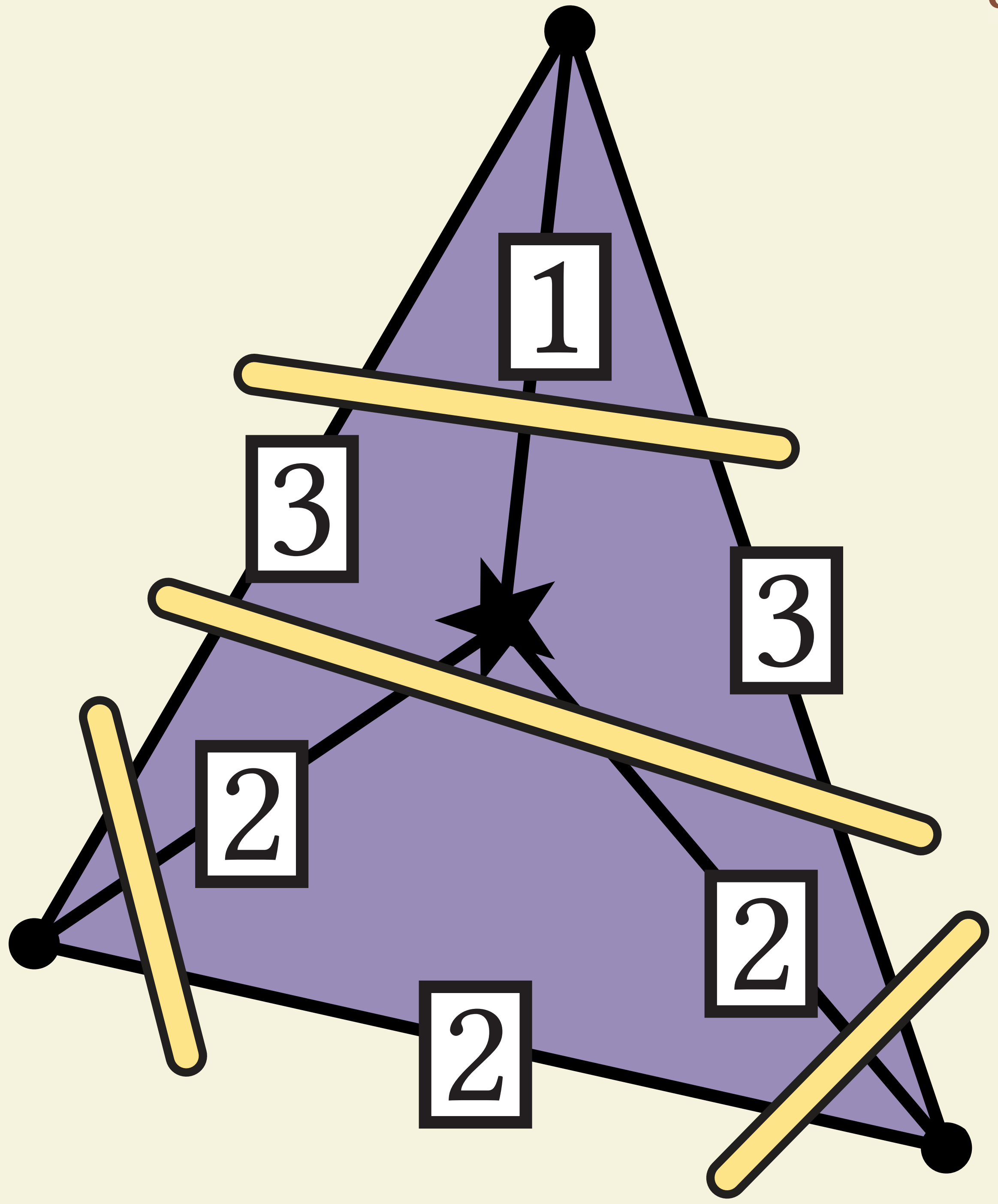
Vertex removal

edge
curve

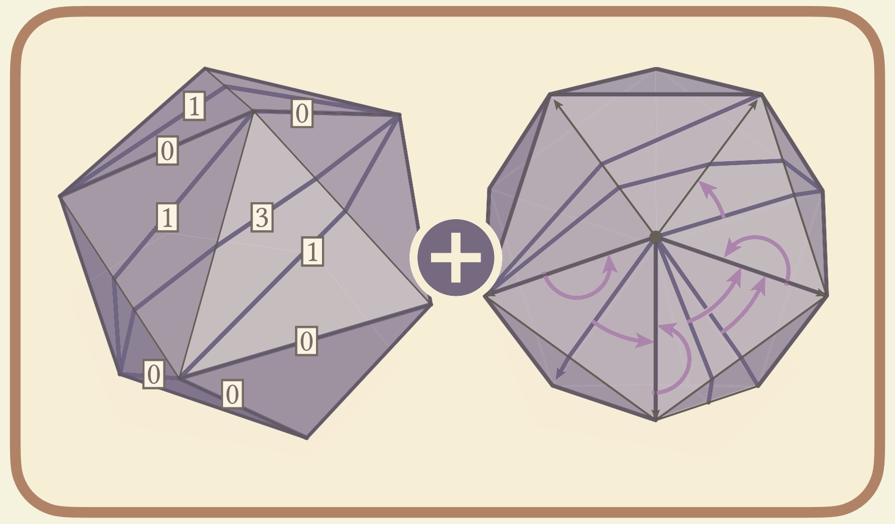


II. Integer coordinates for intrinsic triangulations
‣ *connectivity changes*

- Only remove inserted vertices
- Strategy: reduce to degree-3 case

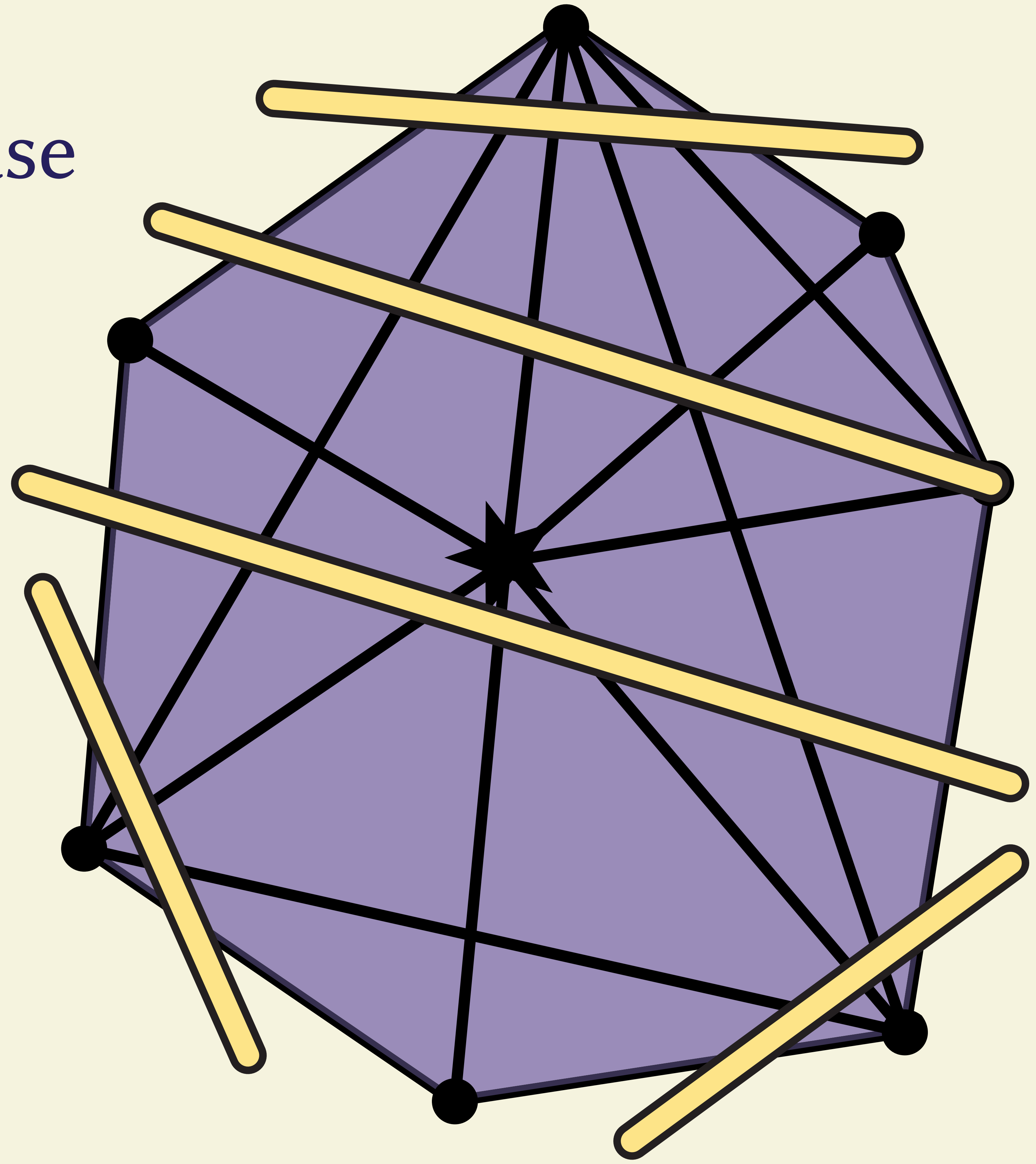


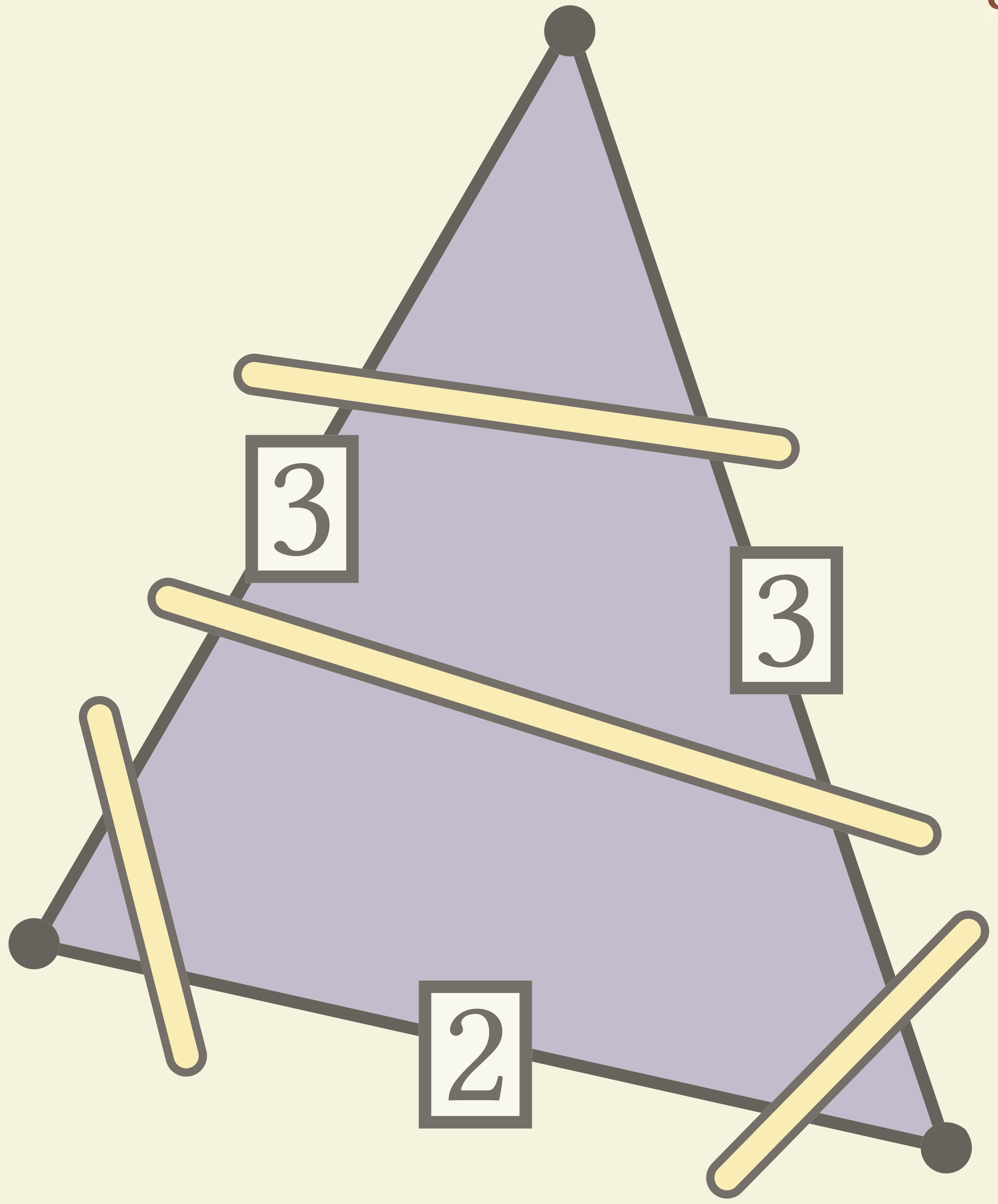
Vertex removal



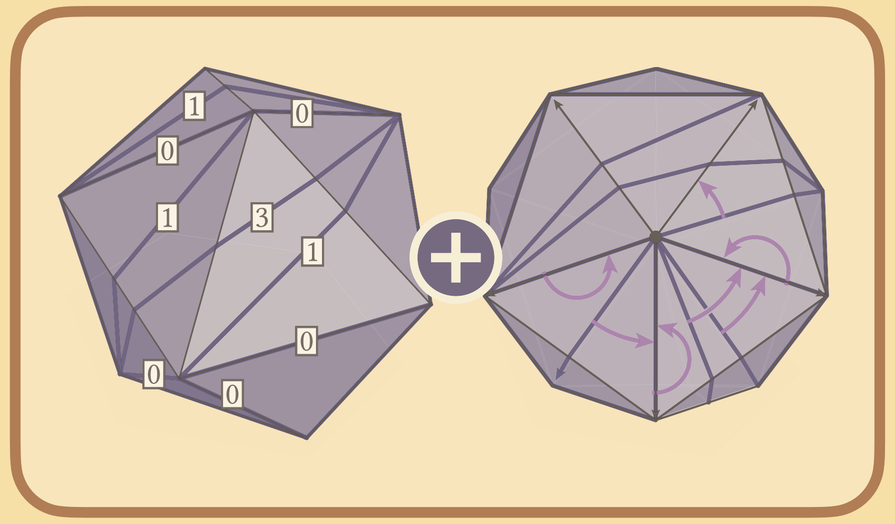
II. Integer coordinates for intrinsic triangulations
▸ *connectivity changes*

- Only remove <u>inserted</u> vertices
- Strategy: reduce to degree-3 case



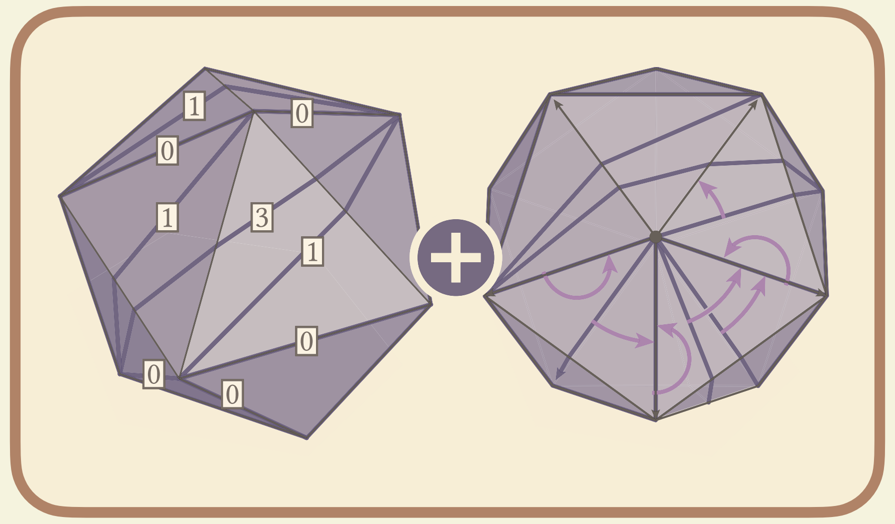


Applications

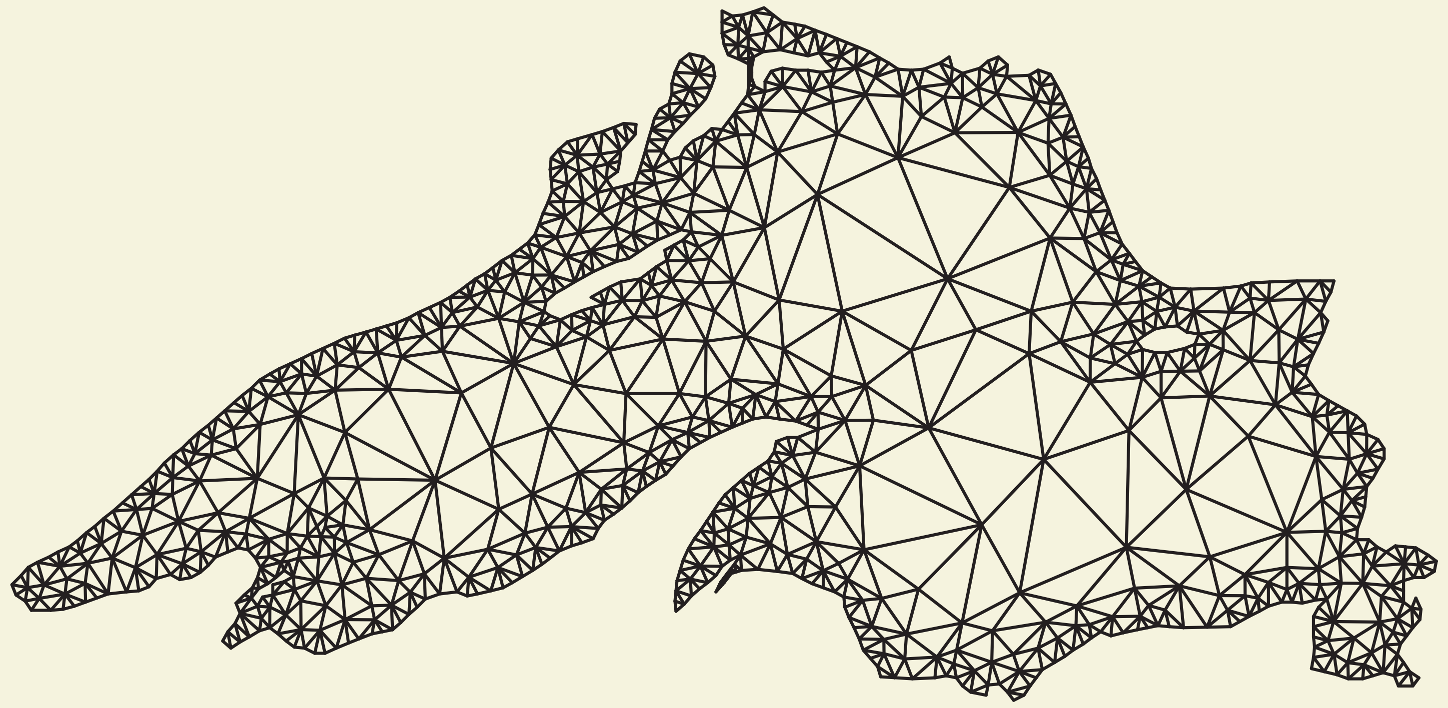


II. Integer coordinates for intrinsic triangulations

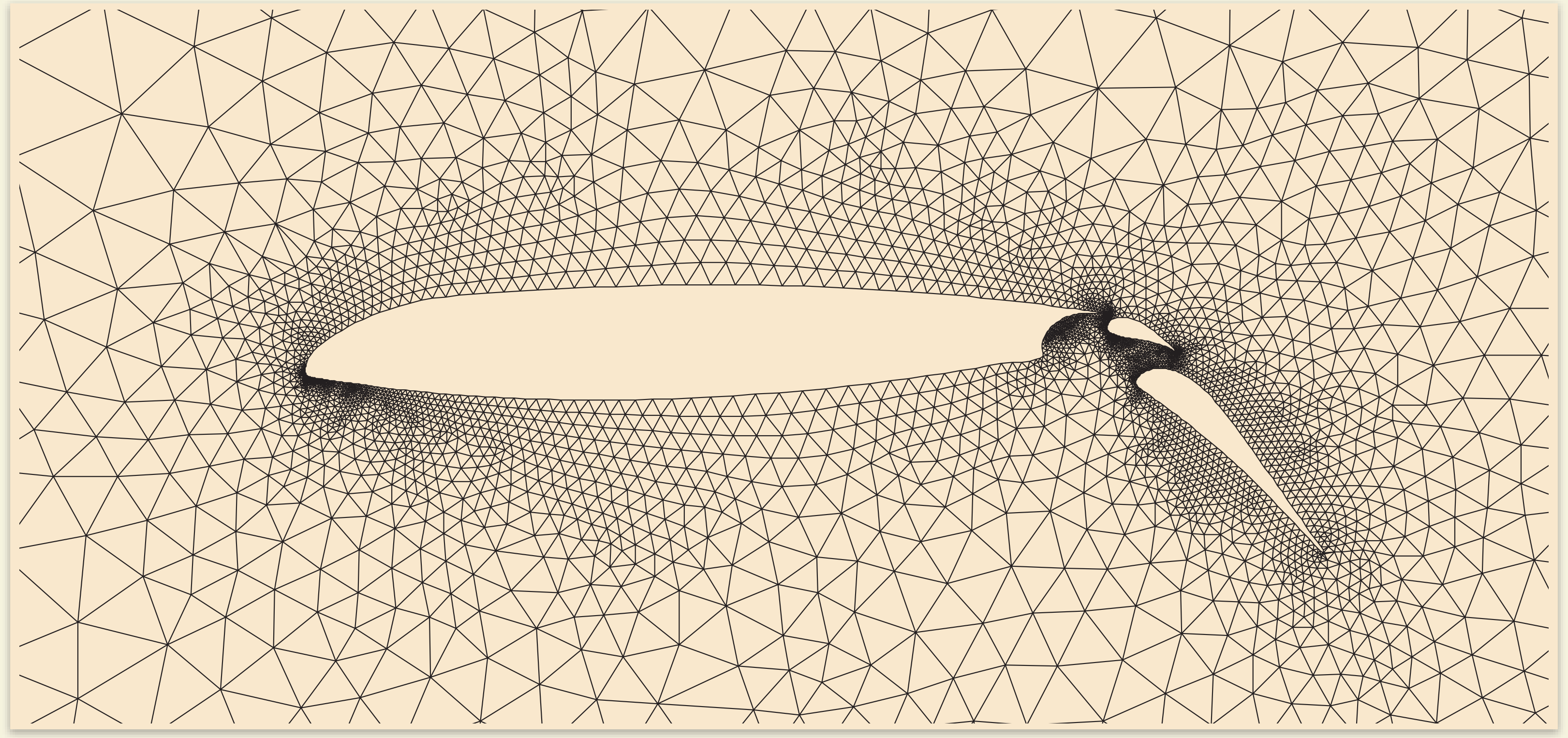
Delaunay refinement for planar meshing



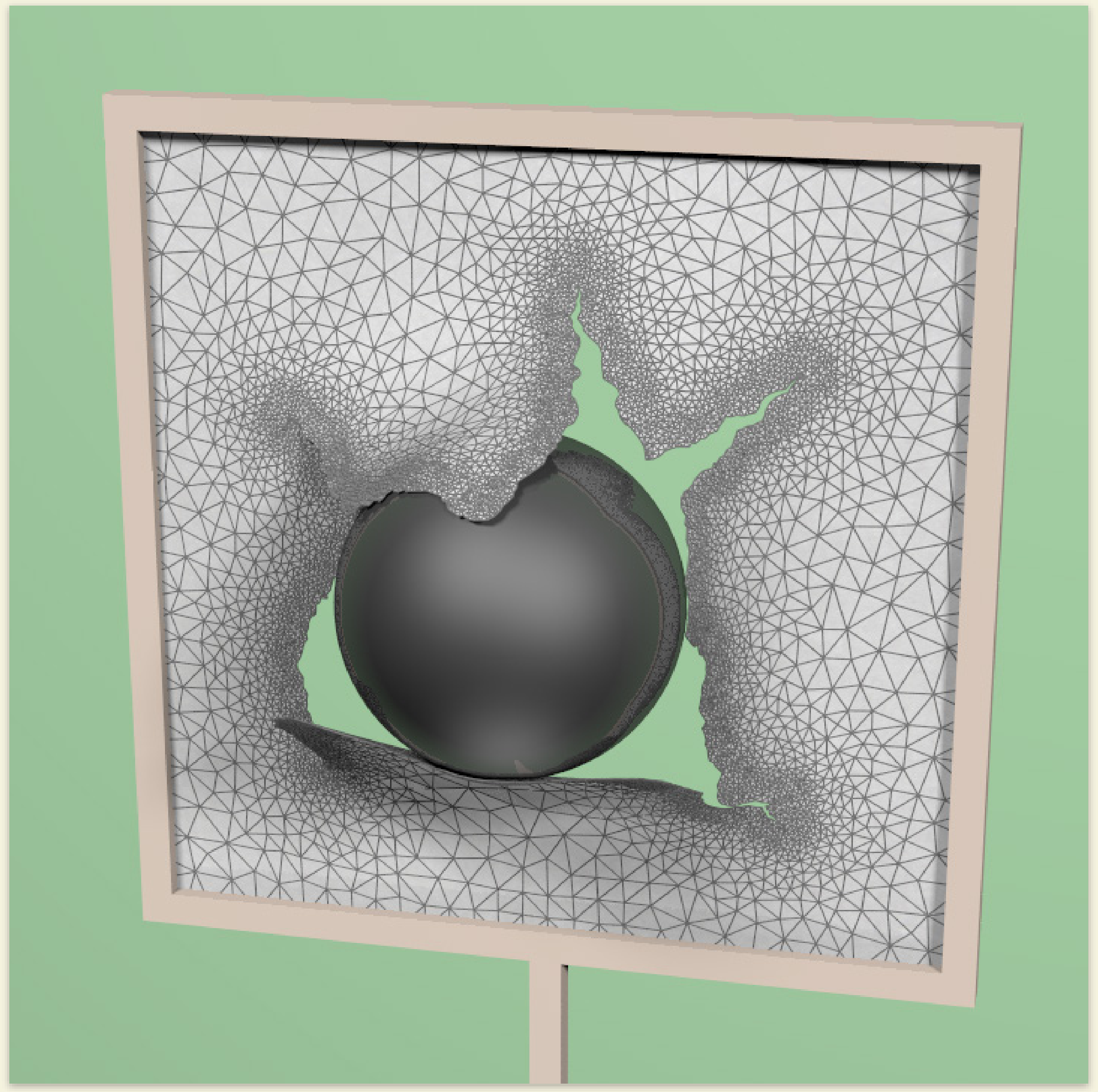
- Crucial tool in 2D - remesh with guaranteed quality bounds [Chew 1993; Shewchuk 1997]



[Shewchuk 1997]

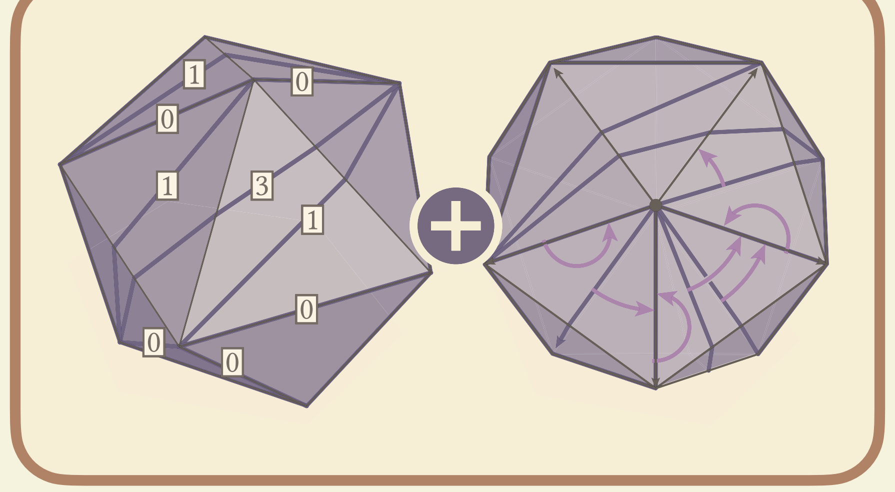


[Shewchuk 1997]



[Busaryev+ 2013]

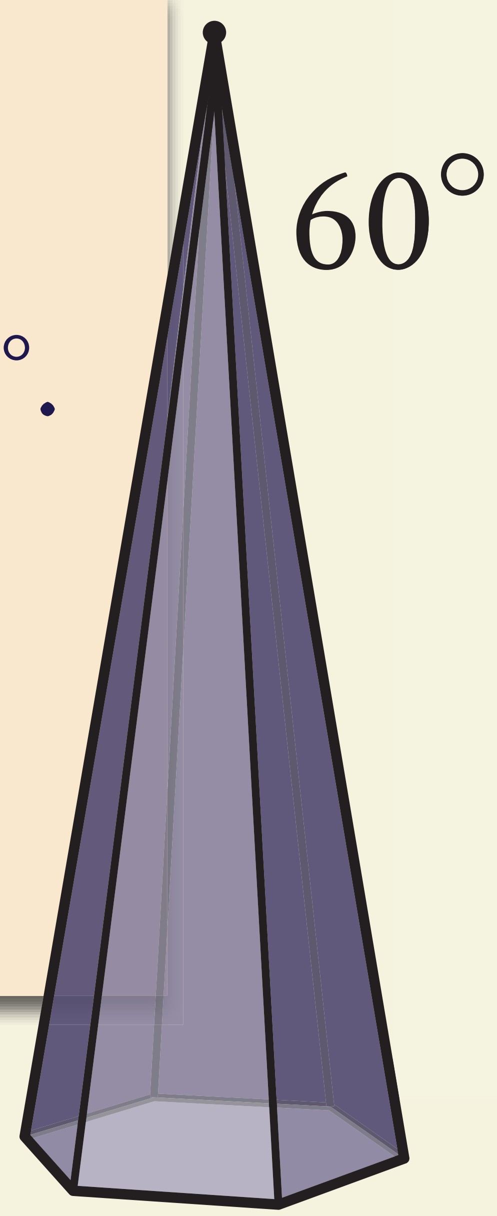
Intrinsic Delaunay refinement

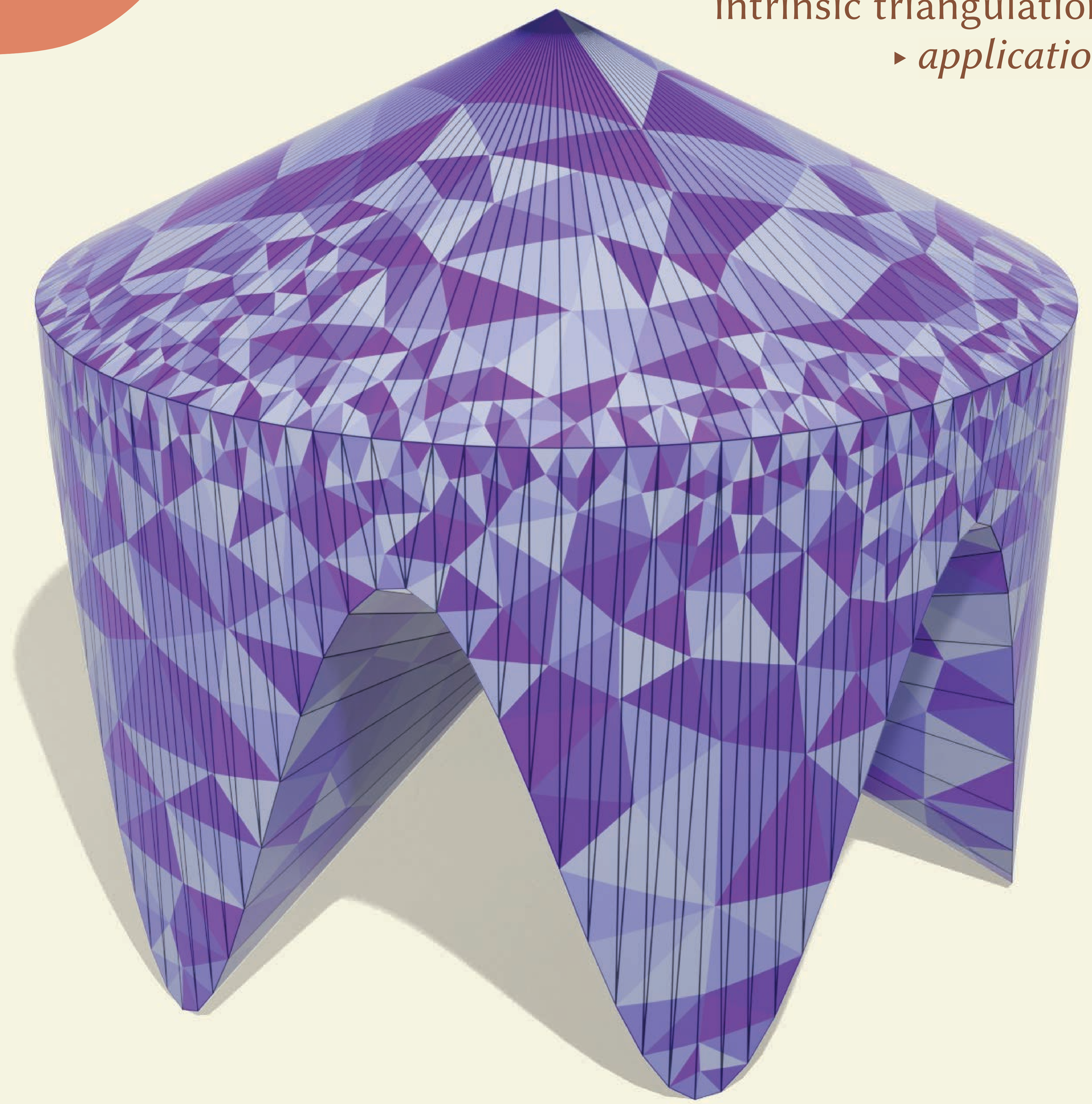


- Intrinsic retriangulation algorithm proposed by [Sharp, Soliman & Crane 2019]

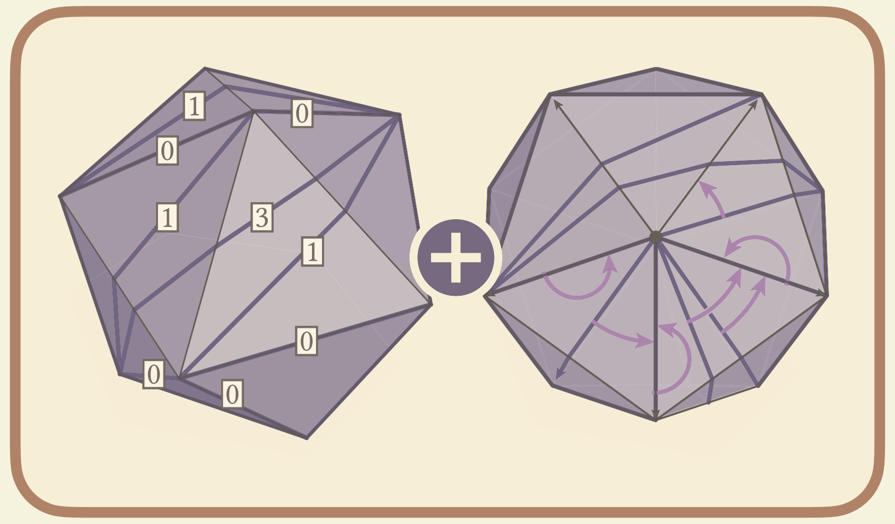
Theorem [G., Sharp & Crane 2021]

Let M be a mesh without boundary whose cone angles are all at least 60°. Then intrinsic Delaunay refinement produces a Delaunay mesh with triangle corner angles at least 30°



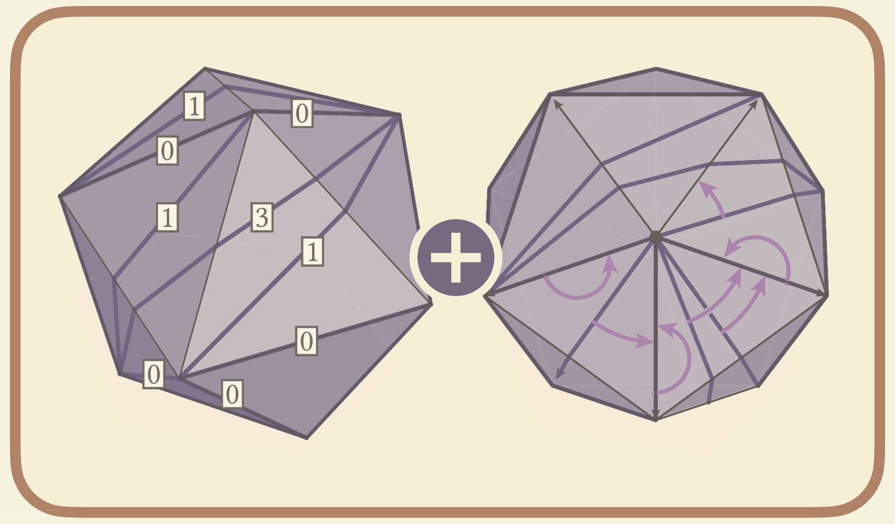


Common subdivisions of intrinsic Delaunay refinements



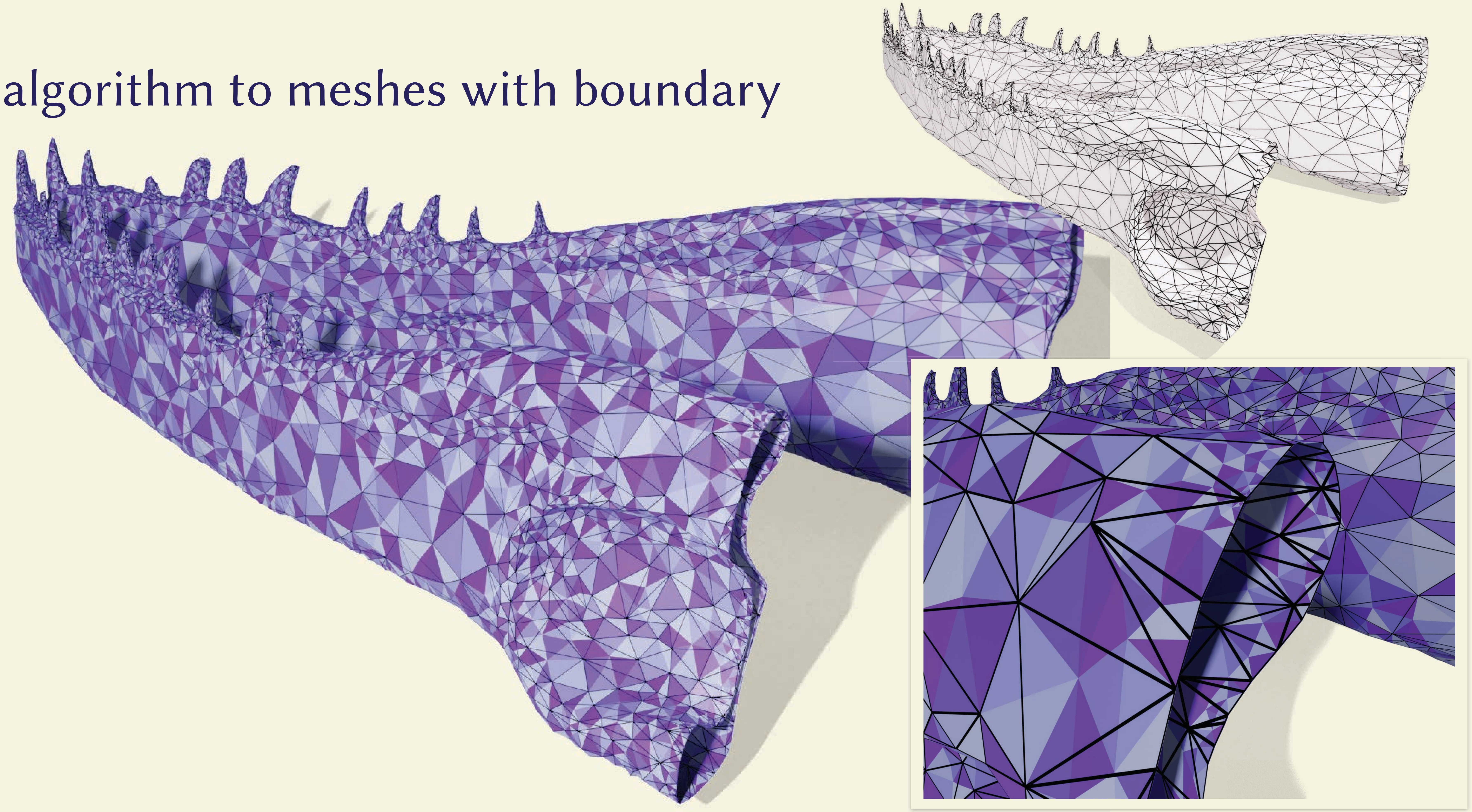
- Integer coordinates can be crucial to recovering the common subdivision

Intrinsic Delaunay refinement of meshes with boundary



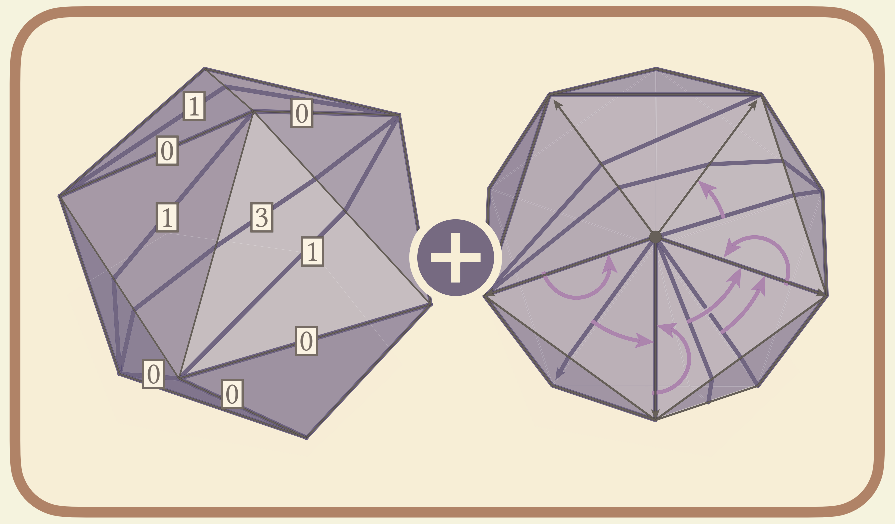
II. Integer coordinates for intrinsic triangulations
▸ *applications*

- Extend algorithm to meshes with boundary



ThingiID 48352

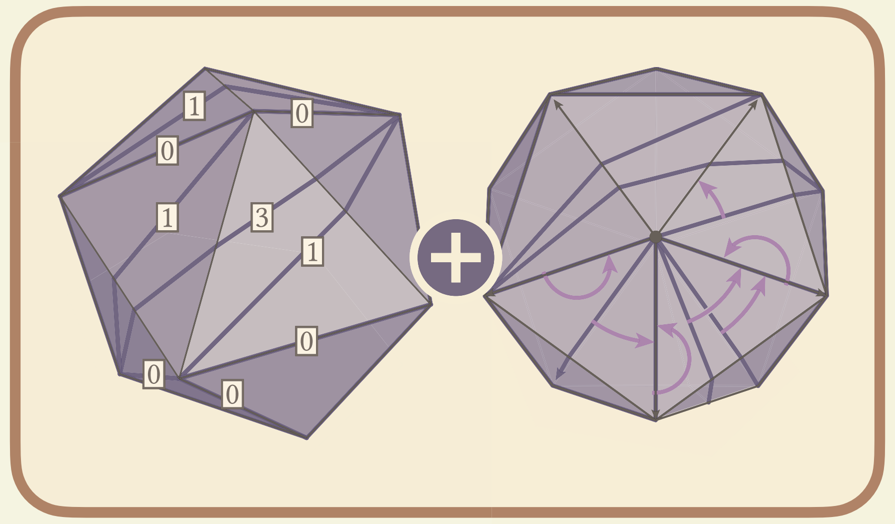
Intrinsic Delaunay refinement — validation



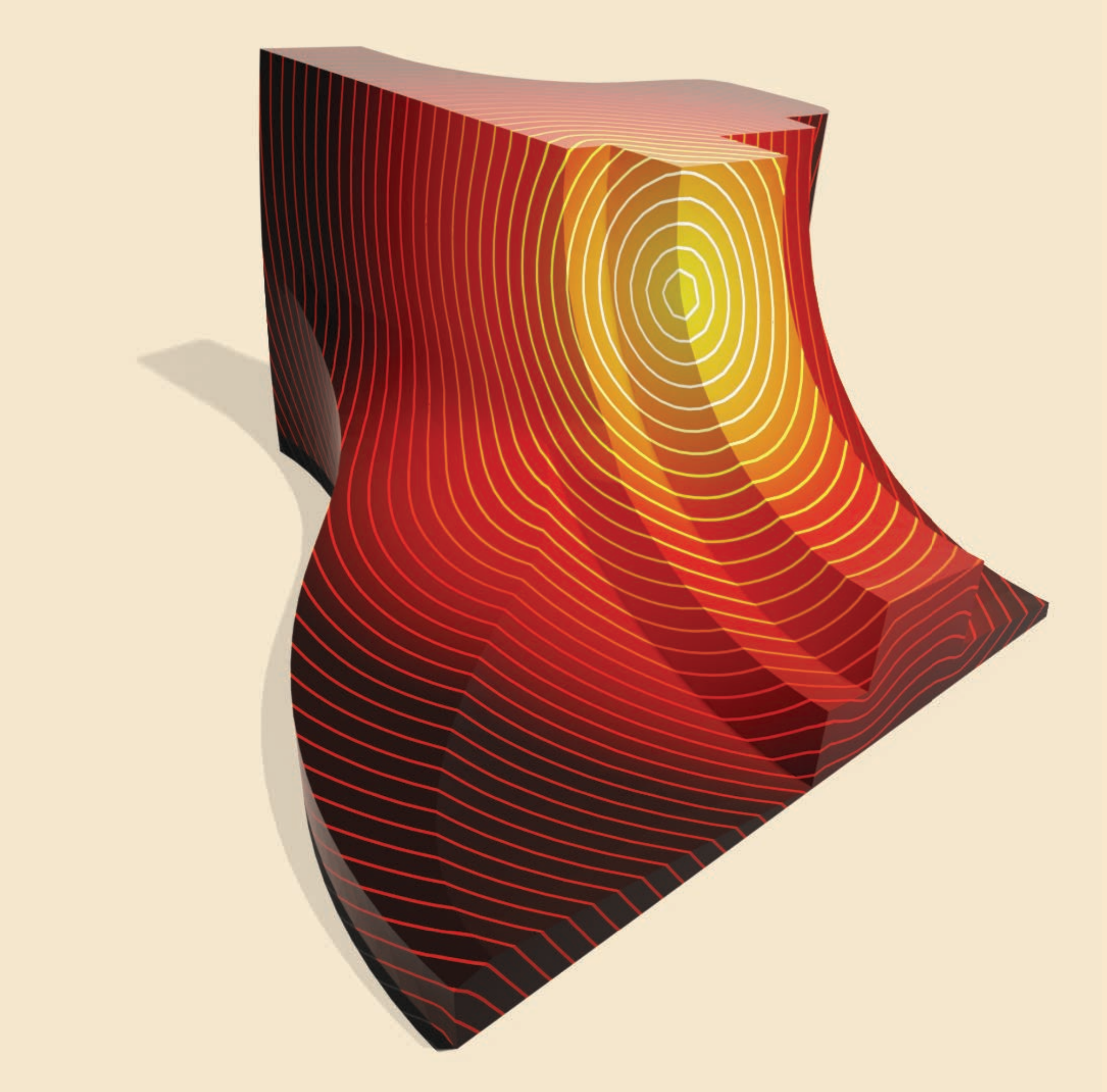
II. Integer coordinates for intrinsic triangulations
▸ *applications*

- Compute refinements & common subdivisions for Thingi10k dataset [Zhou & Jacobson 2016]
  - 7696 manifold meshes
- < 1s on most meshes; only took > 1m on 6 meshes
- 100% success rate for refinement & common subdivision
  - [Sharp, Soliman & Crane 2019] succeed on only 69.1% of meshes

Application: PDE-Based Geometry Processing



II. Integer coordinates for intrinsic triangulations
▸ *applications*



heat method for geodesic distance [Crane, Weischedel & Wardetzky 2013]

ThingiID 44395

mean error: 28%

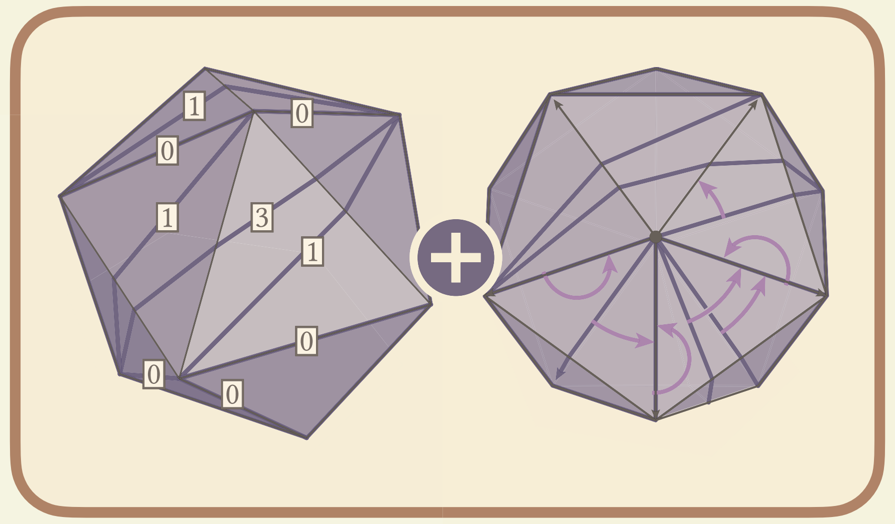
result on input mesh

mean error: 2%

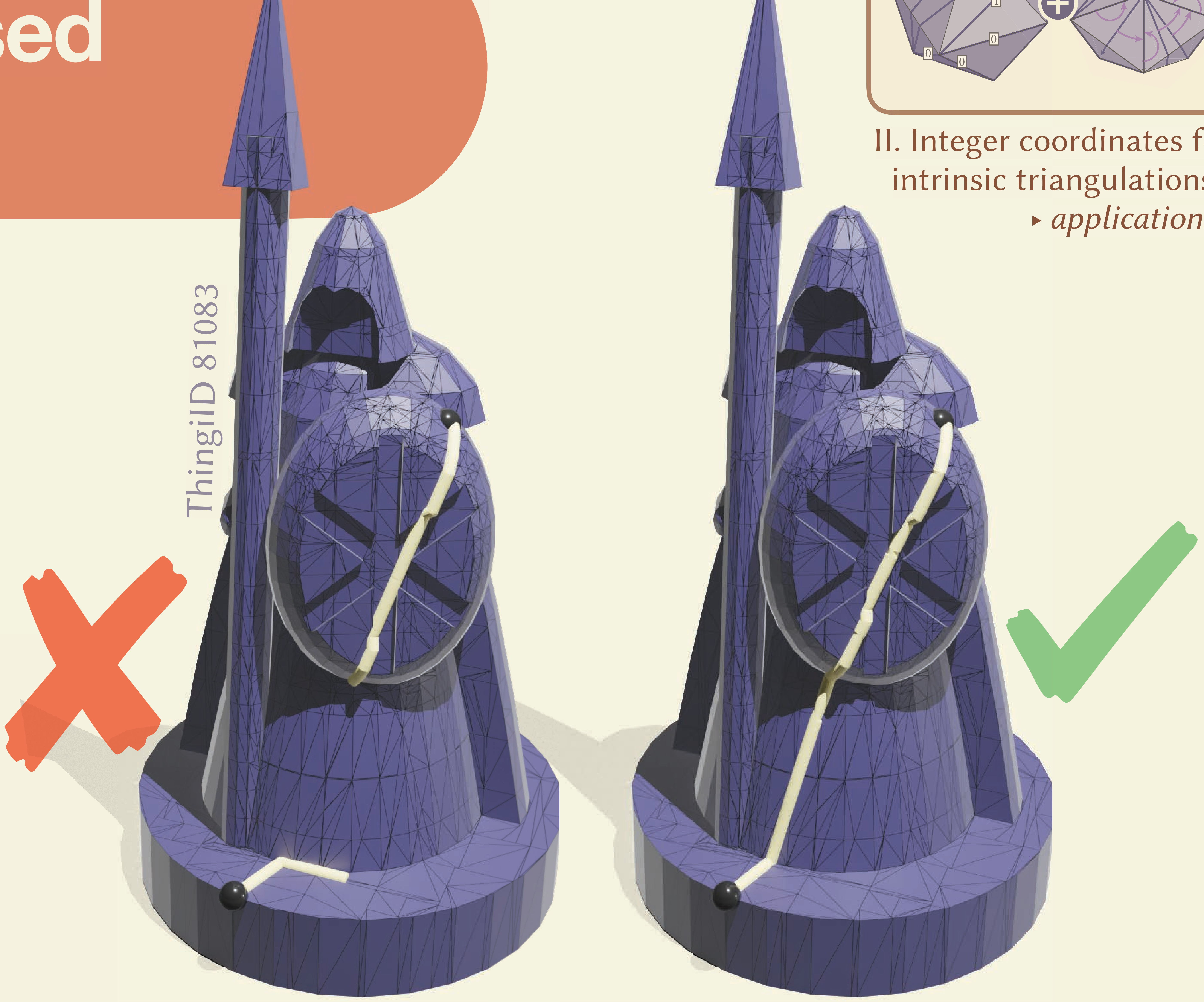
result on Delaunay refinement

Application: Flip-Based Geodesic Paths

- FlipOut [Sharp & Crane 2020]:
  - computes geodesic paths via edge flips



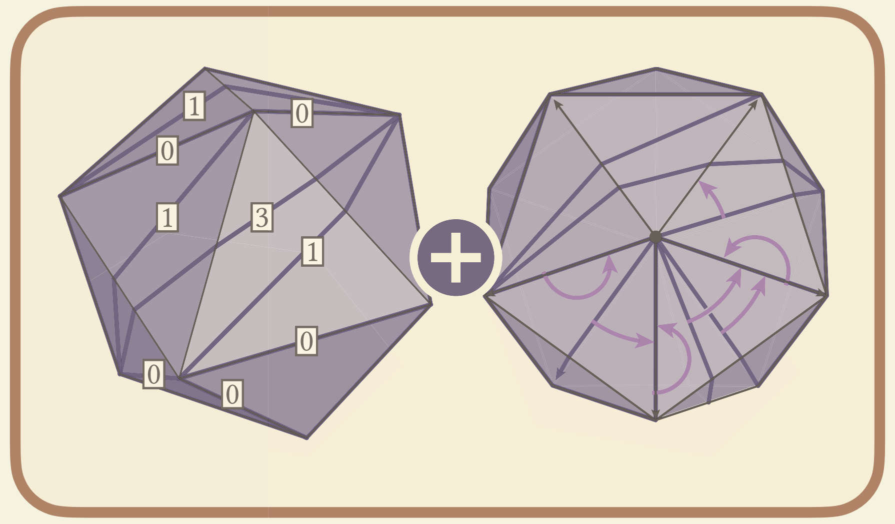
II. Integer coordinates for intrinsic triangulations
▸ *applications*



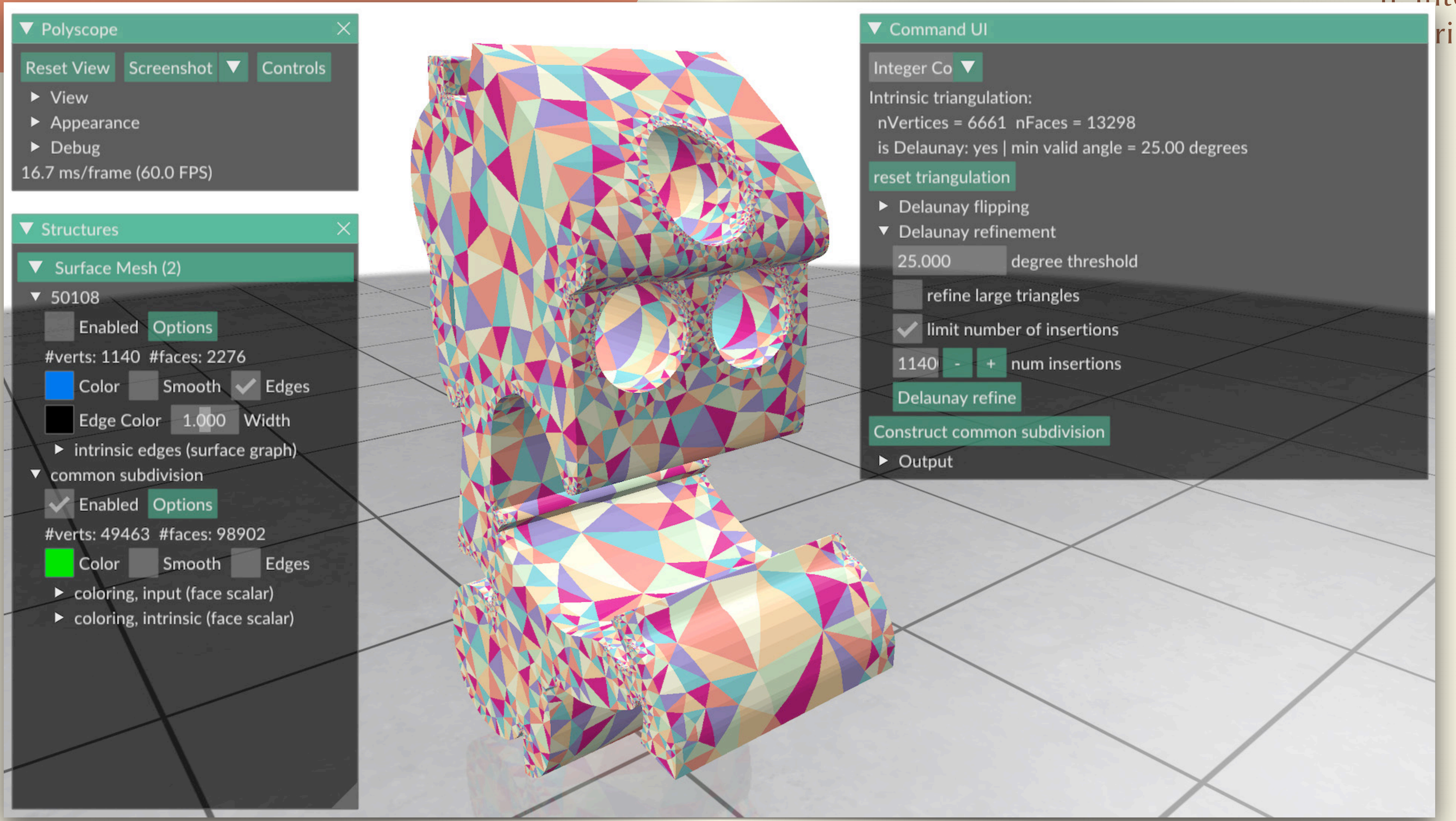
[Sharp, Soliman & Crane 2019]

[Integer coordinates]

Try it out yourself



II. Integer coordinates for
rinsic triangulations

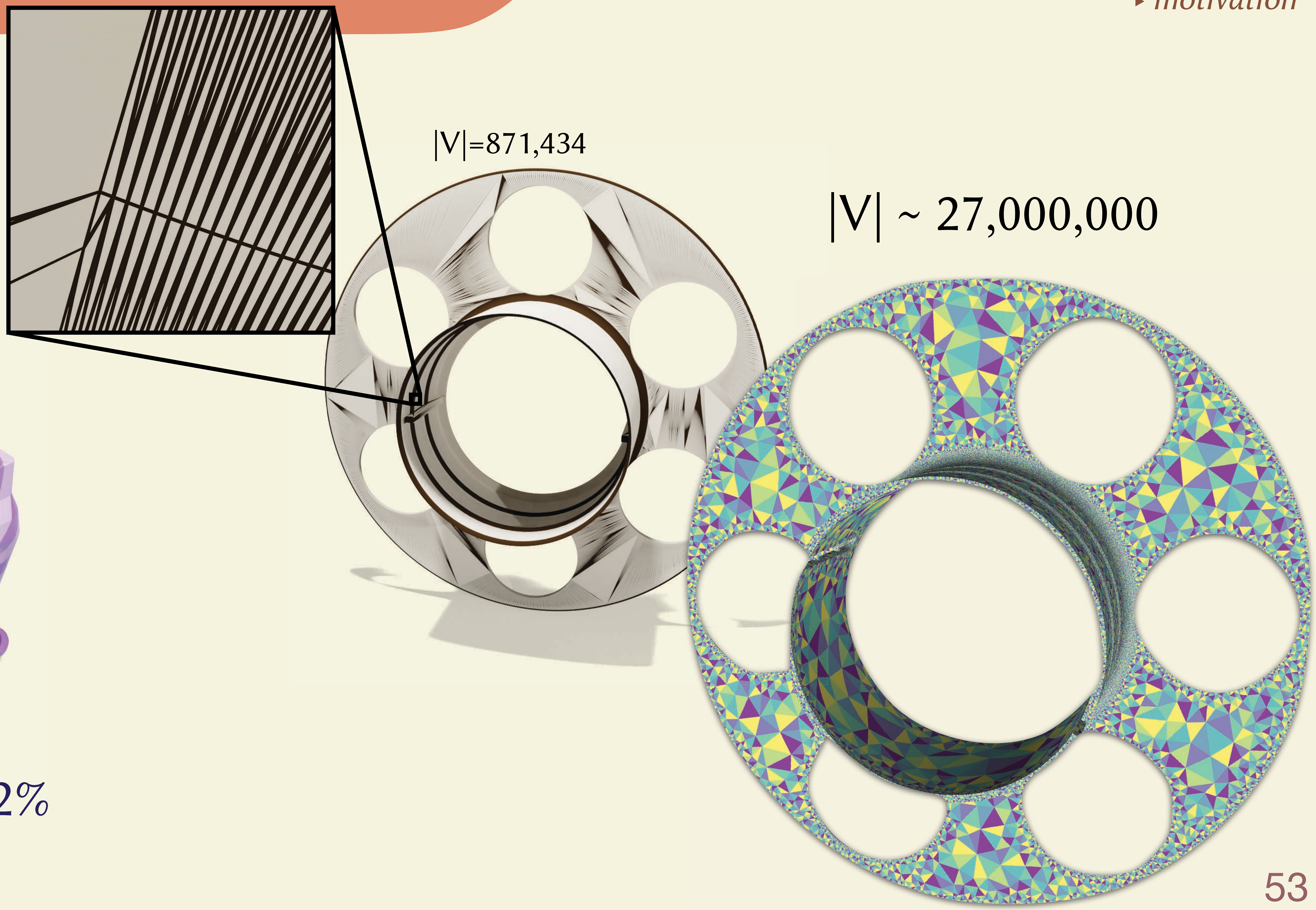


https://github.com/MarkGillespie/intrinsic-triangulations-demo

III. Simplifying Intrinsic Triangulations

[Liu, **G.**, Chislett, Sharp, Jacobson, & Crane. 2023. Surface Simplification using Intrinsic Error Metrics. *ACM Transactions on Graphics*]

Exact geometry preservation: a blessing and a curse



Coarse meshes can be perfectly adequate

Coarse meshes can be perfectly adequate

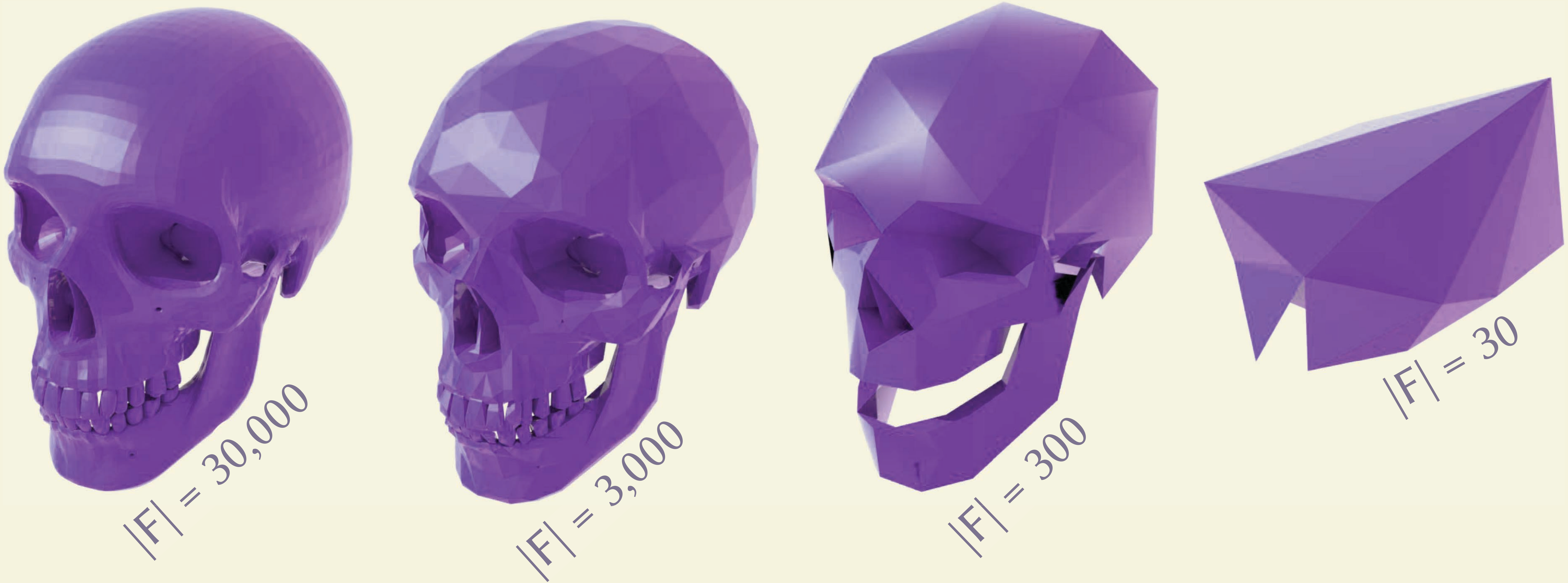
runtime:
23.14 s

runtime:
0.9 s

Near-identical, but 25x faster

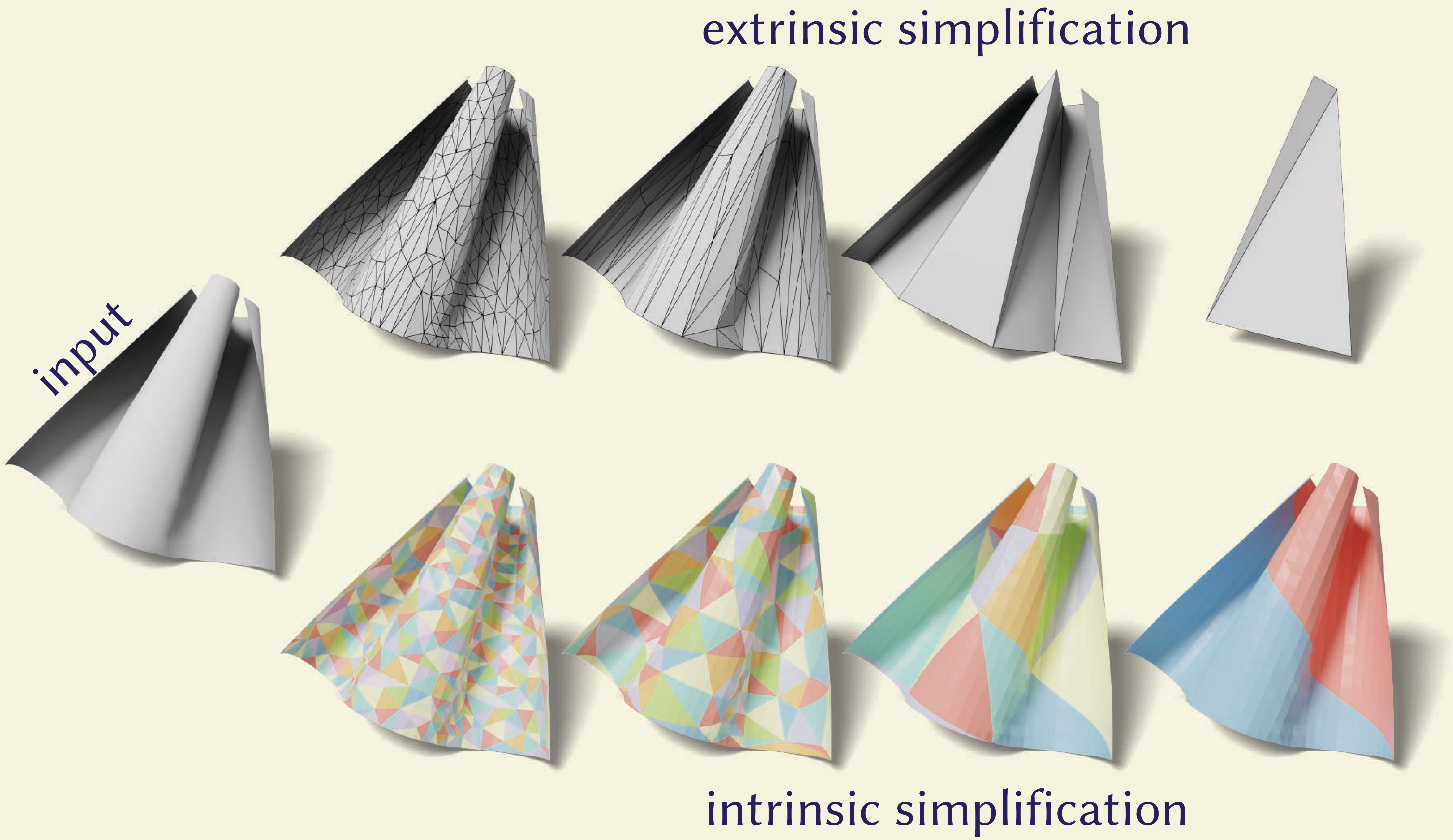
Traditional goal: *extrinsic* simplification

- Find a coarse mesh close in space to the original
    - Often designed to optimize for visual fidelity



Intrinsic problems benefit from intrinsic simplification

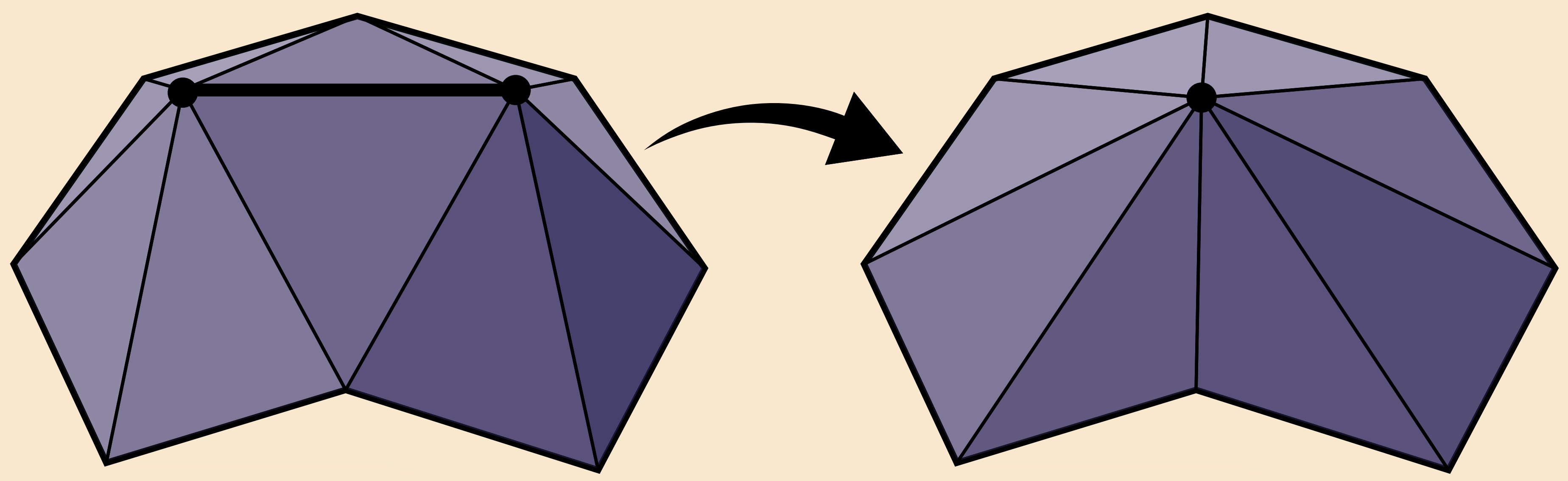
- Extrinsic methods preserve irrelevant extrinsic details
- Intrinsic approach opens up a larger space of triangulations
- Extreme example: near-developable surfaces



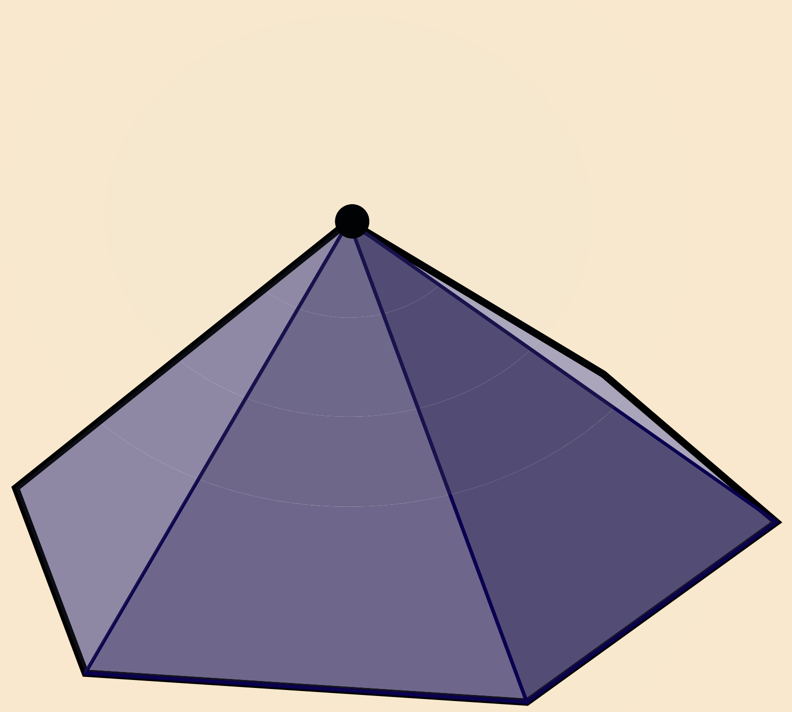
Inspiration: quadric error simplification

[Garland & Heckbert 1997]

1. Local simplification operation



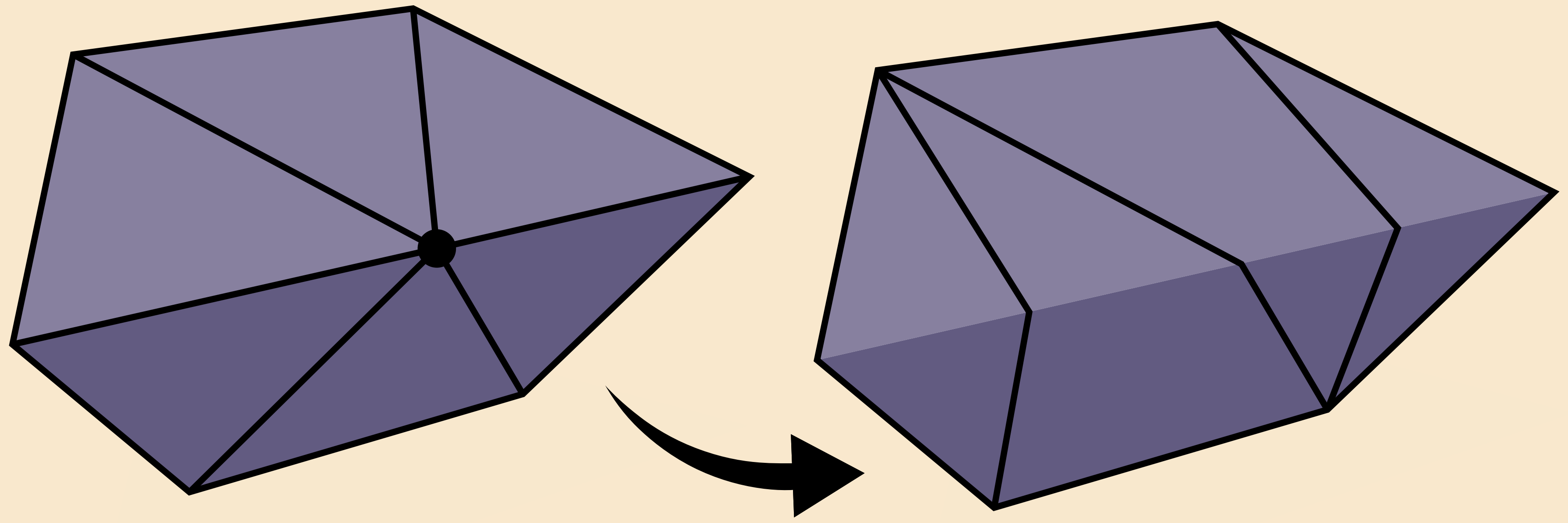
2. Accumulated distortion measurements



- Algorithm: repeatedly collapse cheapest edge
  - Efficient: all local operations
  - Accurate: accumulates error estimates

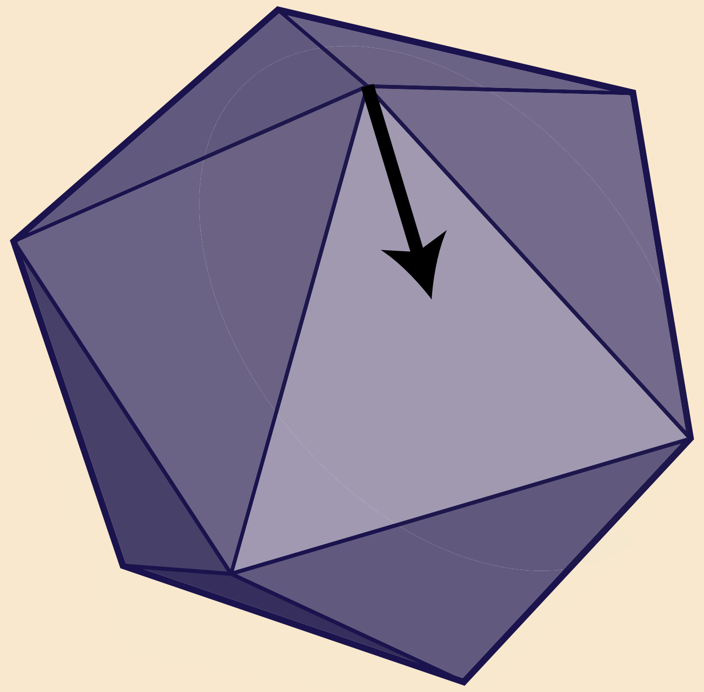
Intrinsic simplification

1. Local simplification operation



intrinsic vertex removal

2. Accumulated distortion measurements

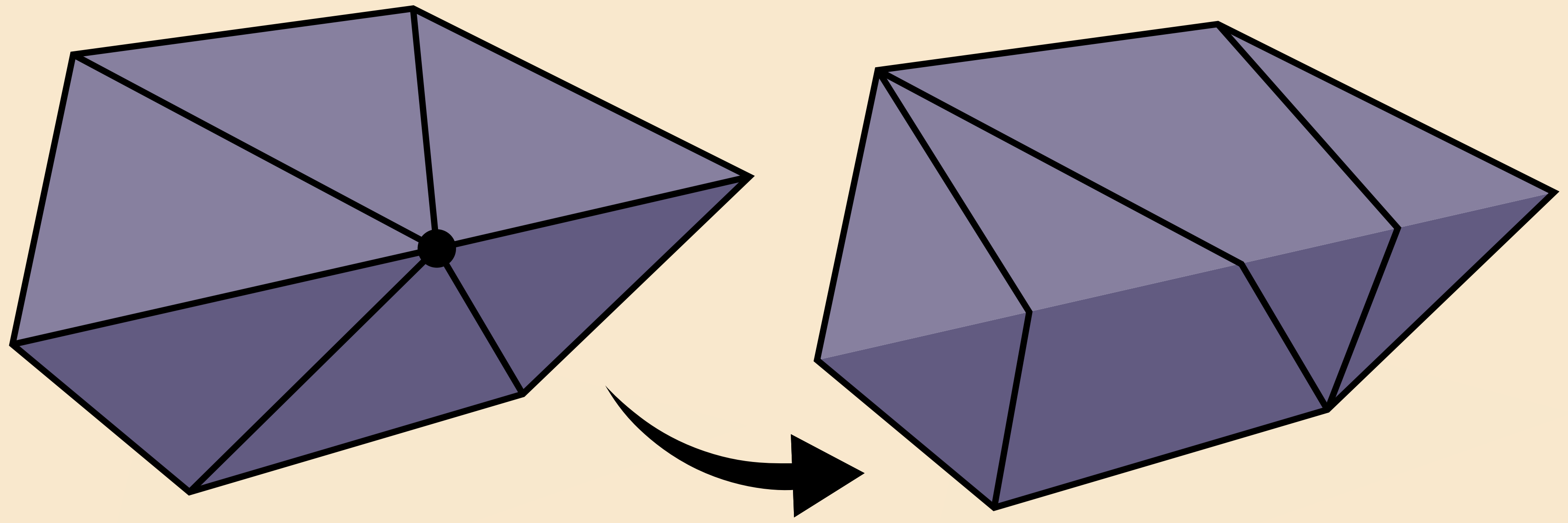


intrinsic curvature error

- Algorithm: repeatedly remove cheapest vertex

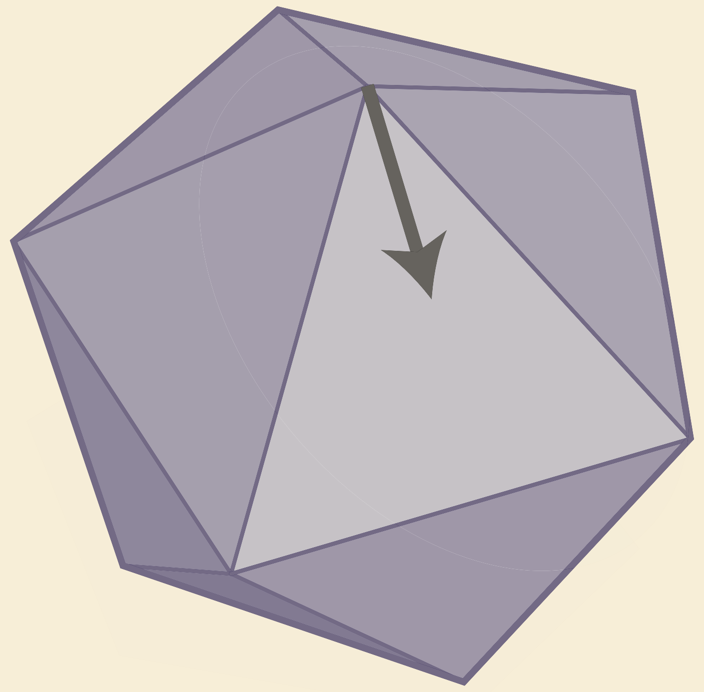
Intrinsic simplification

1. Local simplification operation



intrinsic vertex removal

2. Accumulated distortion measurements

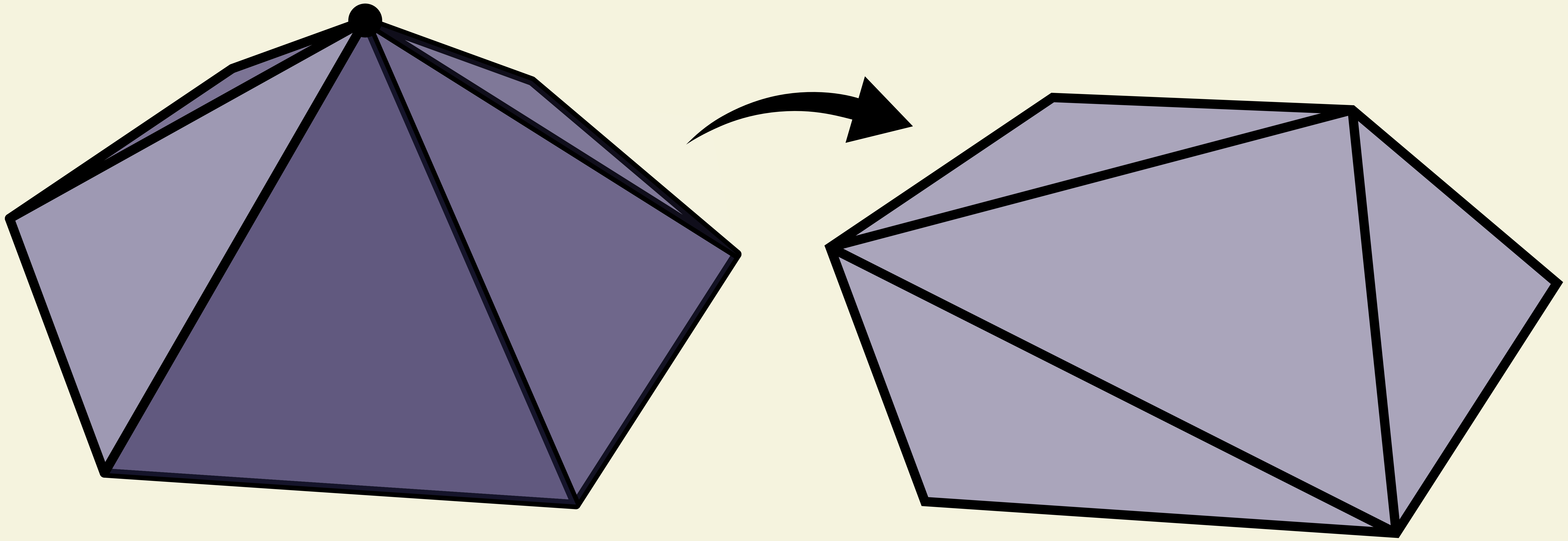


intrinsic curvature error

- Algorithm: repeatedly remove cheapest vertex

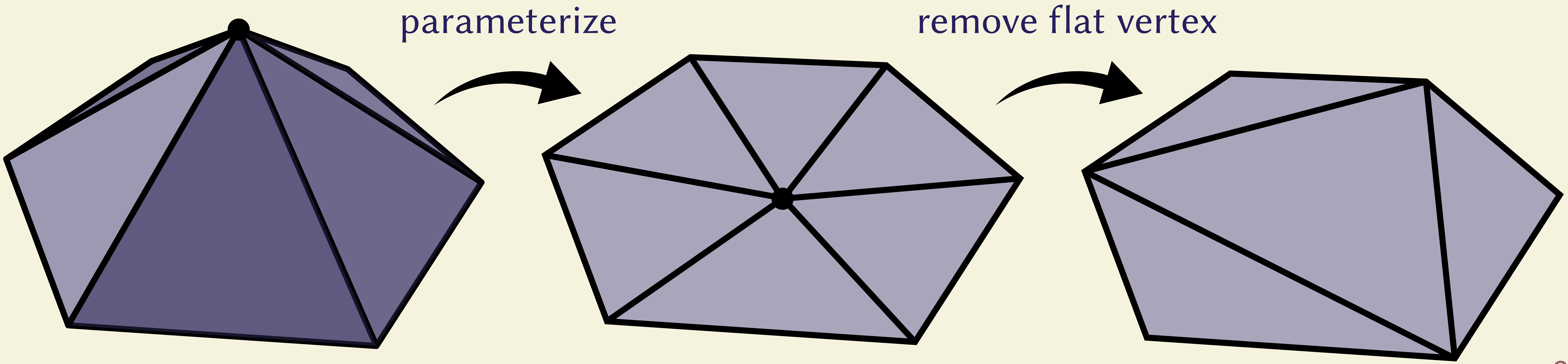
Intrinsic vertex removal

- Intrinsic view: replace curved vertex with flat patch
  - *i.e.*, parameterization problem

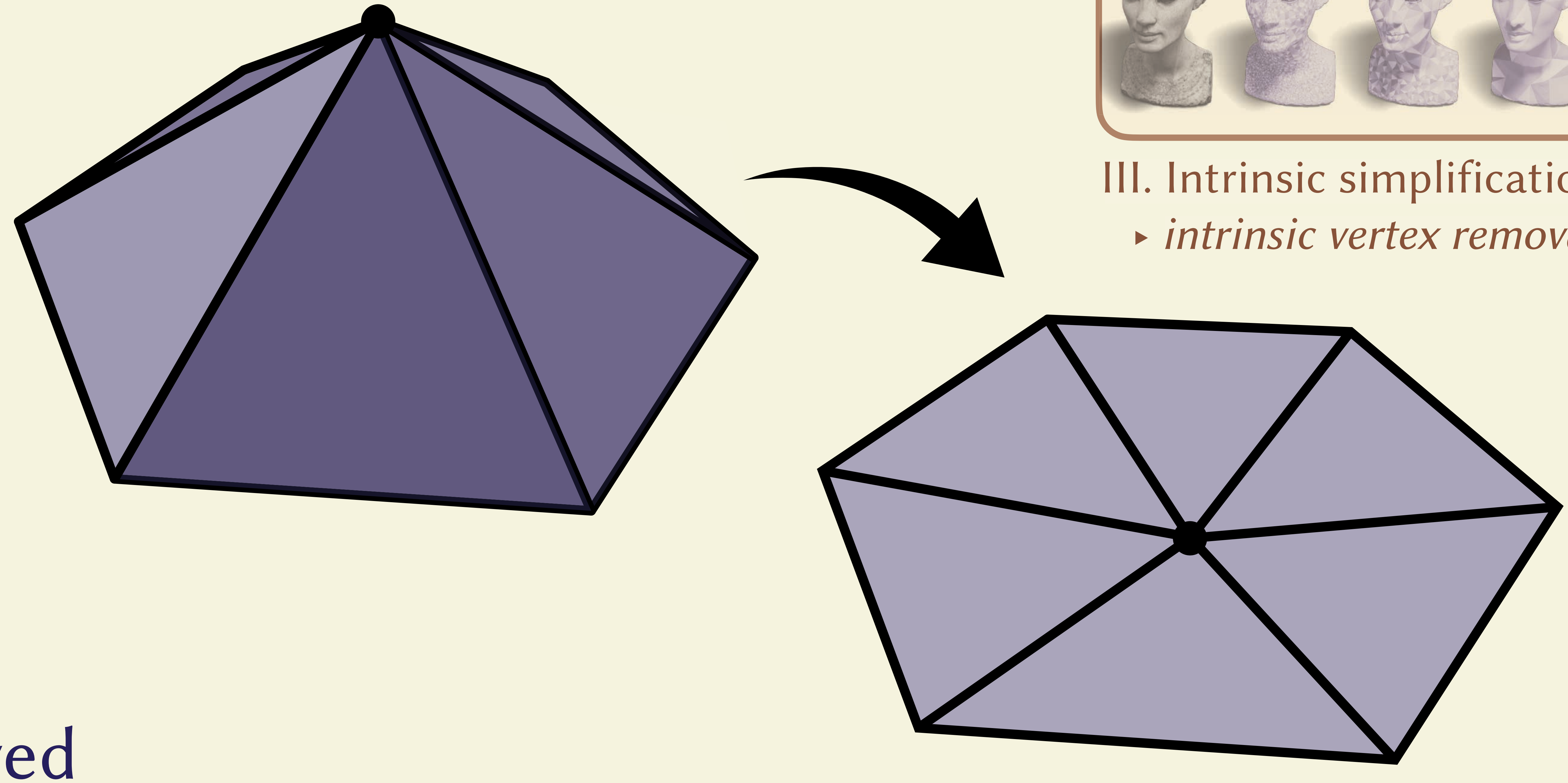


Intrinsic vertex removal

- Intrinsic view: replace curved vertex with flat patch
  - *i.e.*, parameterization problem



Vertex flattening



- Map 1-ring to plane such that:

  (1) Distortion is low

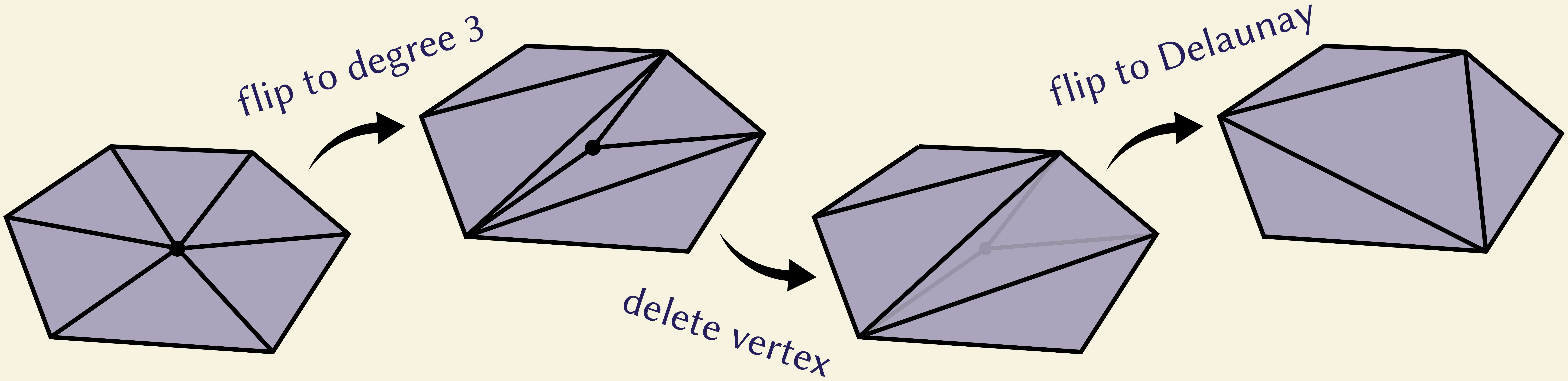
  (2) Boundary edge lengths are preserved

- Discrete conformal map [Springborn, Schröder & Pinkall 2008]
  - Fix $u = 0$ on boundary
  - Efficient 1D optimization problem

(If parameterization fails, pick a different vertex to remove)

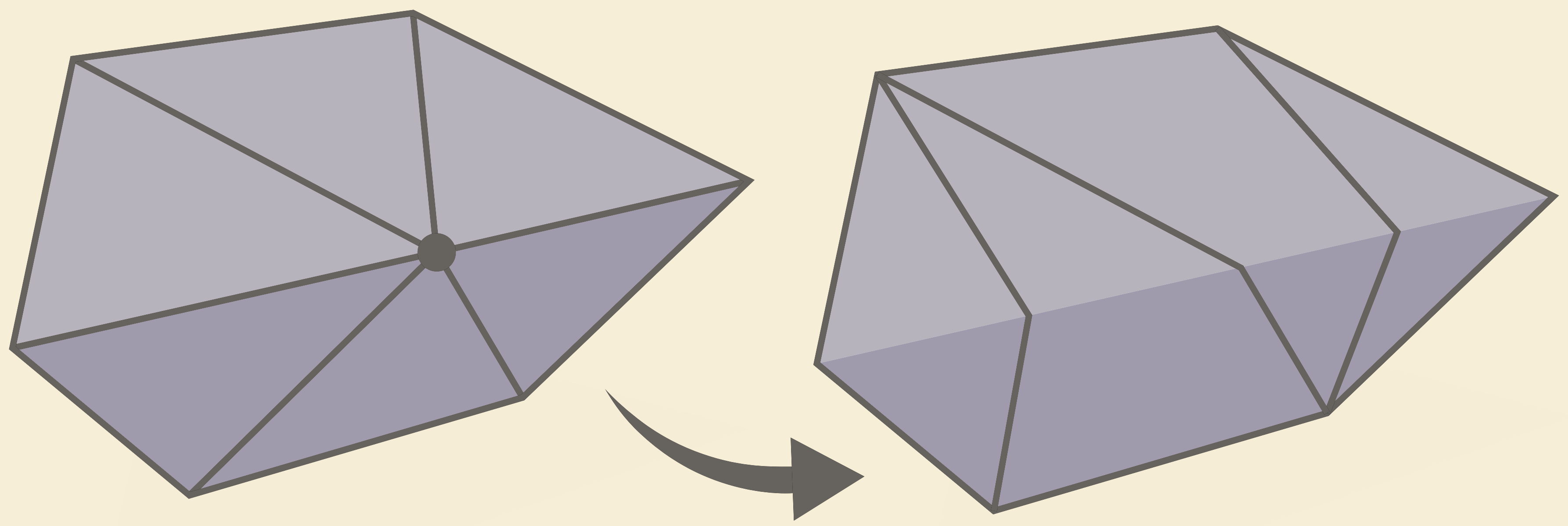
Flat vertex removal

- Same as before



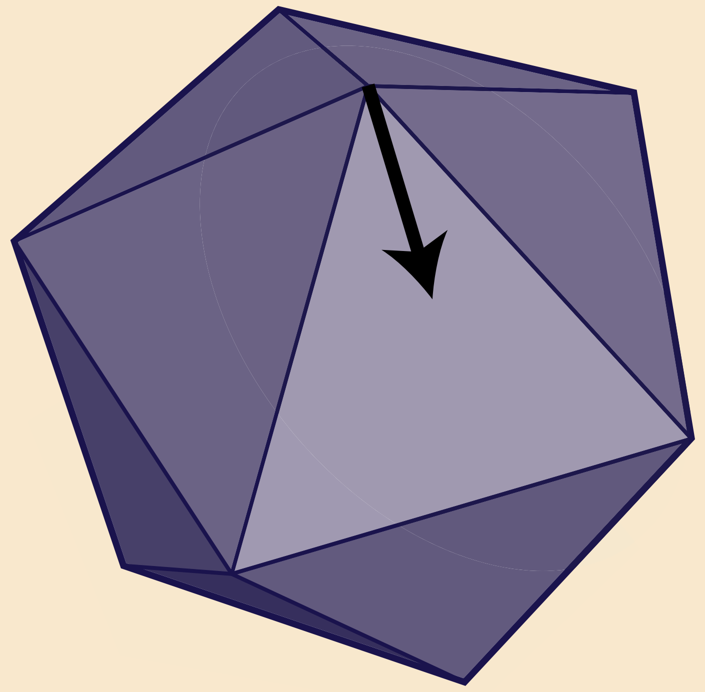
Intrinsic simplification

1. Local simplification operation



intrinsic vertex removal

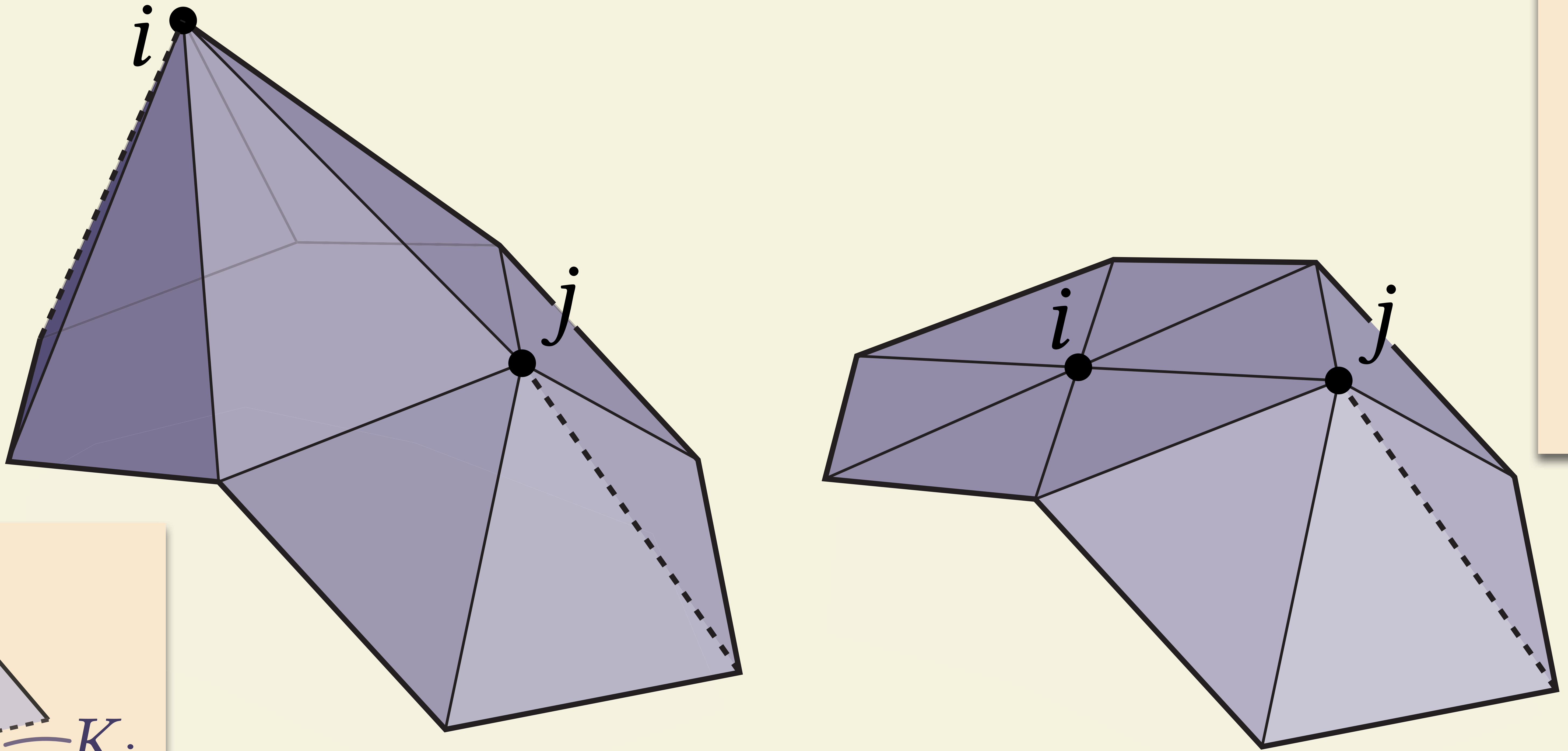
2. Accumulated distortion measurements

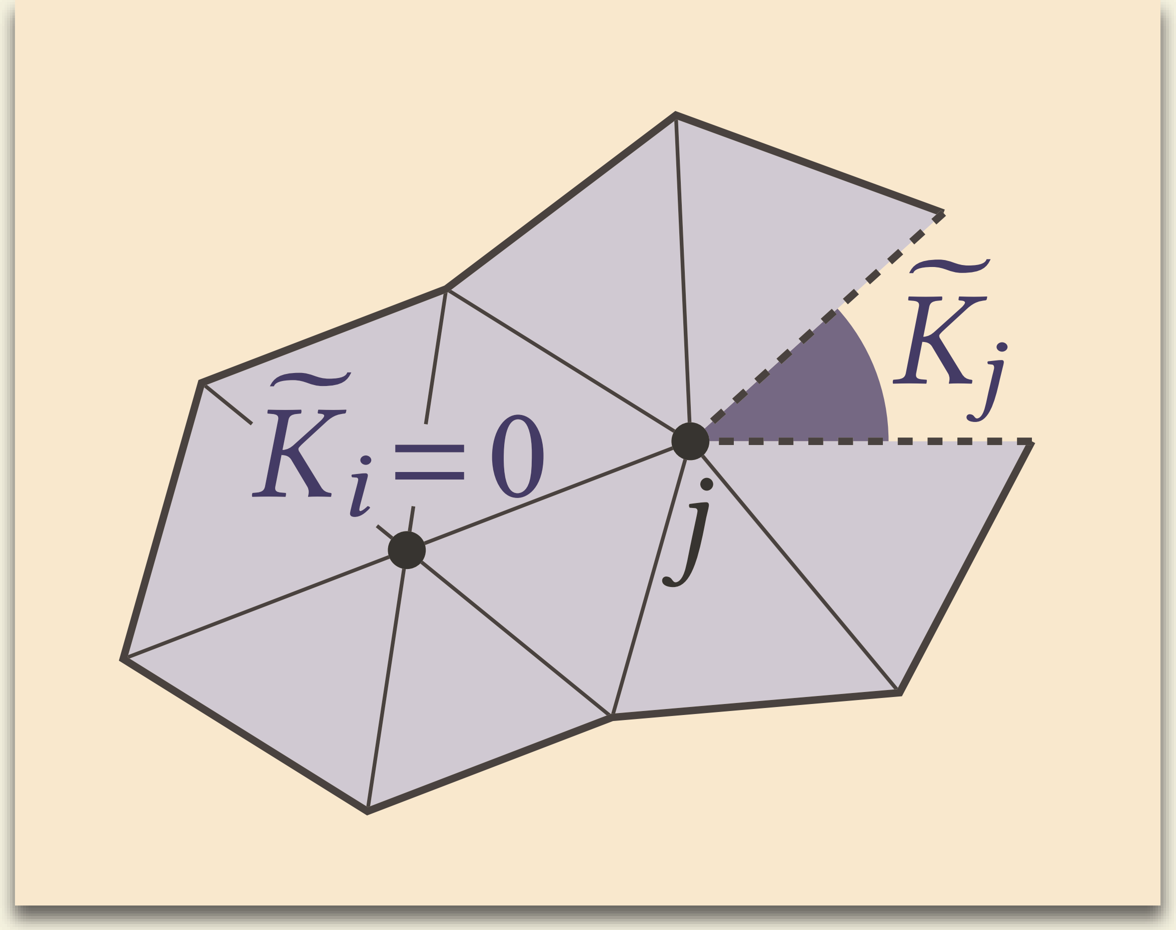


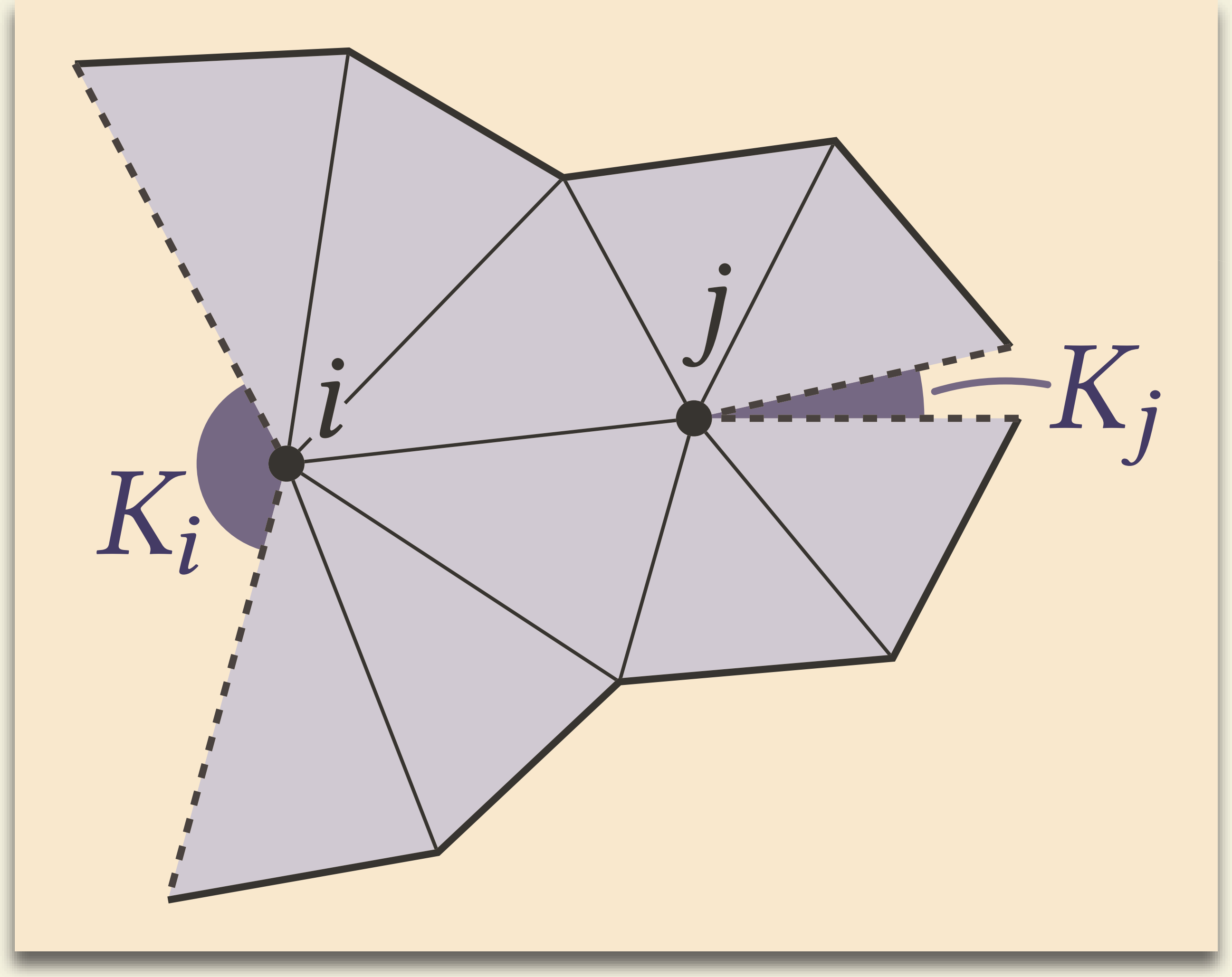
intrinsic curvature error

- Algorithm: repeatedly remove cheapest vertex

Distortion: curvature redistribution

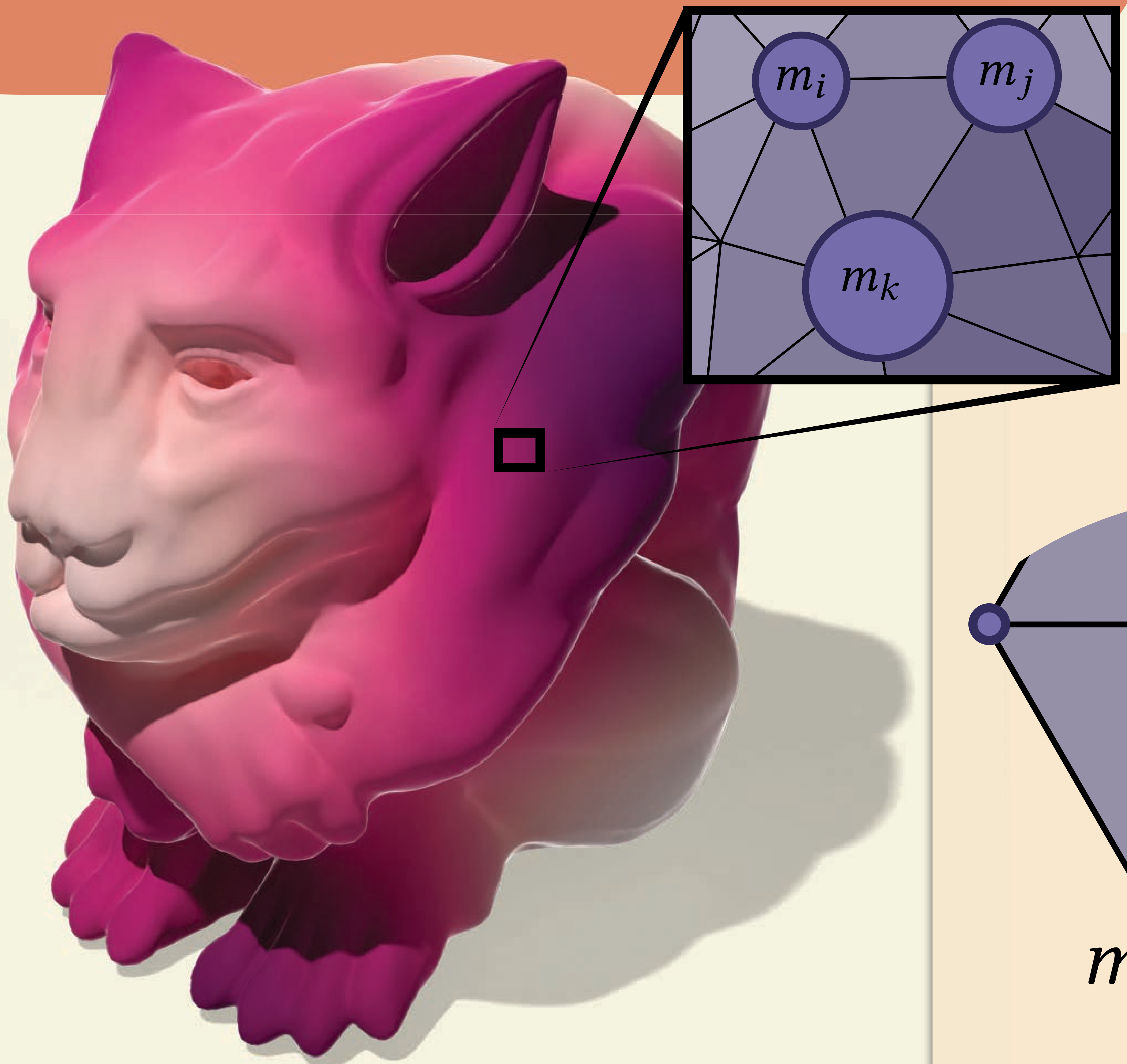






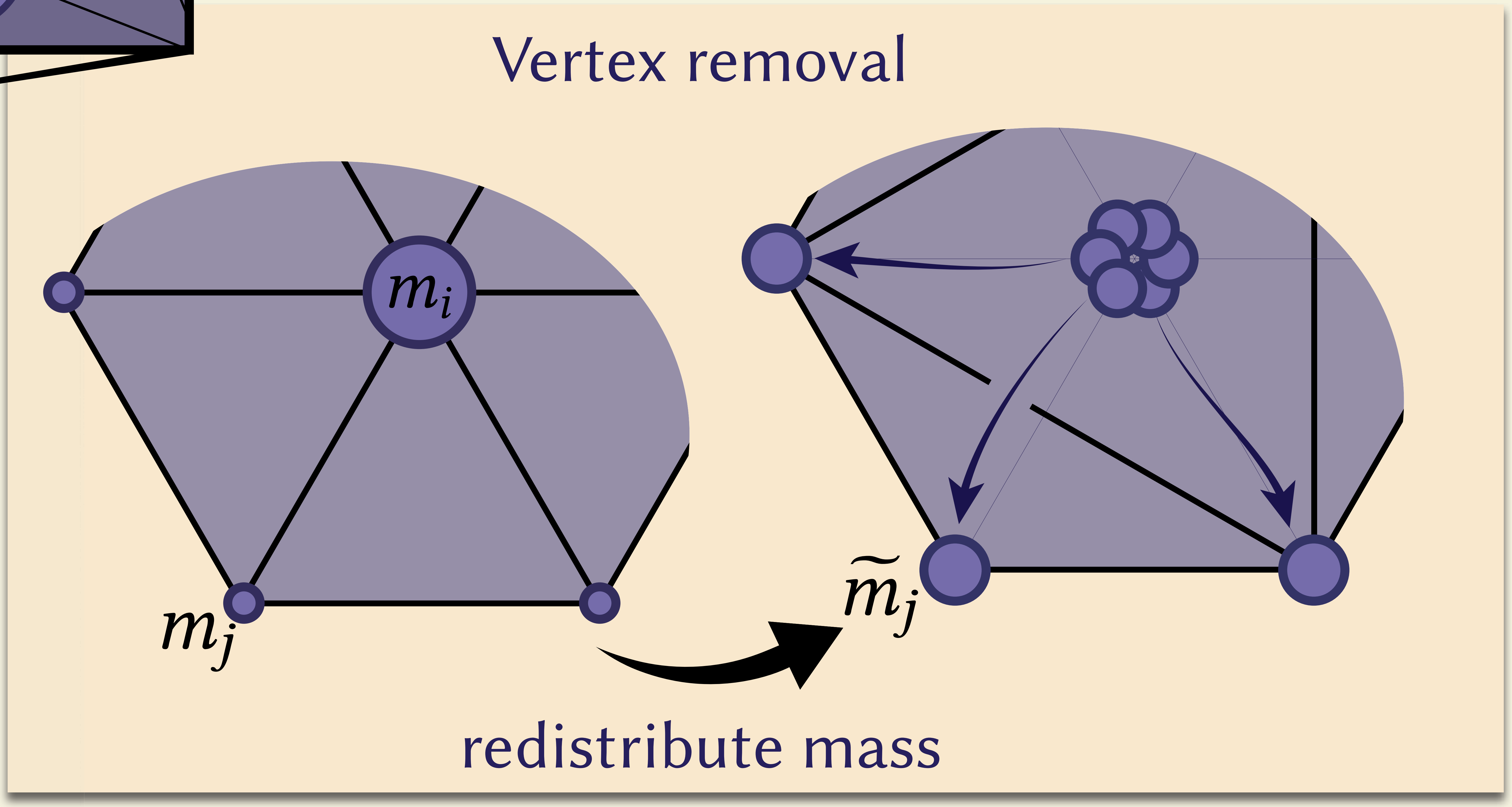
We approximate the *transport cost* of this curvature redistribution

Mass transport cost



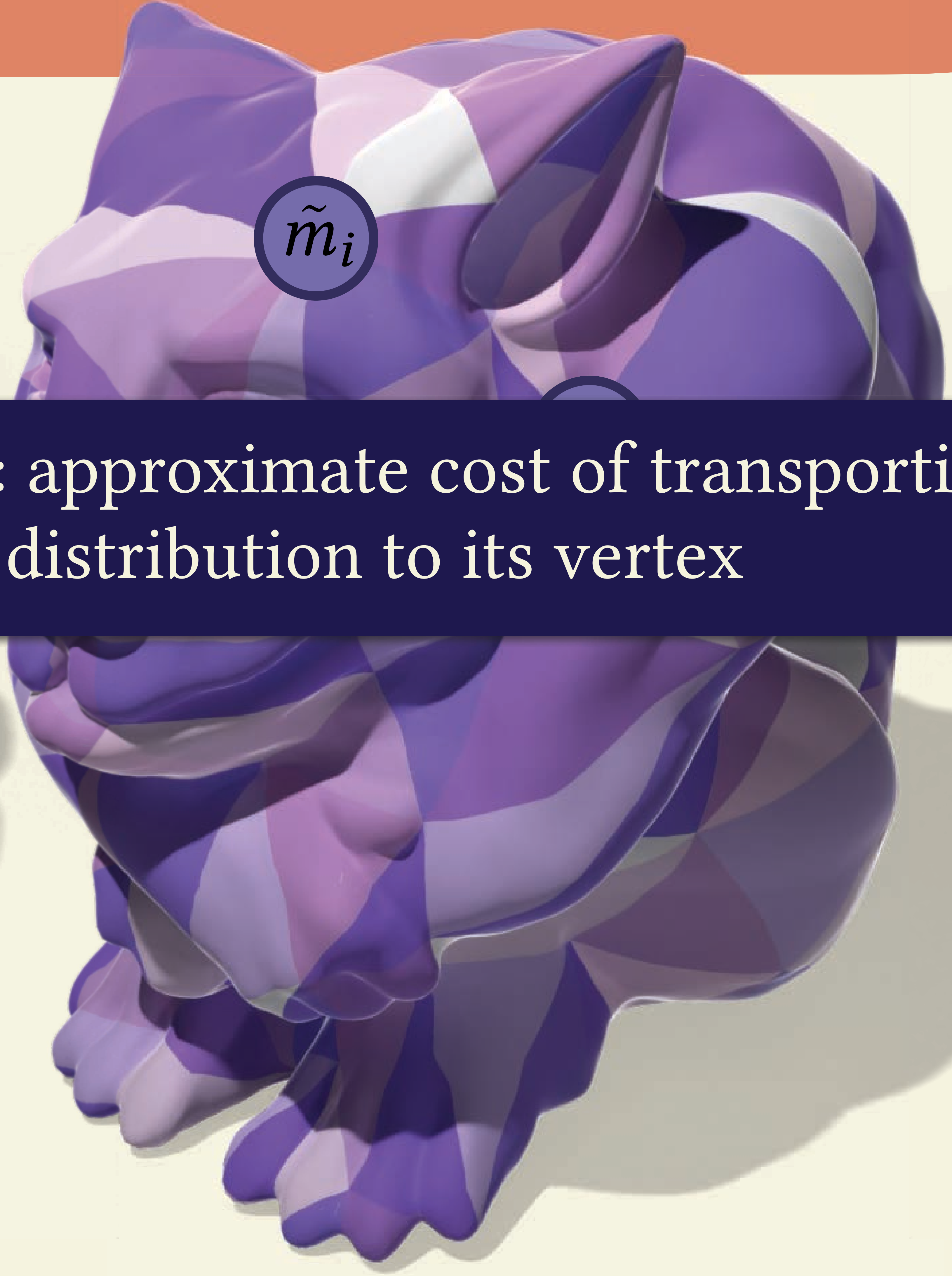
nonnegative mass
distribution $m : V \to \mathbb{R}_{\geq 0}$

III. Intrinsic simplification
▸ *intrinsic curvature error*



Comparing mass distributions

III. Intrinsic simplification
▸ *intrinsic curvature error*

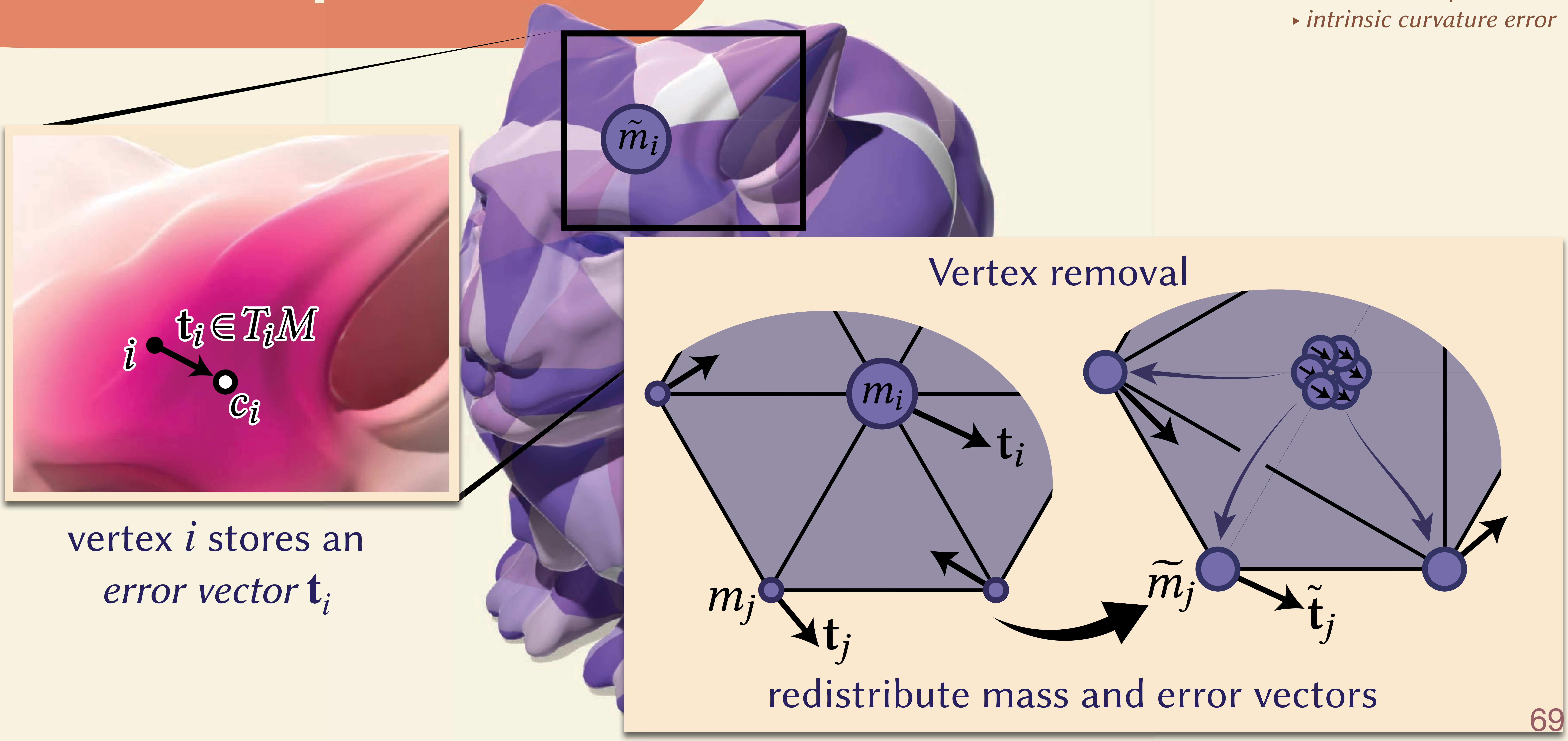


Goal: approximate cost of transporting each distribution to its vertex

mass distribution transported to vertex i

mass distribution transported to vertex j

Approximating the mass transport cost

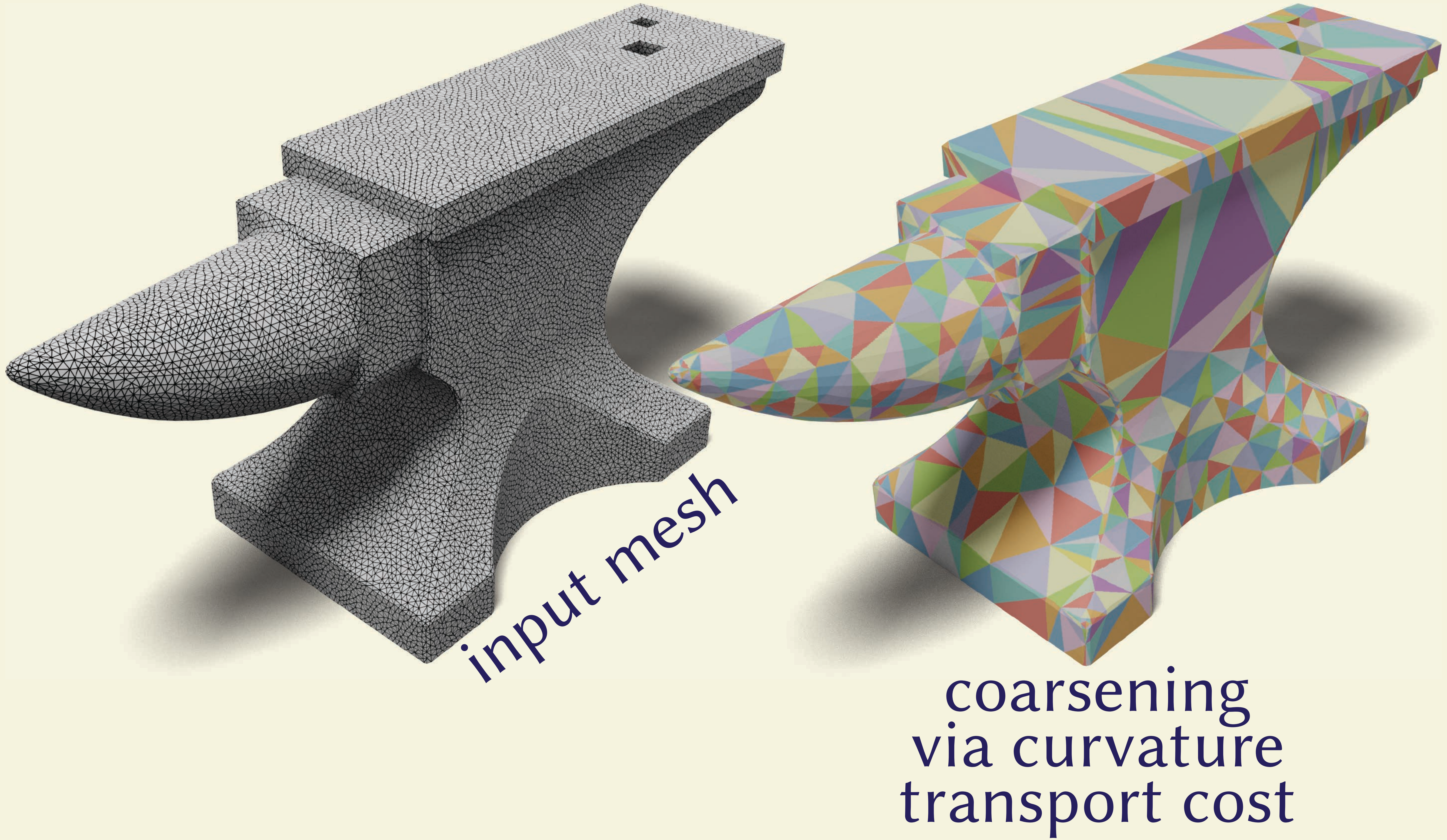


vertex i stores an
error vector $\mathbf{t}_i$

Specializing to curvature

- Challenge: curvature is signed
  - Just track positive and negative parts separately

Simplification with the curvature transport cost



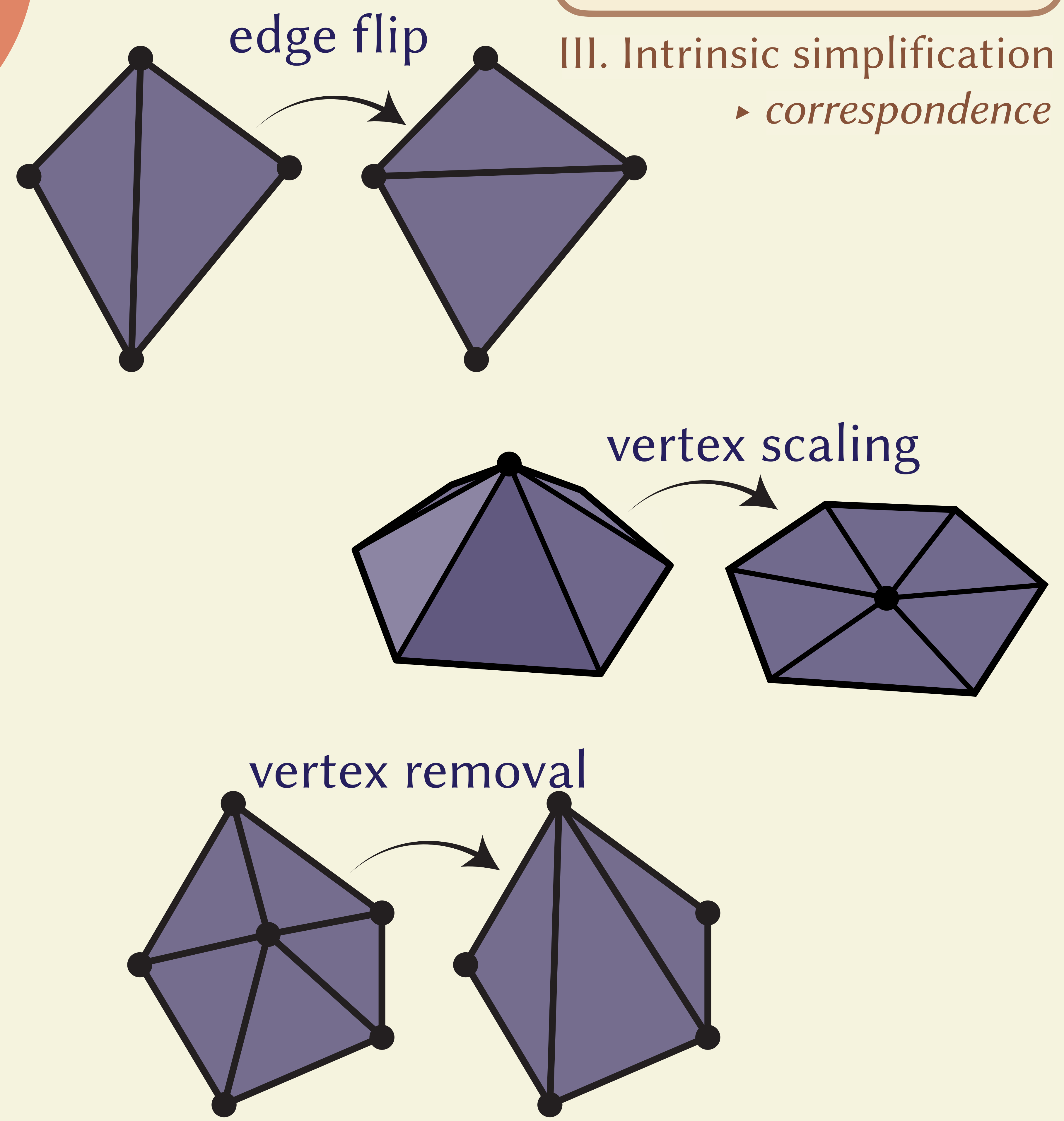
Other transport costs

- Track transport of other data (*e.g.* area) in same way
  - Can take weighted combinations of costs

Surface correspondence

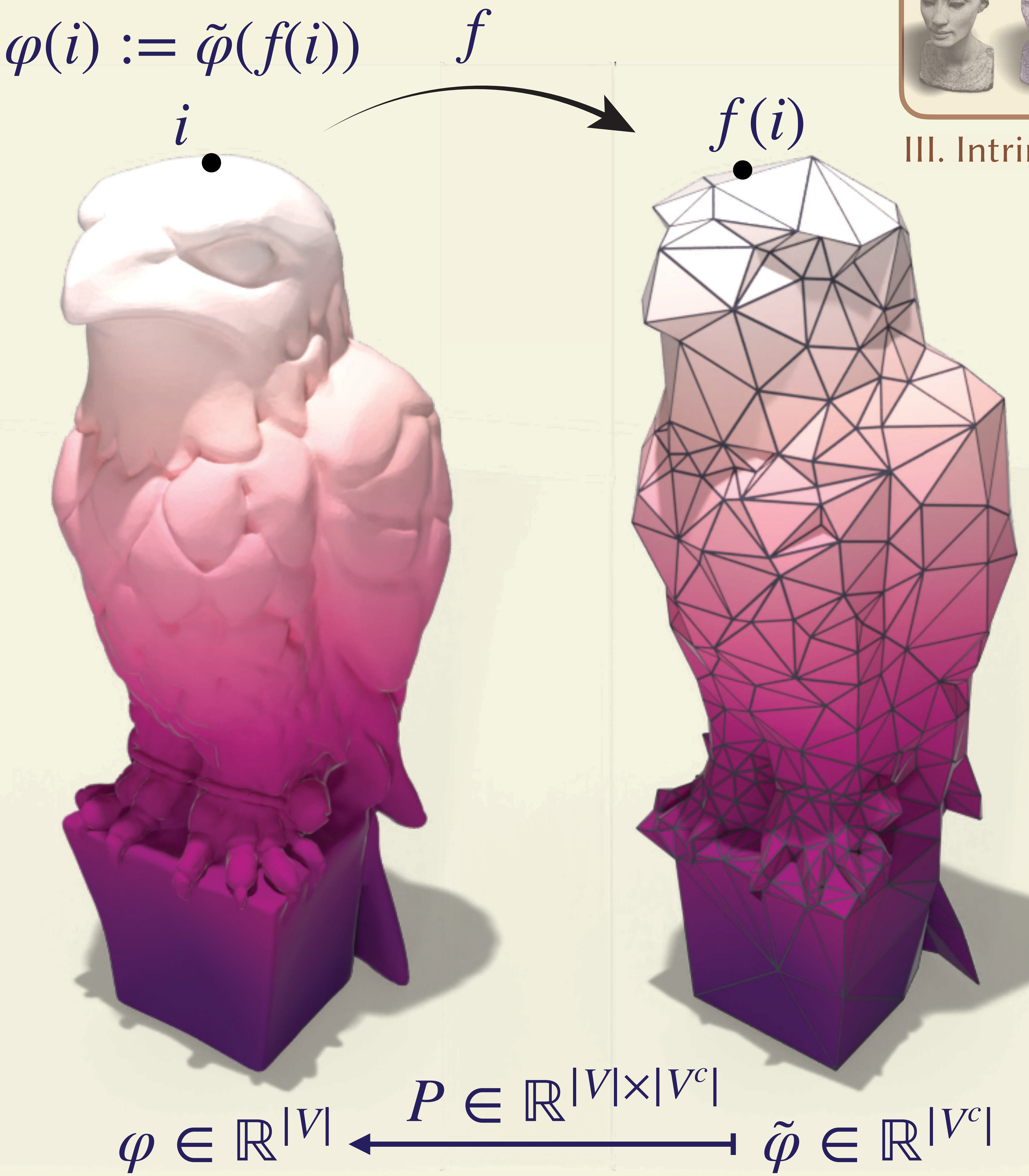
- Simplified mesh <u>*not*</u> isometric to original surface
  - Breaks existing data structures
- But, only uses a few local operations
  - Correspondence easy for each operation
- Encode correspondence via list of operations

1. Flip edge 1
2. Scale vertex 5
3. Remove vertex 5
4. Flip edge 8
5. Flip edge 12
6. Scale vertex 2



Prolongation

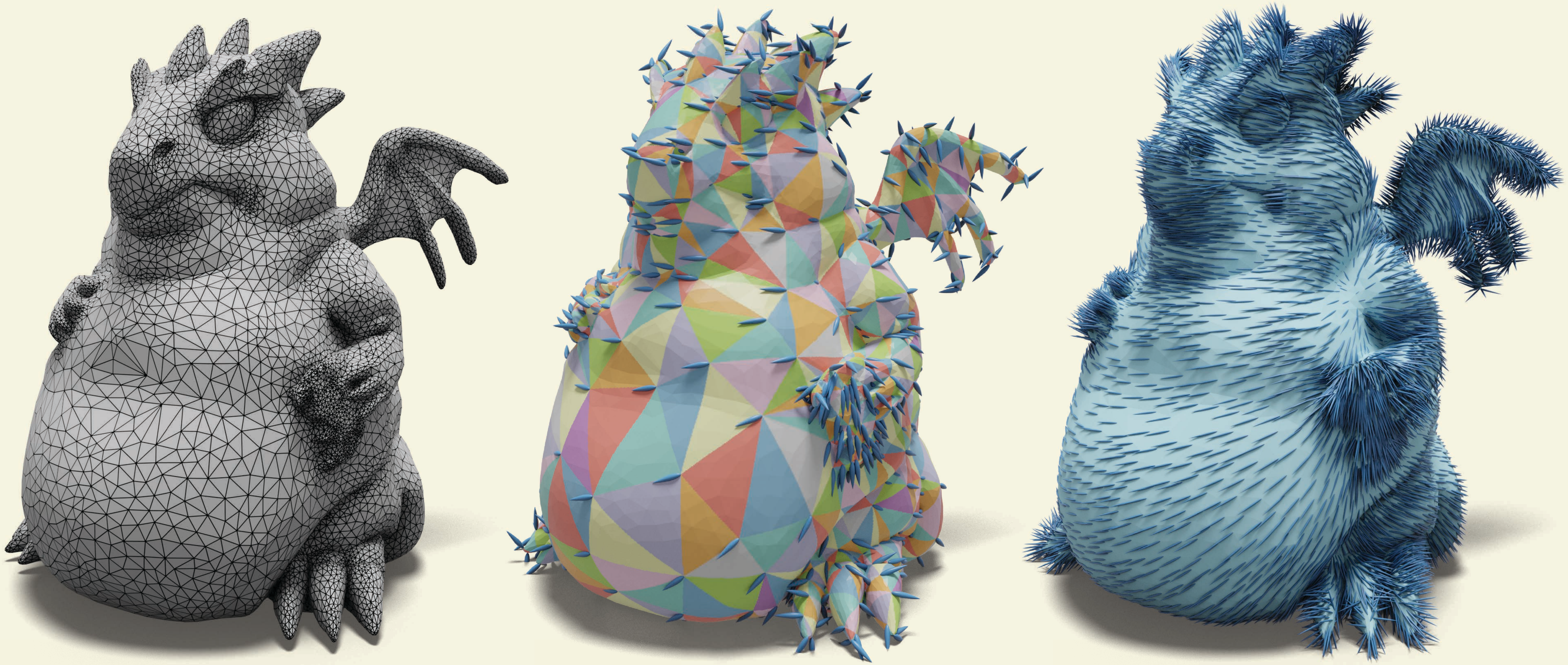
- Transfer piecewise-linear functions:
  - Just find values at vertices
  - Encode by a matrix



Pulling back vector fields

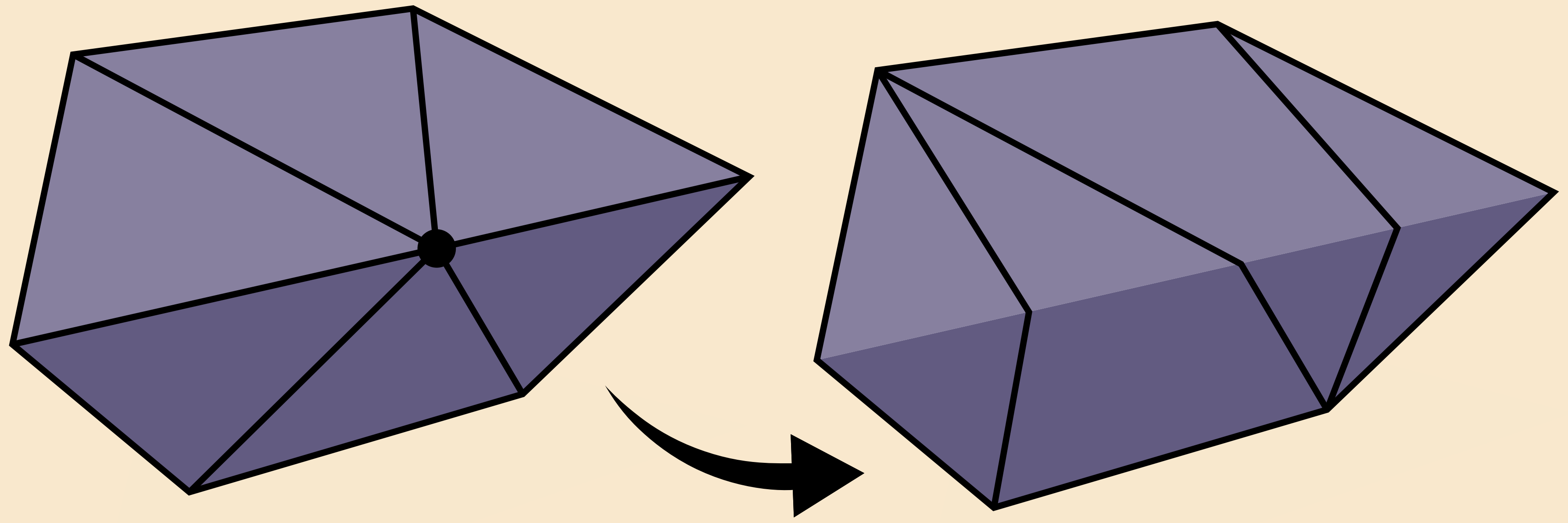
- Approximate differential of correspondence map

Encode by complex prolongation matrix



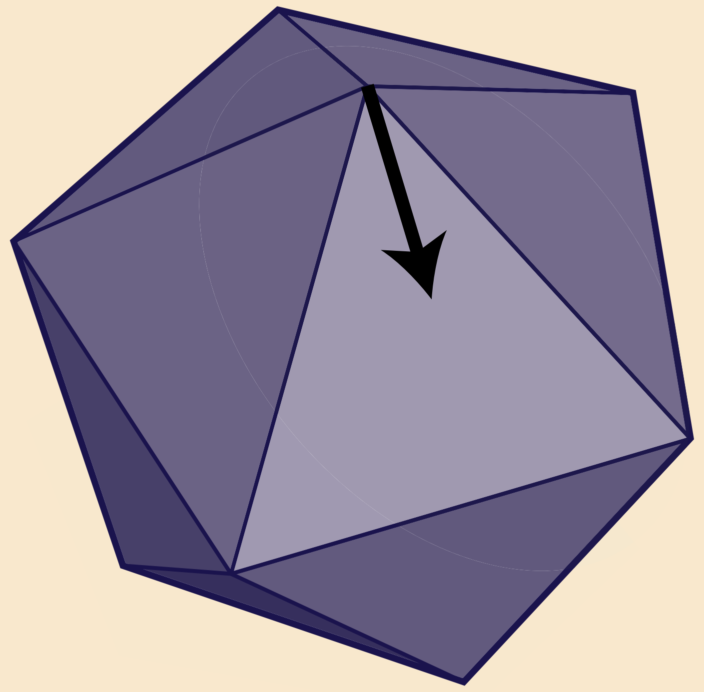
Intrinsic simplification—summary

1. Local simplification operation



intrinsic vertex removal

2. Accumulated distortion measurements



intrinsic curvature error

- Algorithm: repeatedly remove cheapest vertex
- Correspondence: record operation history

III. Intrinsic simplification

Results

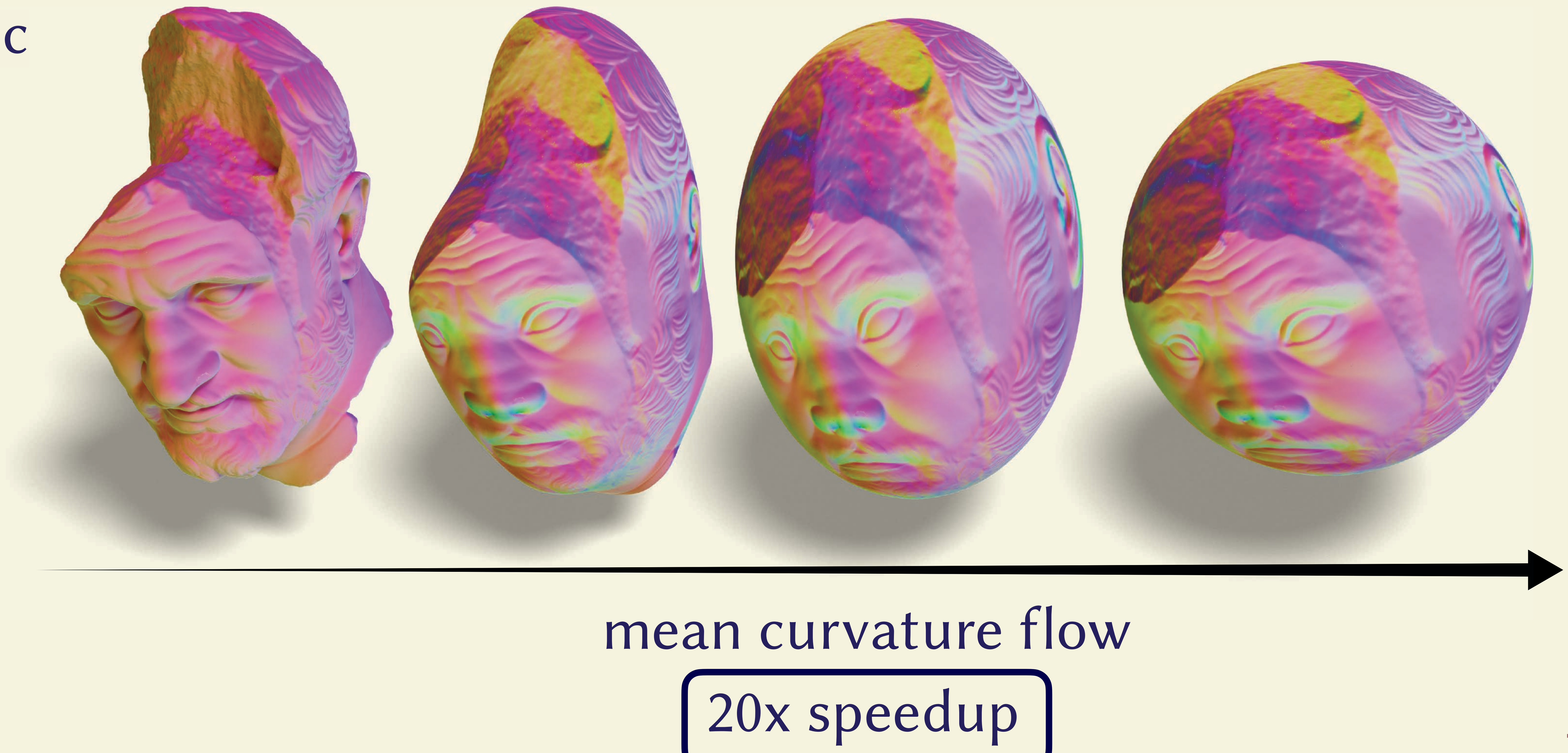
Surface hierarchies

III. Intrinsic simplification
▸ *results*

Hierarchies accelerate computation

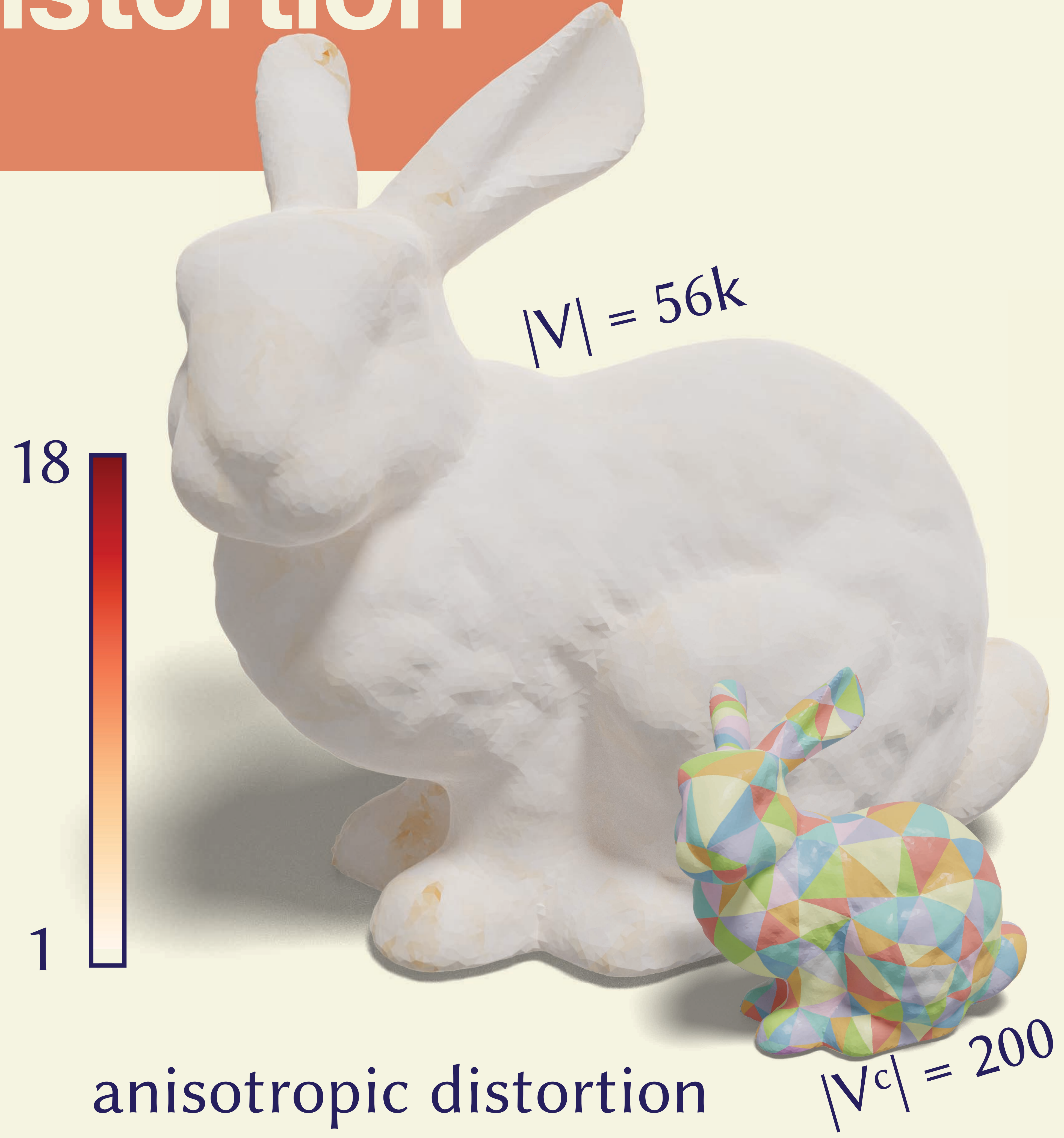
III. Intrinsic simplification
‣ *results*

- Accelerate many geometric tasks
  - Even helps with extrinsic problems

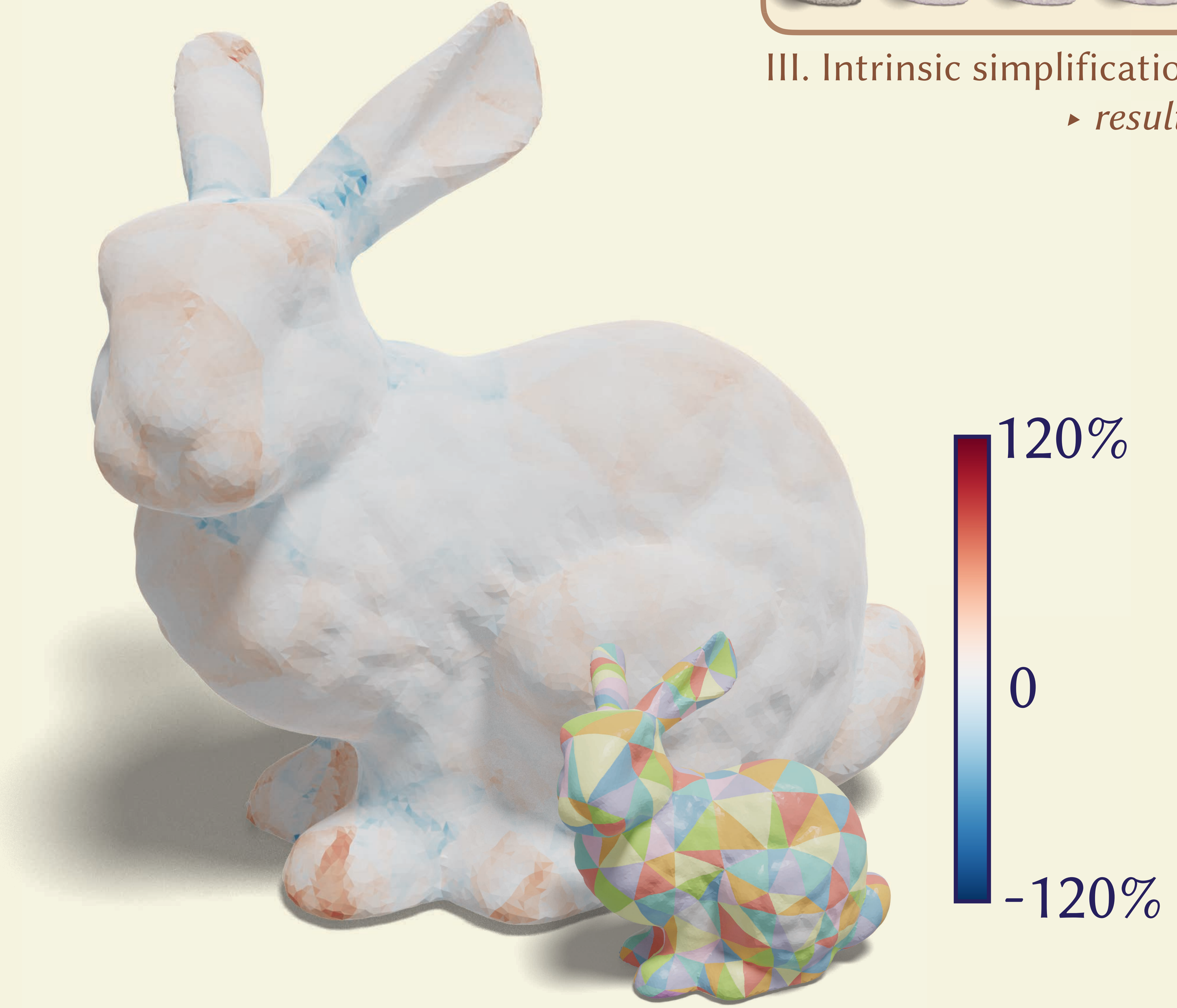


Distortion

III. Intrinsic simplification
▸ *results*



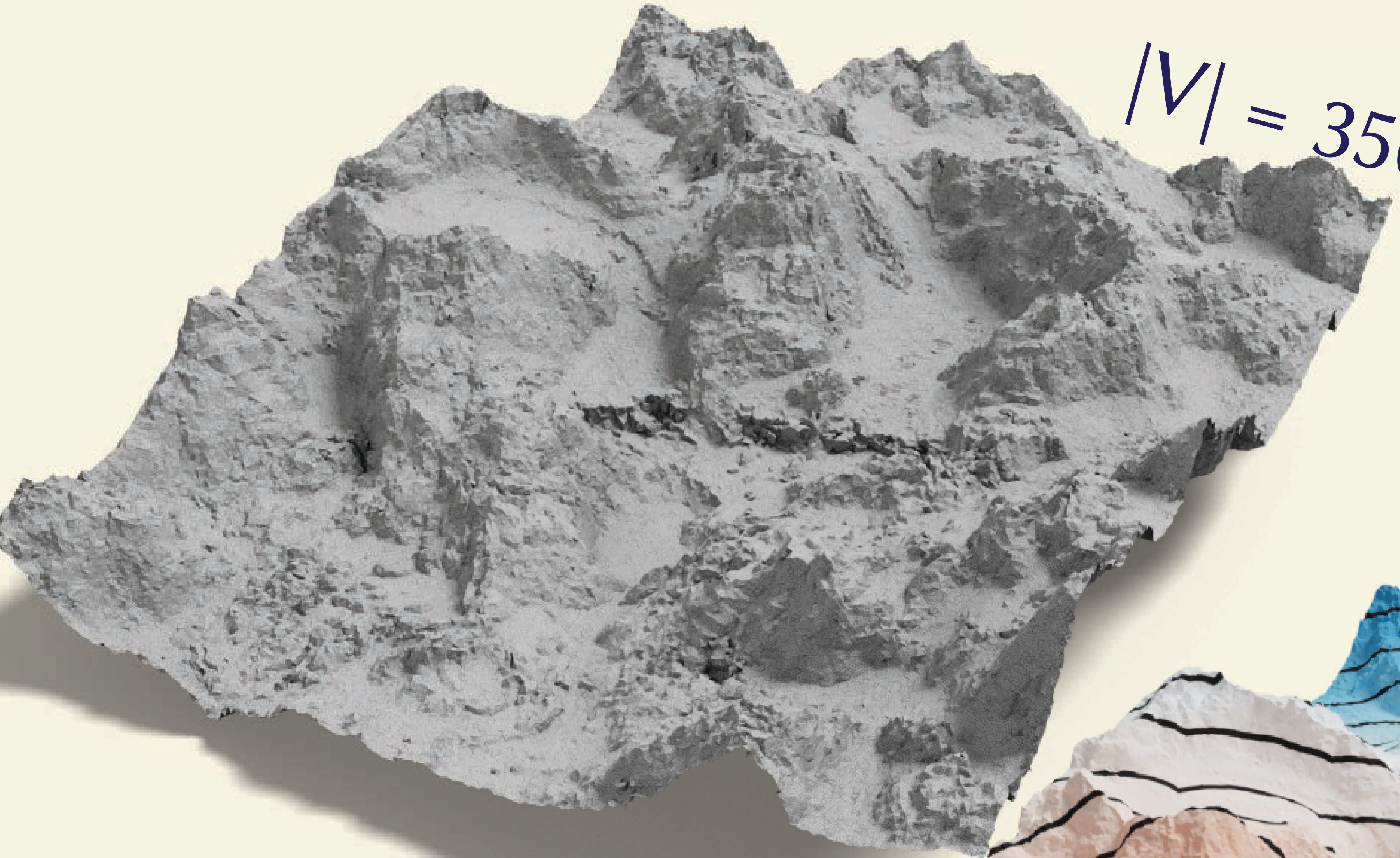
anisotropic distortion
i.e. quasiconformal dilatation
(mean 1.115)



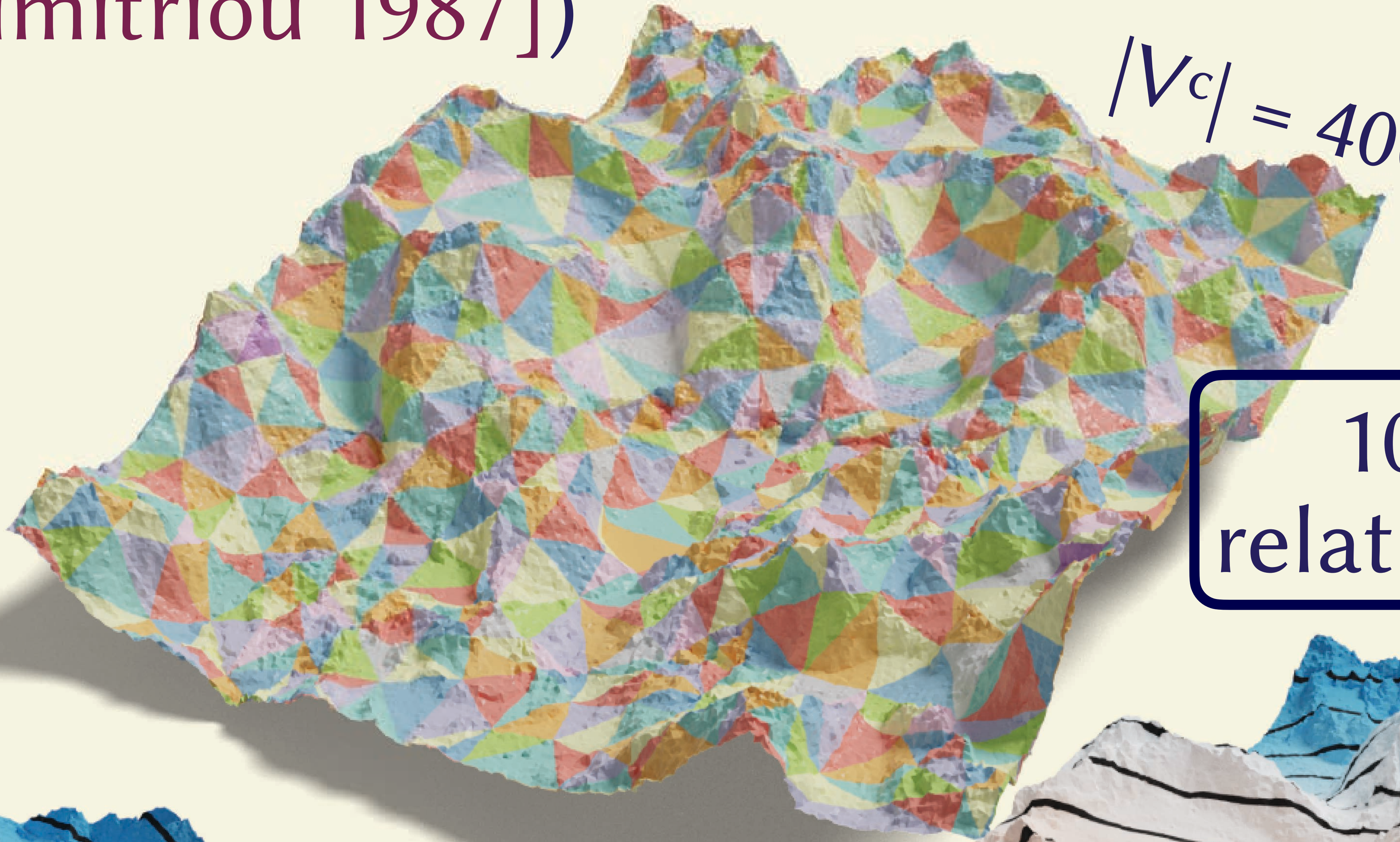
area distortion
(mean 8.1%)

Geodesic distance

(Computed via [Mitchell, Mount & Papadimitriou 1987])

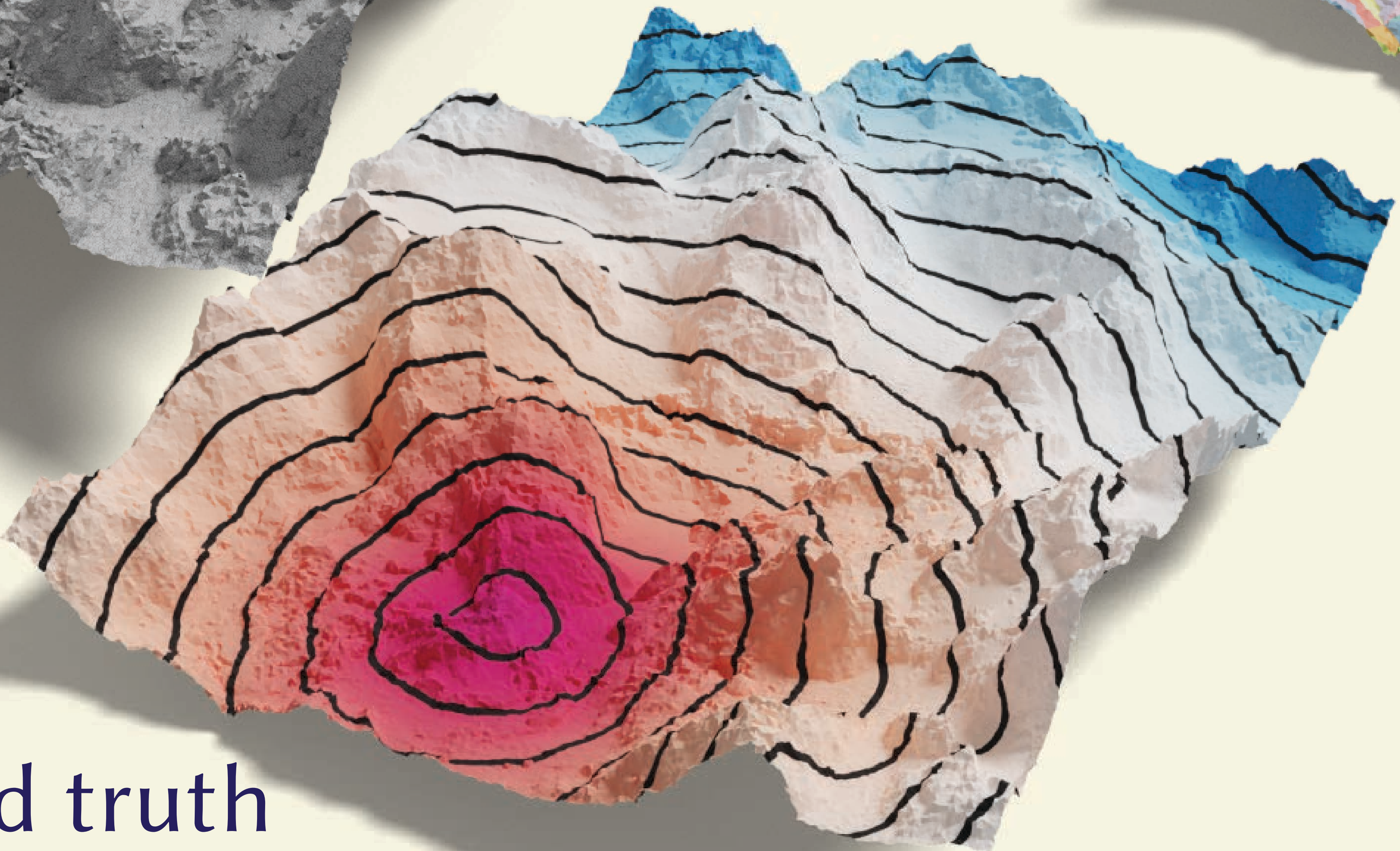


$|V| = 350k$

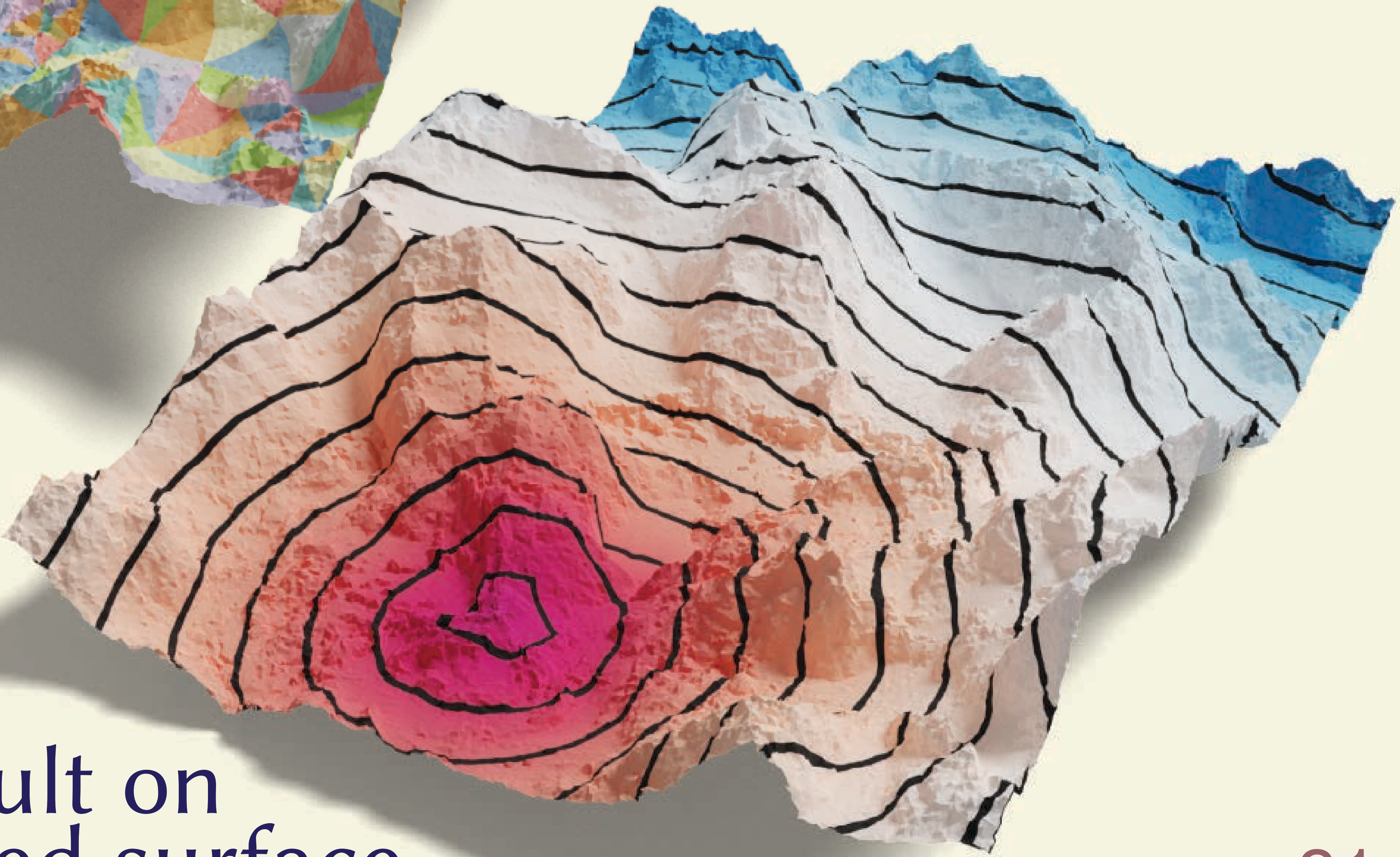


$|V^c| = 400$

1000x smaller
relative error: 0.03%



ground truth

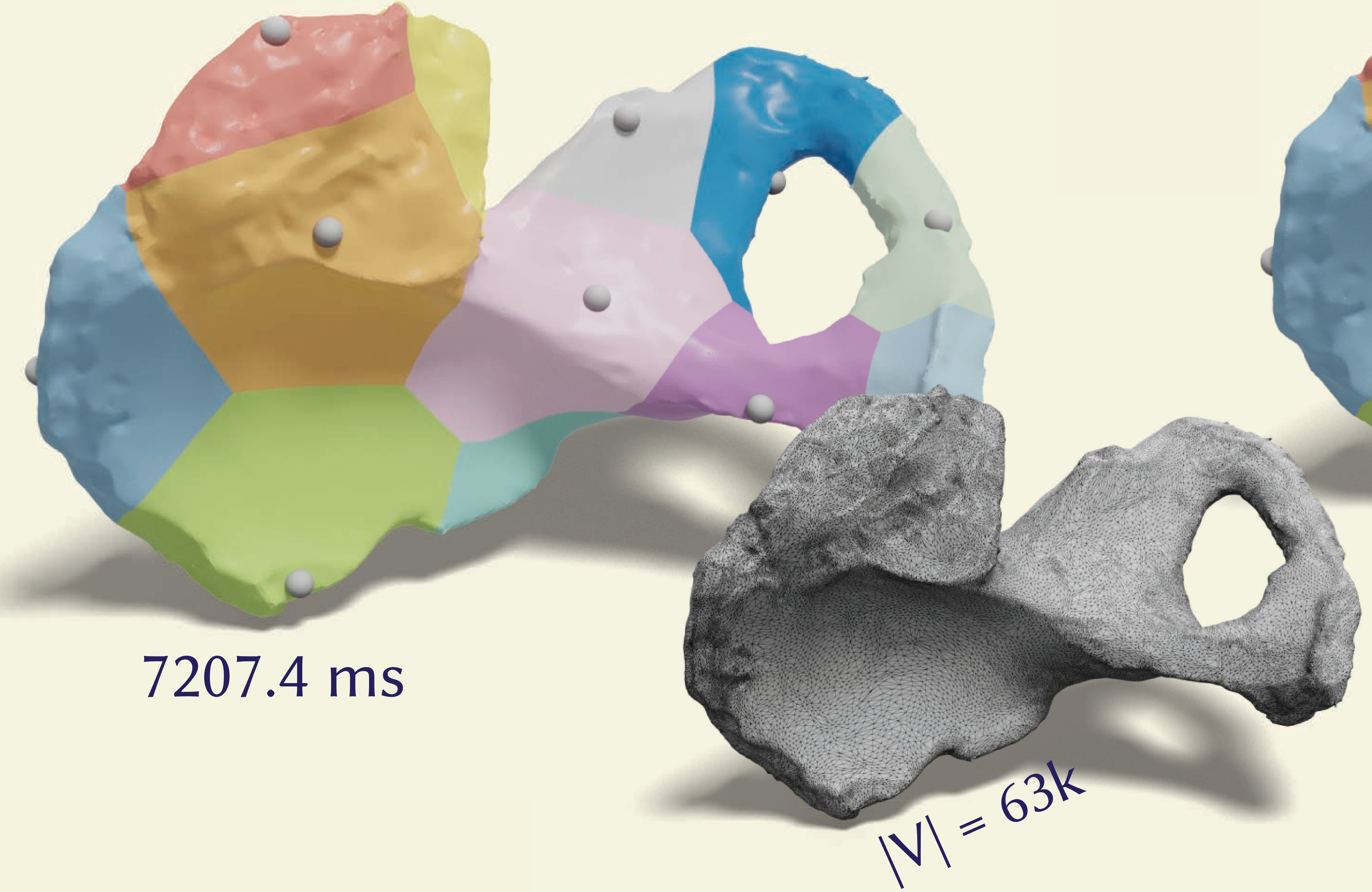


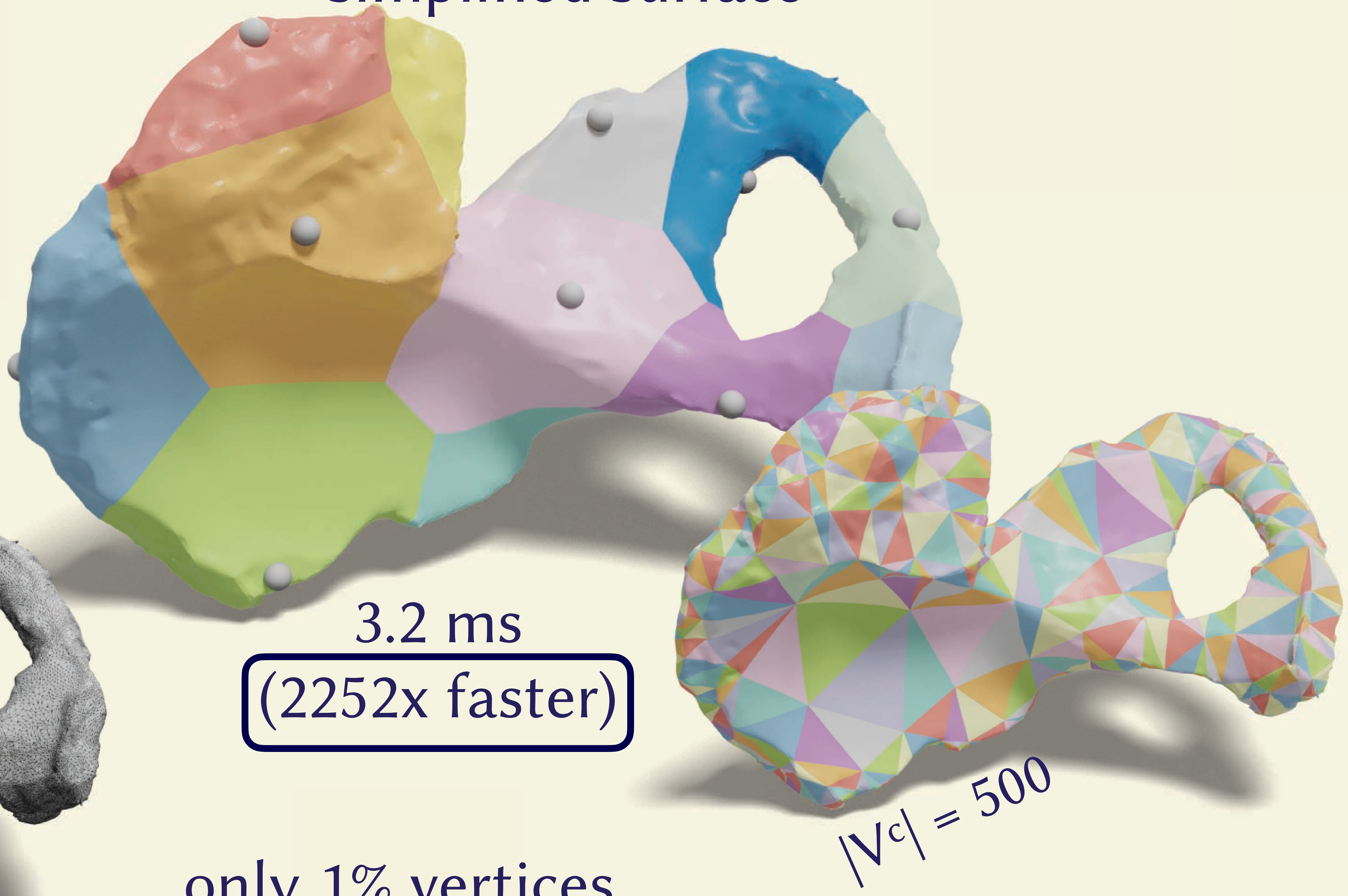
result on
simplified surface

Geodesic Voronoi diagrams

result on
simplified surface

ground truth





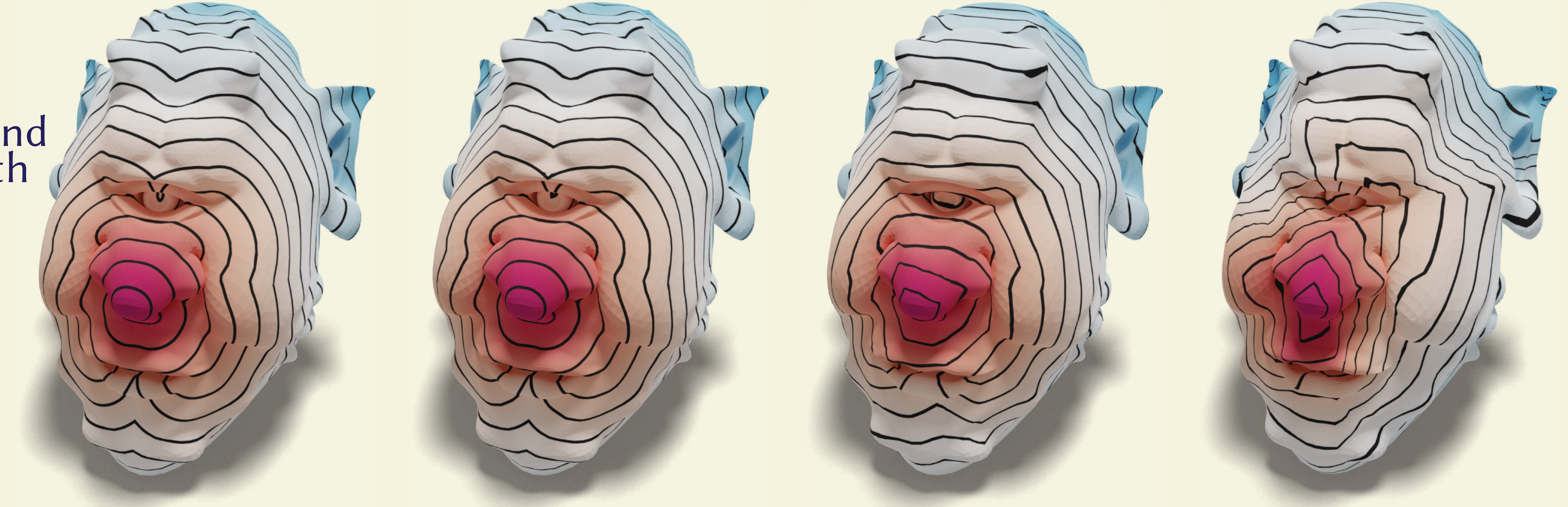
3.2 ms
(2252x faster)

only 1% vertices
misclassified

Speedup vs error in geodesic distance

$|V| = 20k$ $|V^c| = 10k$ $|V^c| = 100$ $|V^c| = 10$

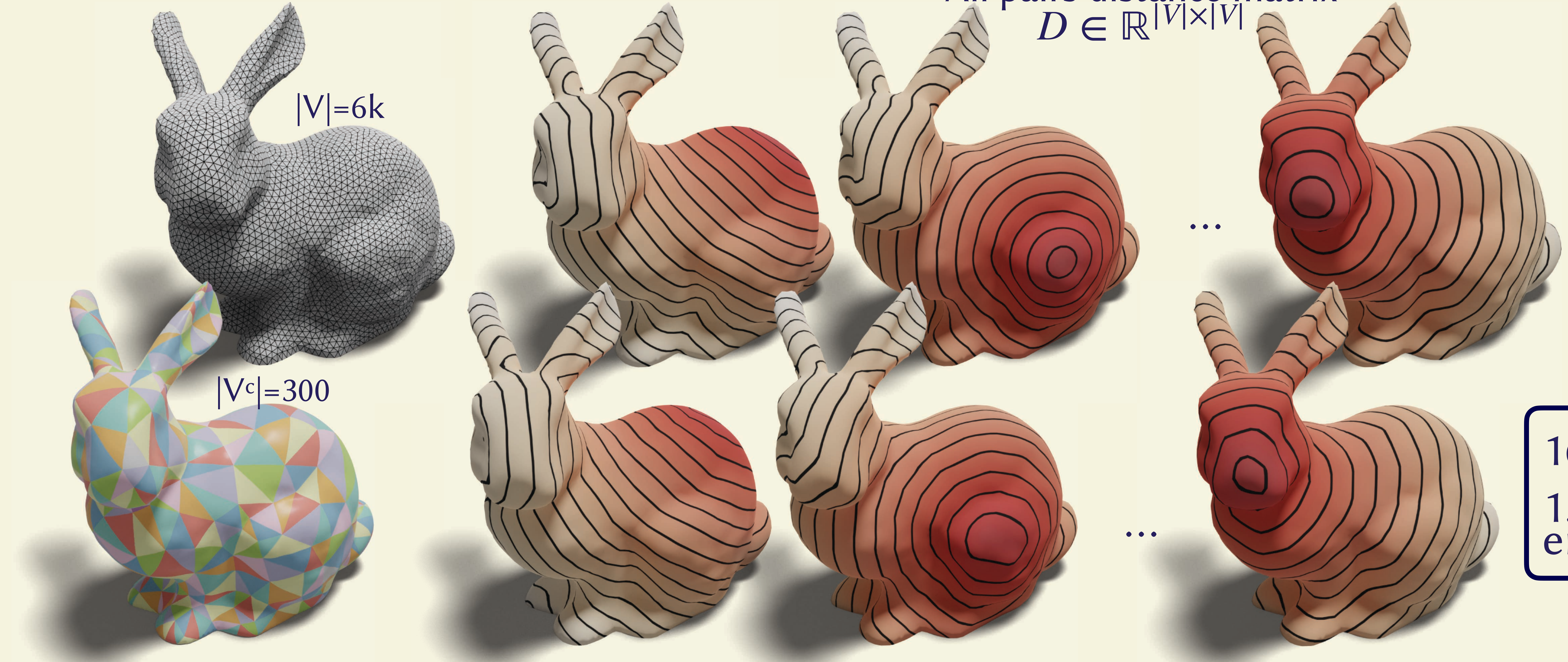
ground truth



speedup/error: 3x / 0.0002% 840x / 0.2% 4880x / 1.5%

Low rank all-pairs distance matrix approximation

All pairs distance matrix
$D \in \mathbb{R}^{|V|\times|V|}$

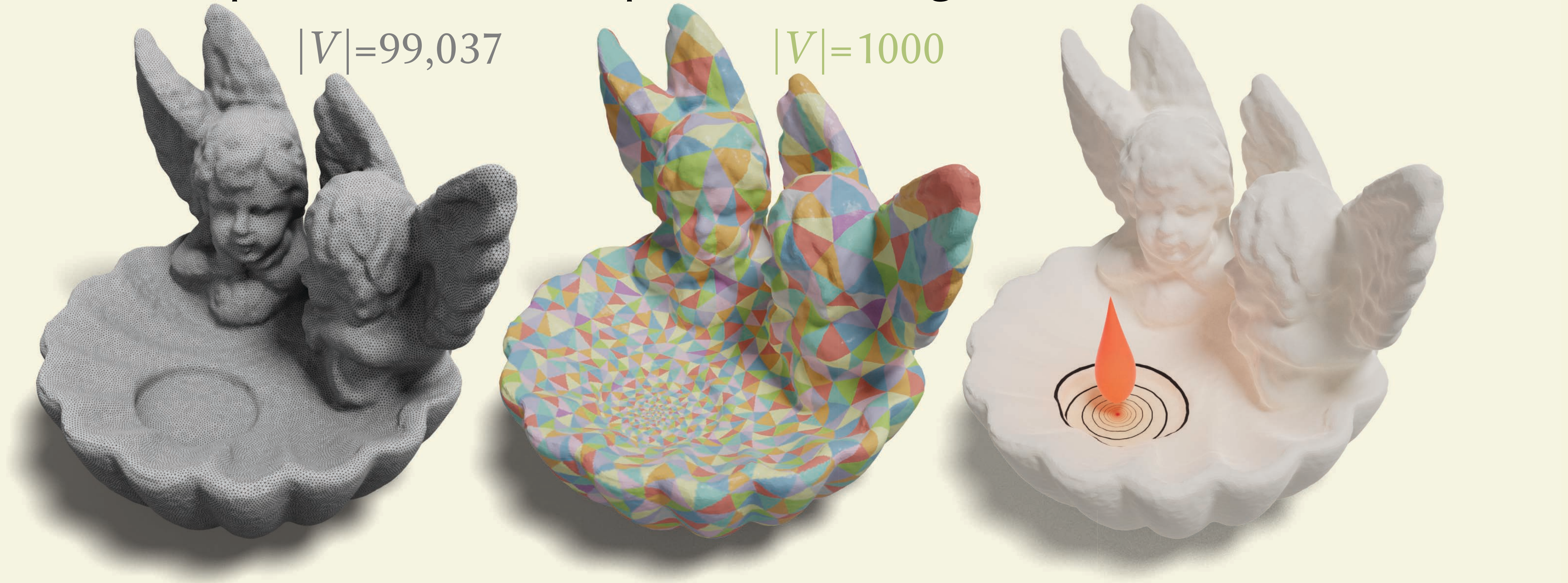


Distance matrix of simplified mesh ➕ Prolongation operator $P : \mathbb{R}^{|V^c|} \to \mathbb{R}^{|V|}$: Approximate distance matrix $\hat{D} = P\tilde{D}P^{\top}$

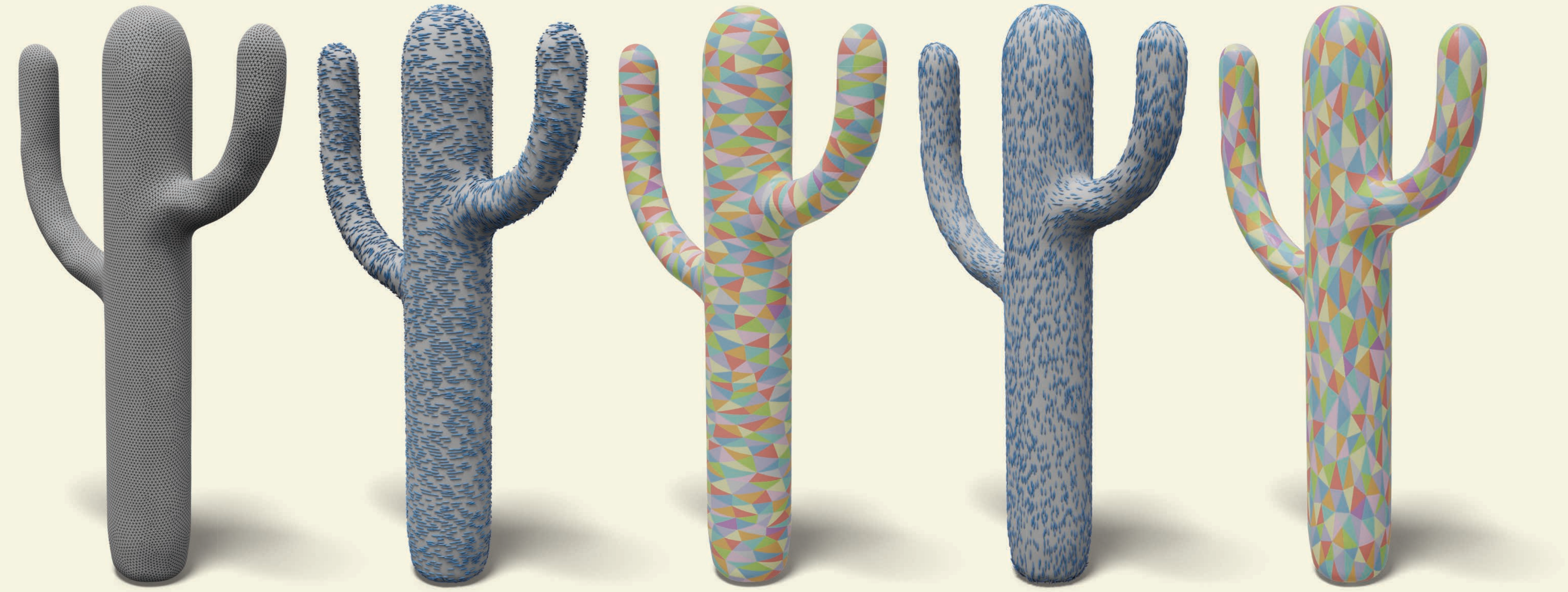
1650x faster
1.4% relative error

Adaptive simplification

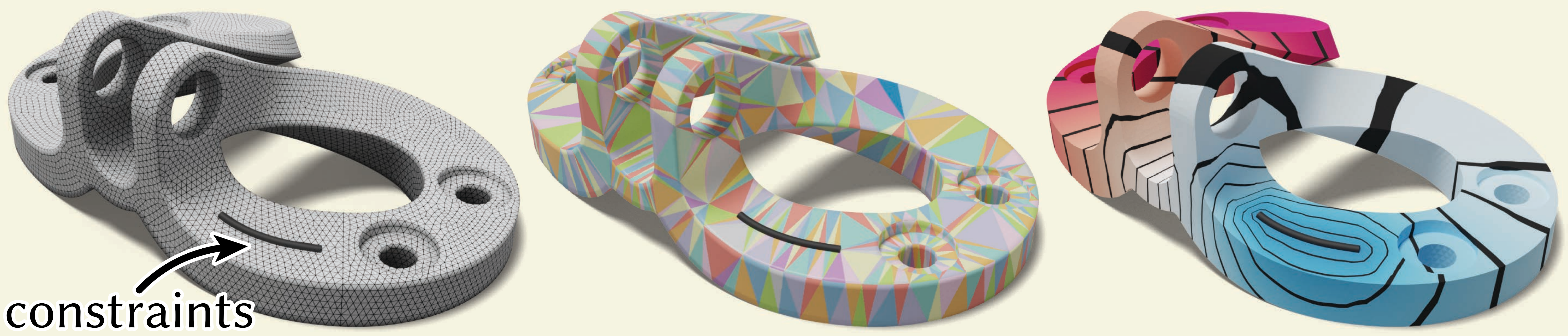
input | adaptive coarsening | heat kernel



input | anisotropic coarsening (max principal direction) | anisotropic coarsening (min principal direction)



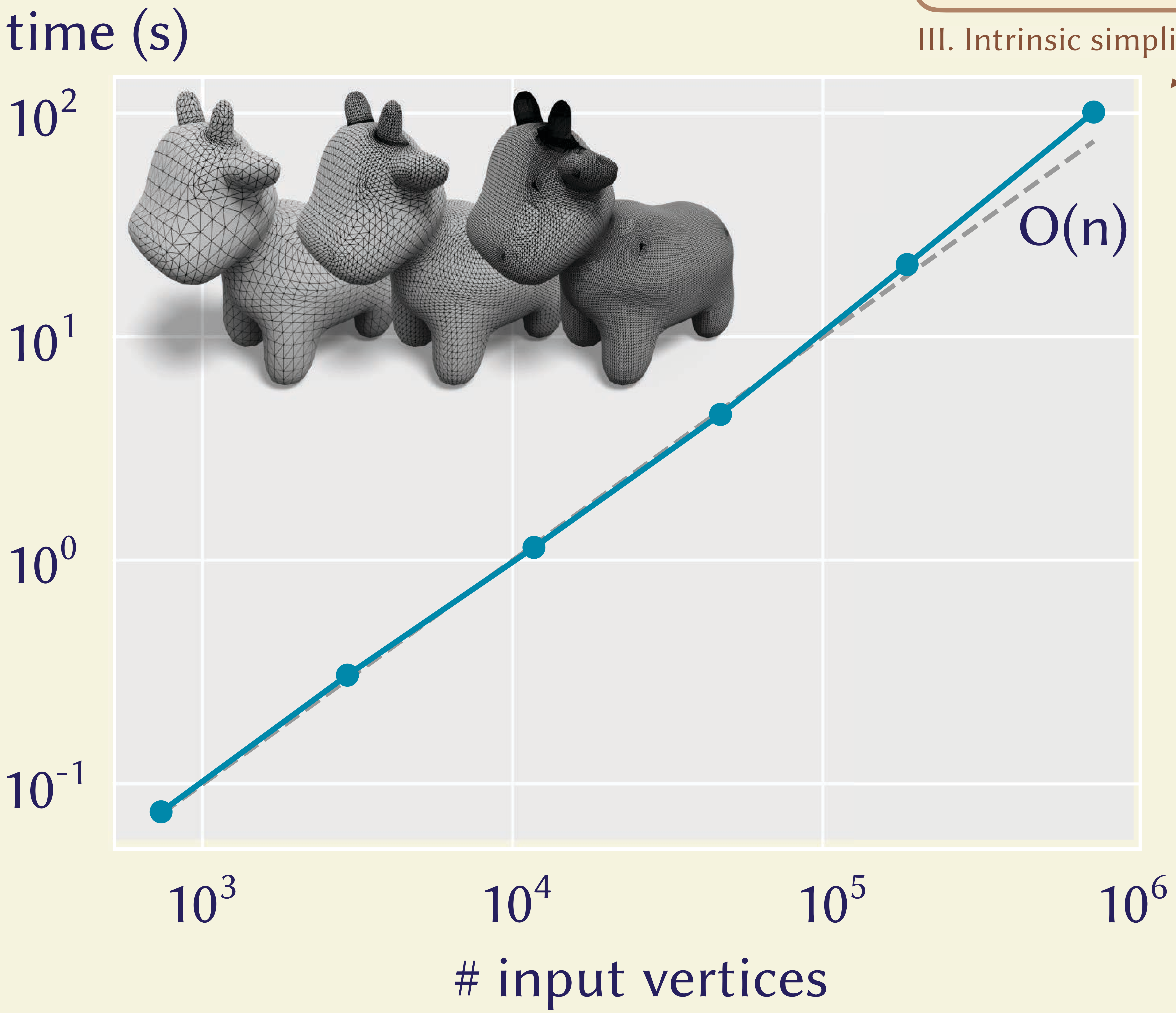
input | constrained coarsening | Poisson solve



Performance

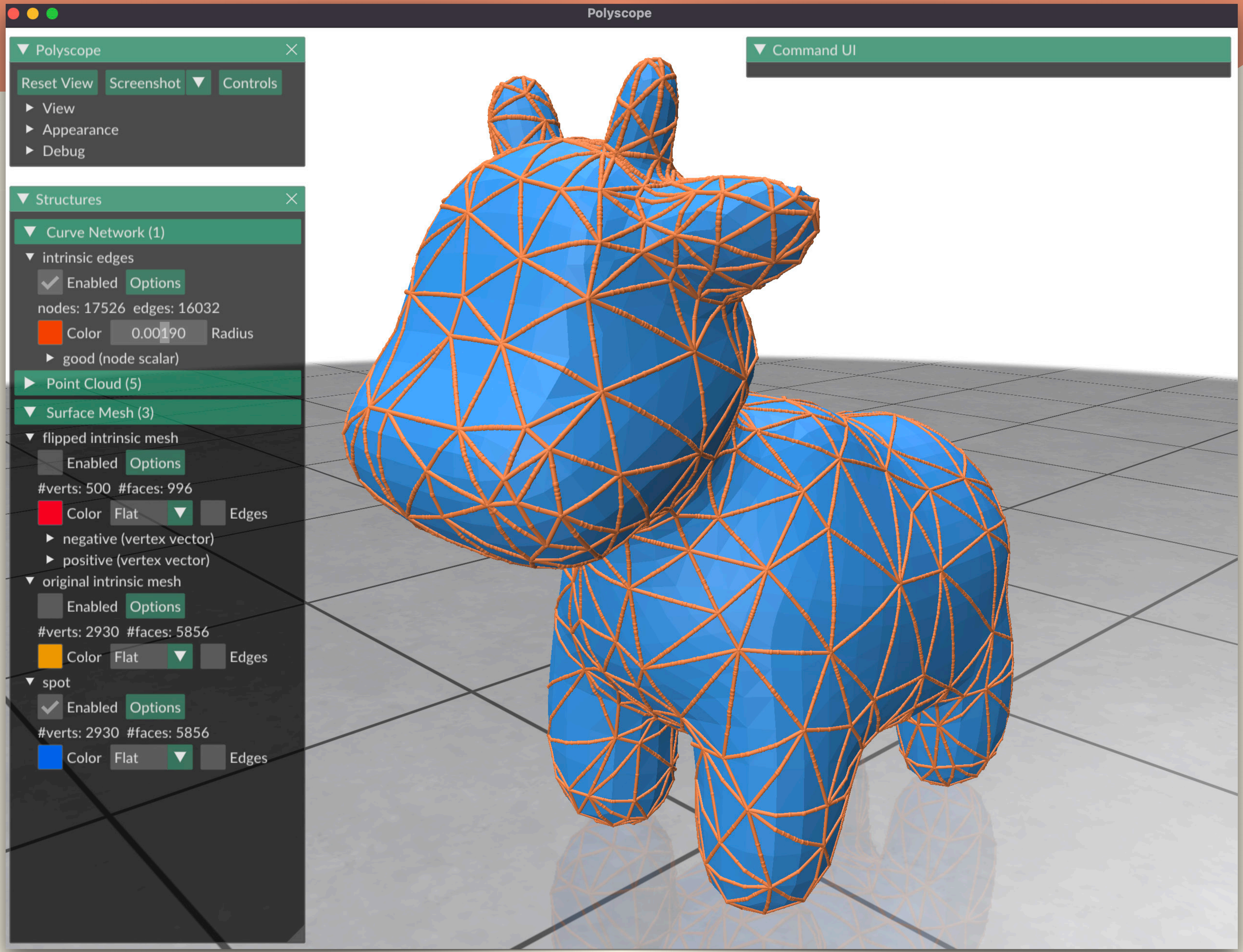
- Linear scaling
  - Constant work per vertex

Removes ~10,000 vertices per second



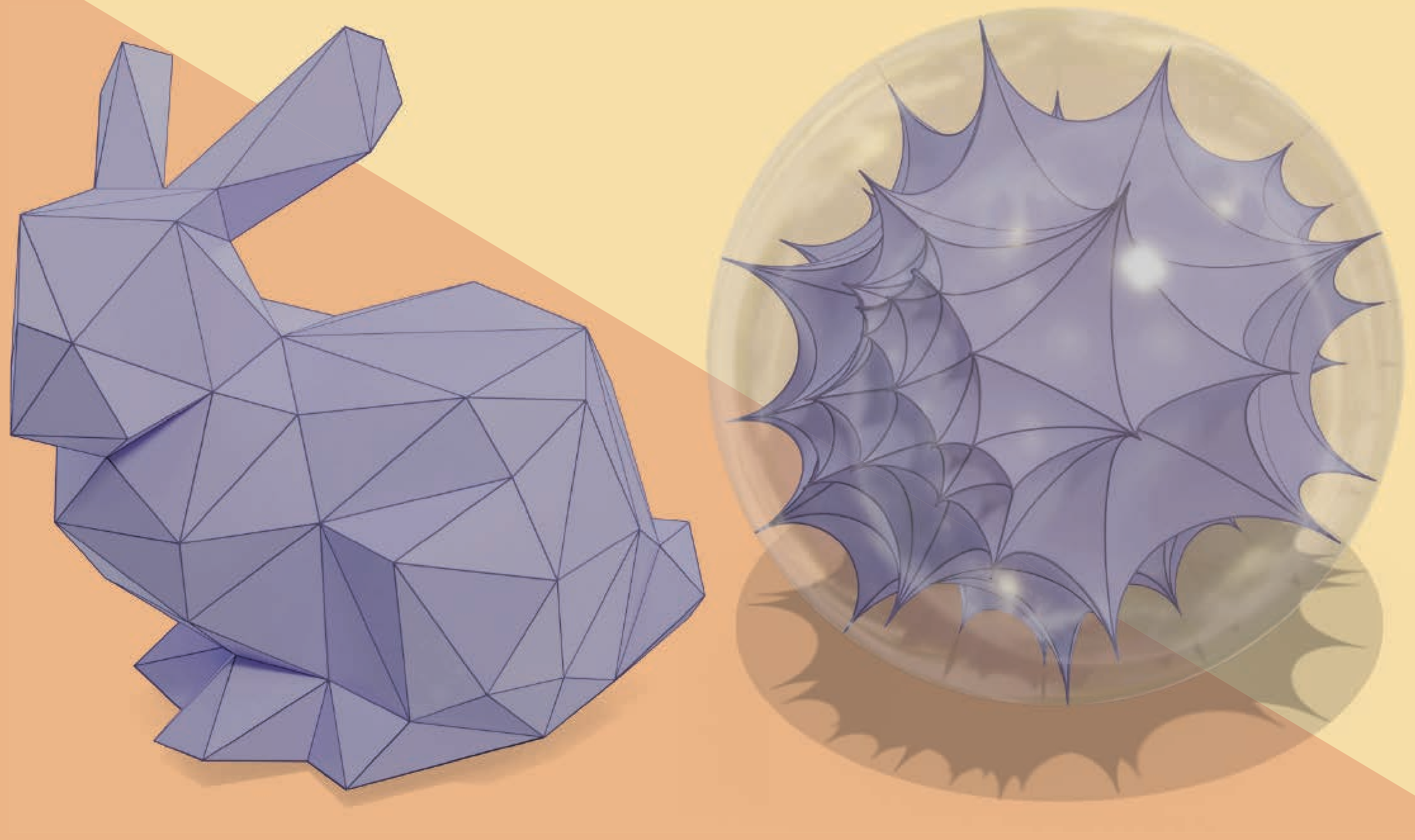
III. Intrinsic simplification
▸ *results*

Try it out yourself (… in the near future)



Coming soon to https://github.com/HTDerekLiu/intrinsic-simplification

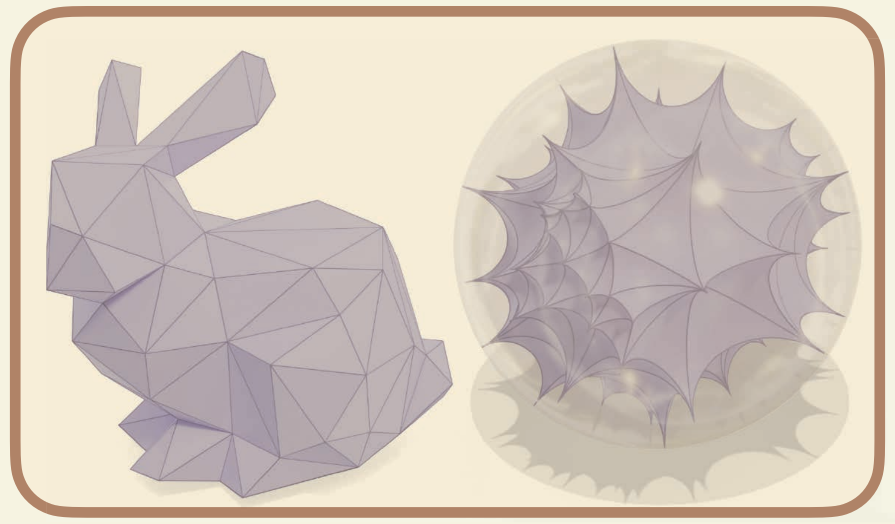
IV. Discrete Uniformization



[**G.**, Springborn, & Crane. 2021. Discrete conformal equivalence of polyhedral surfaces. *ACM Transactions on Graphics*]

The uniformization theorem

[Poincare 1907; Koebe 1907; Troyanov 1991]



Any surface is conformally equivalent to a surface of constant curvature.

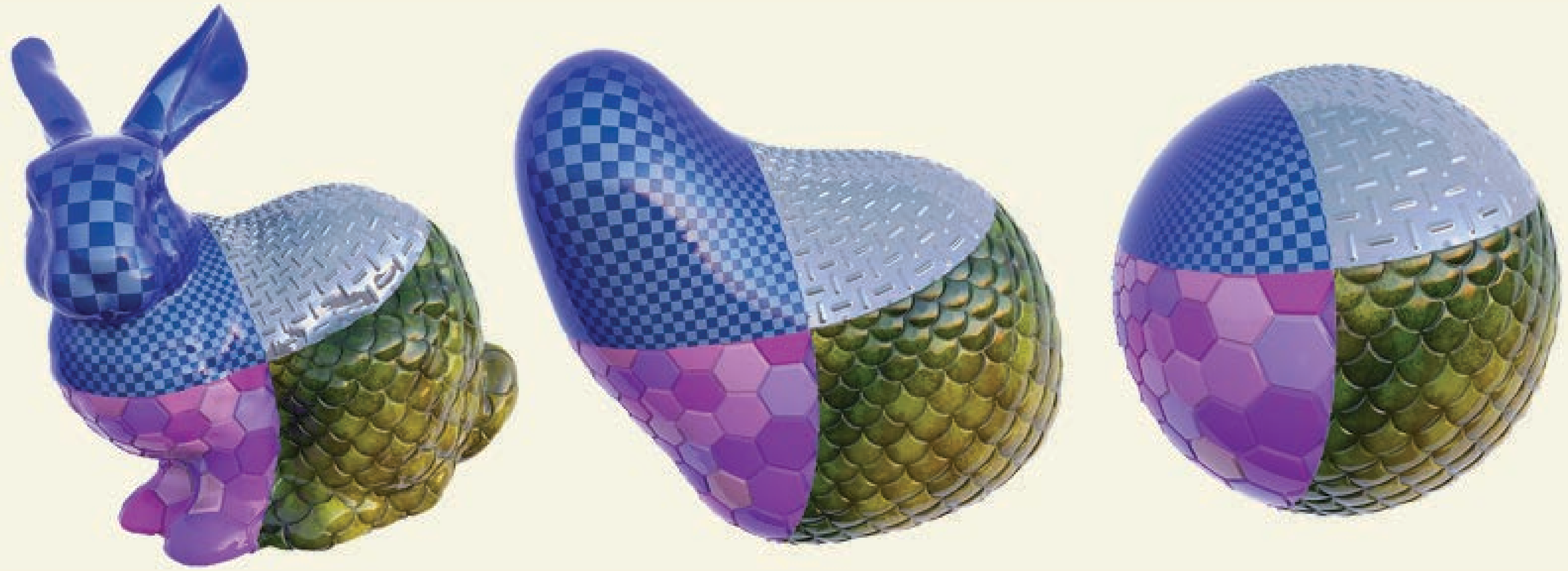
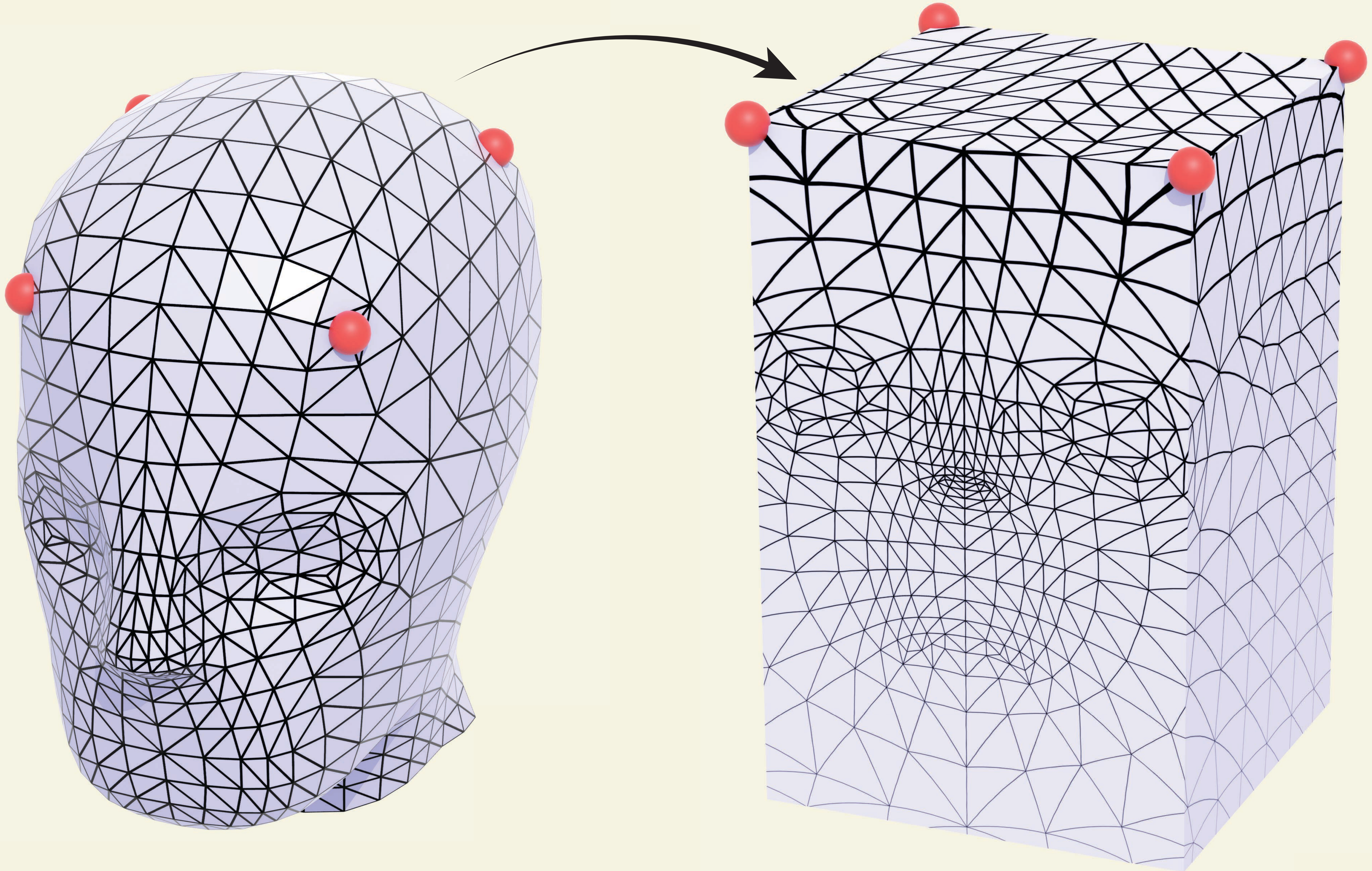
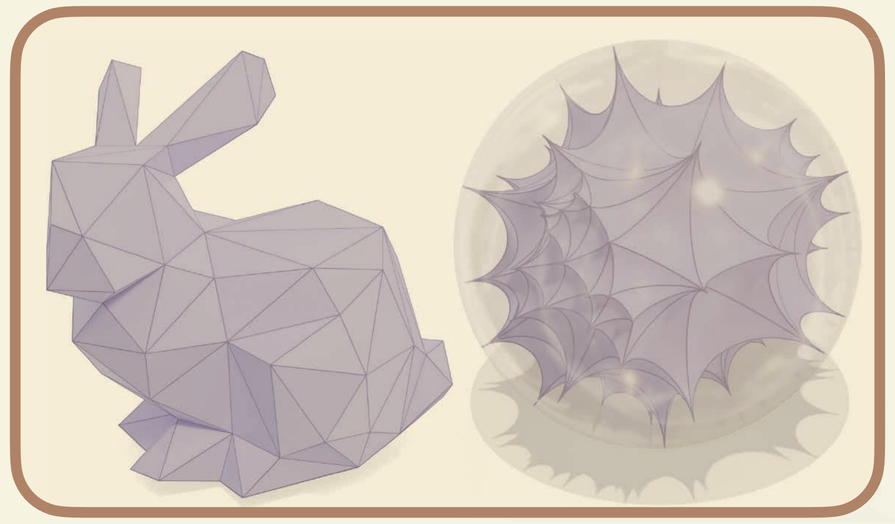


Image: [Crane, Pinkall & Schröder 2013]

The discrete uniformization theorem
[Gu, Luo, Sun & Wu 2018; Springborn 2019]

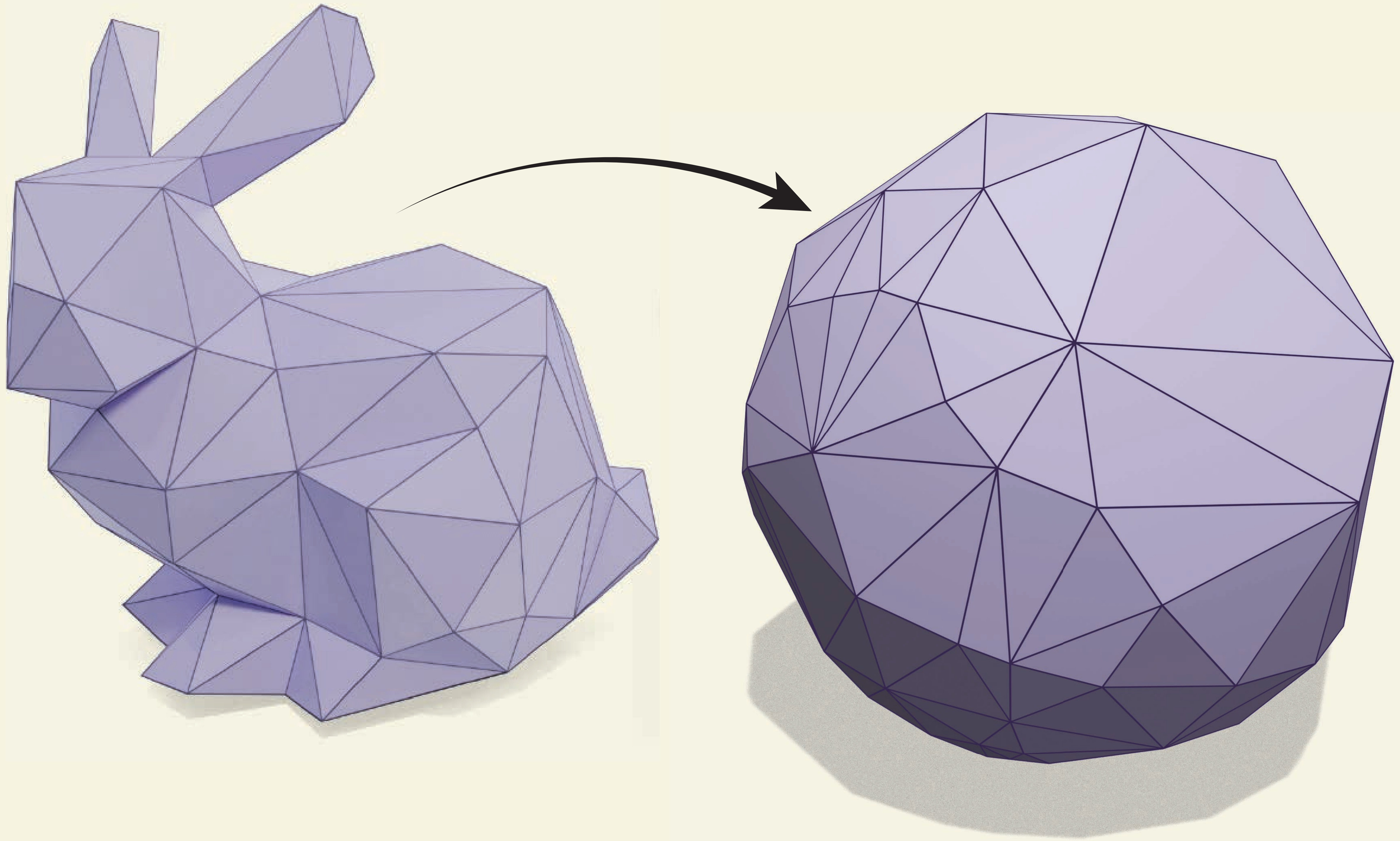
Any positive vertex cone angles satisfying Gauss-Bonnet can be realized by some discrete conformal map.





The discrete spherical uniformization theorem
[Springborn 2019]

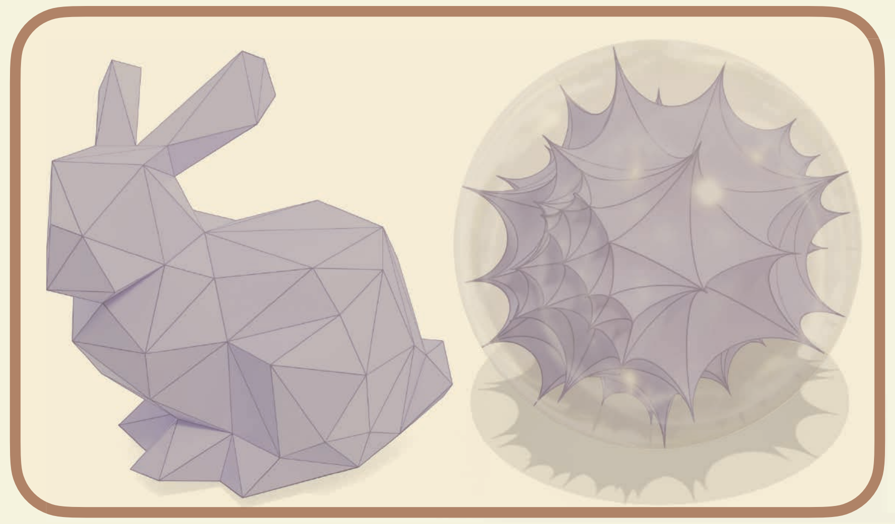
Any simply-connected triangle mesh is discretely conformally equivalent to a mesh whose vertices lie on the unit sphere

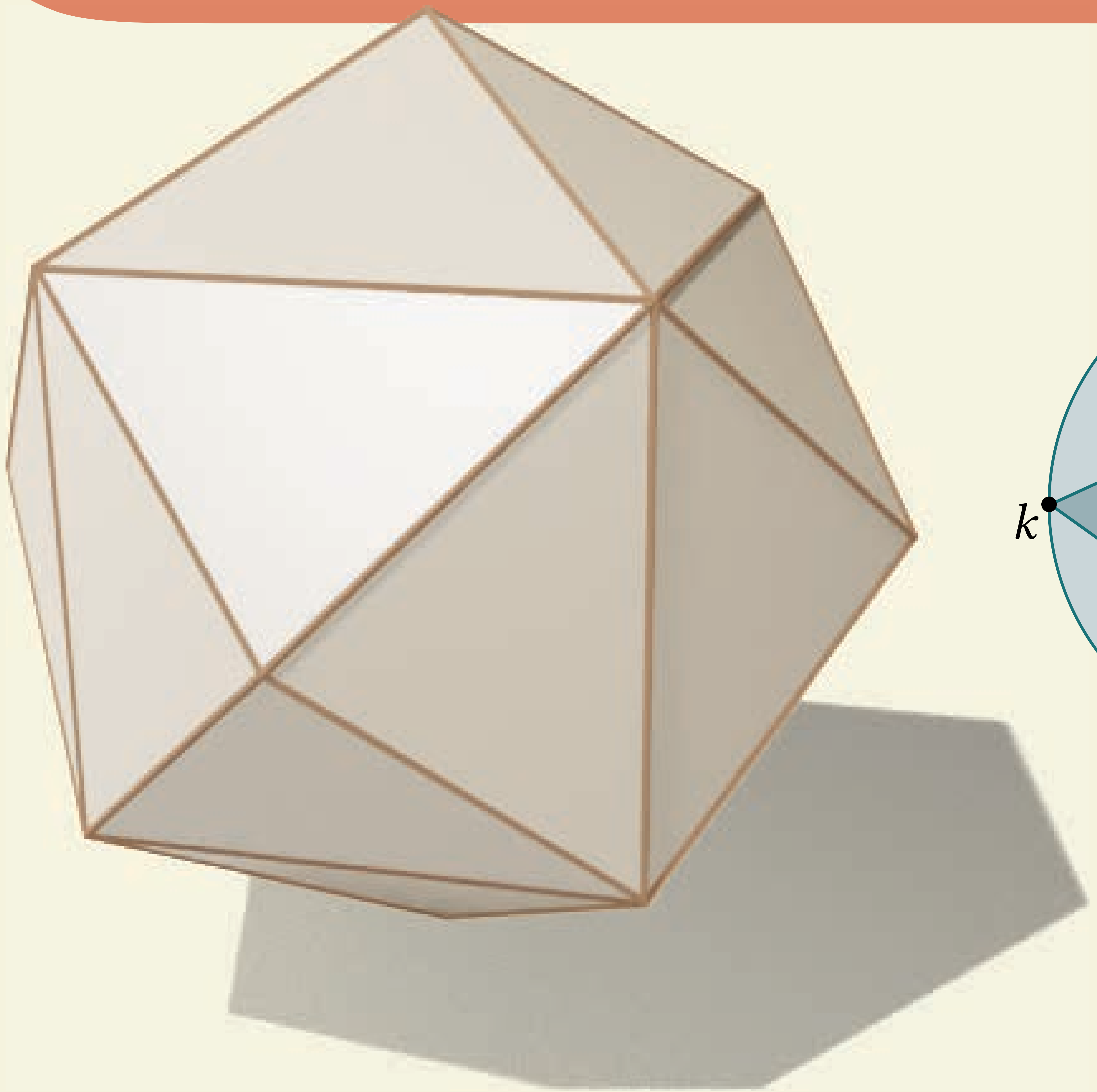


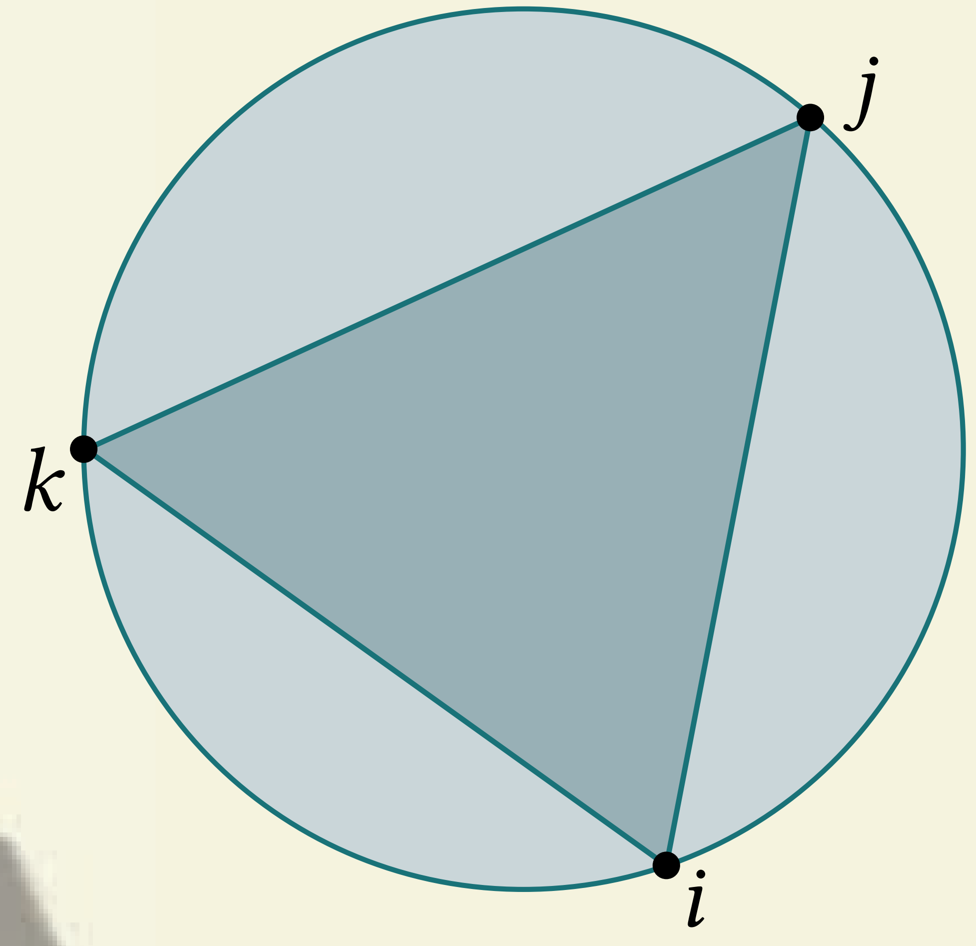
Discrete uniformization in action

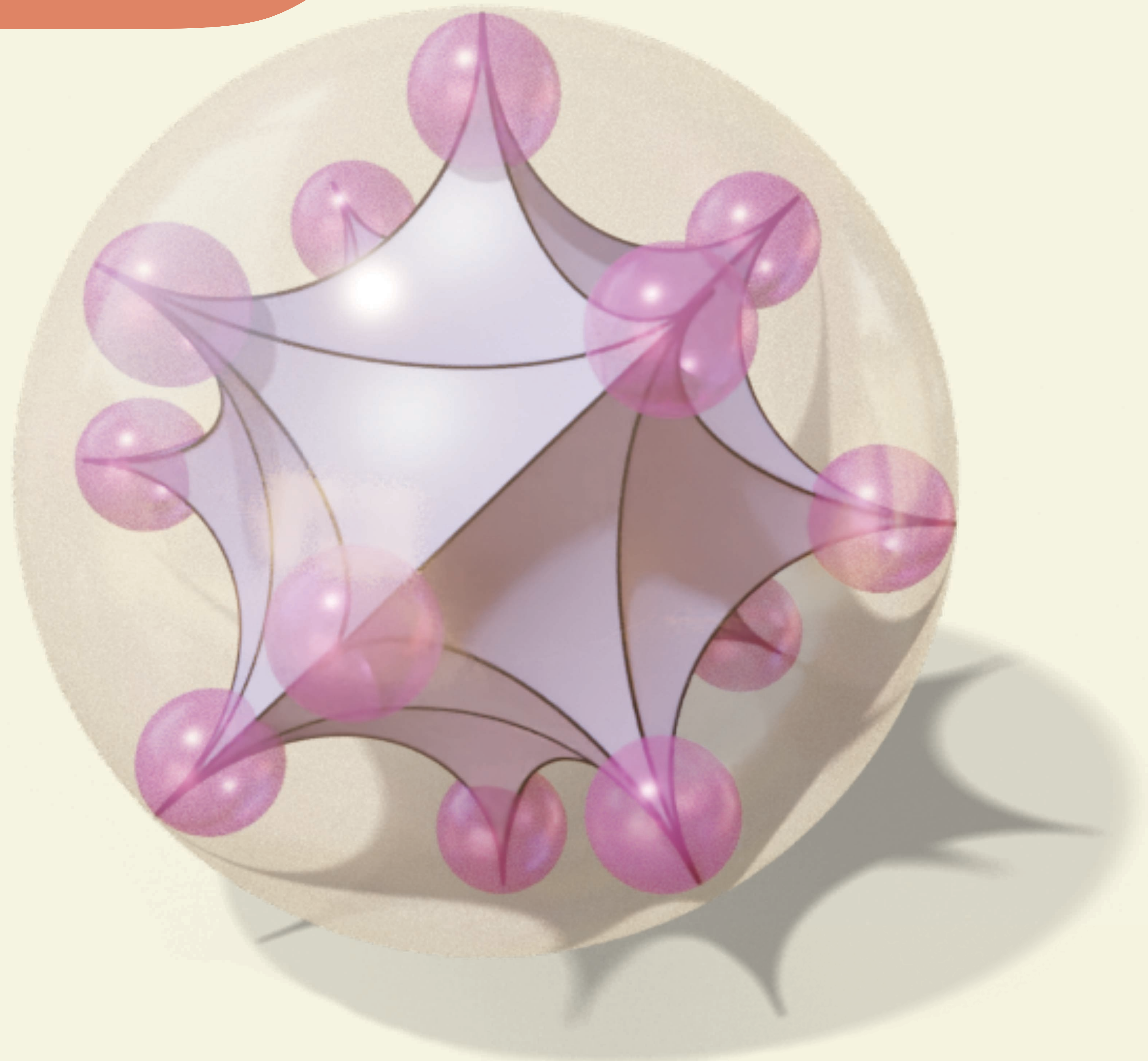
[G., Springborn, & Crane. 2021]

Triangle mesh ⟷ ideal polyhedron
[Bobenko, Pinkall & Springborn 2010]



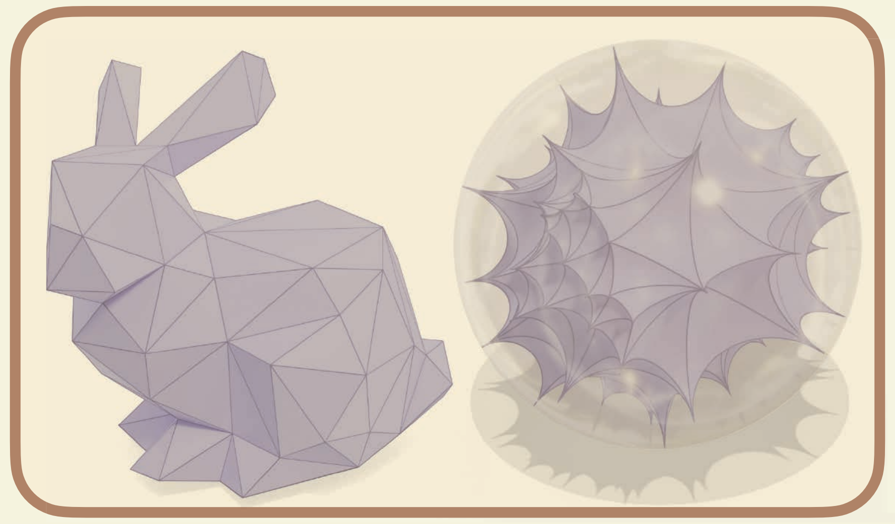


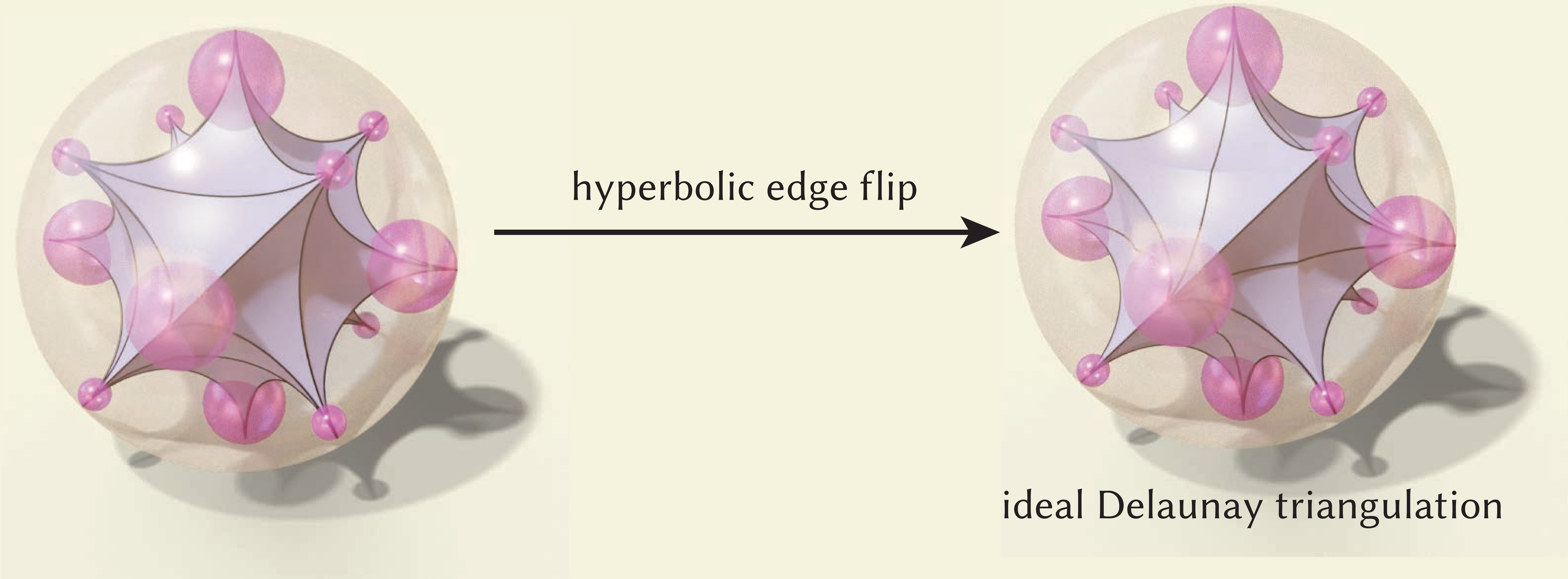




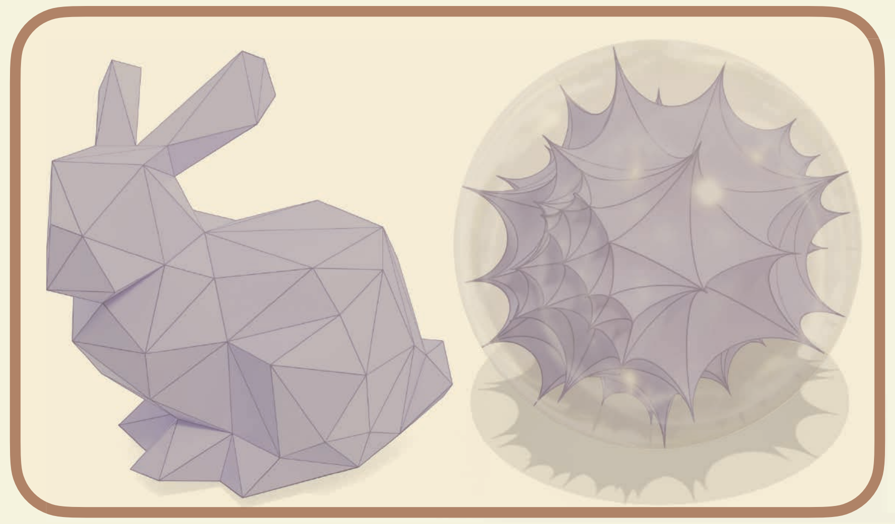
Euclidean triangle in circumcircle ⟷ Klein ideal triangle

Ideal Delaunay triangulations



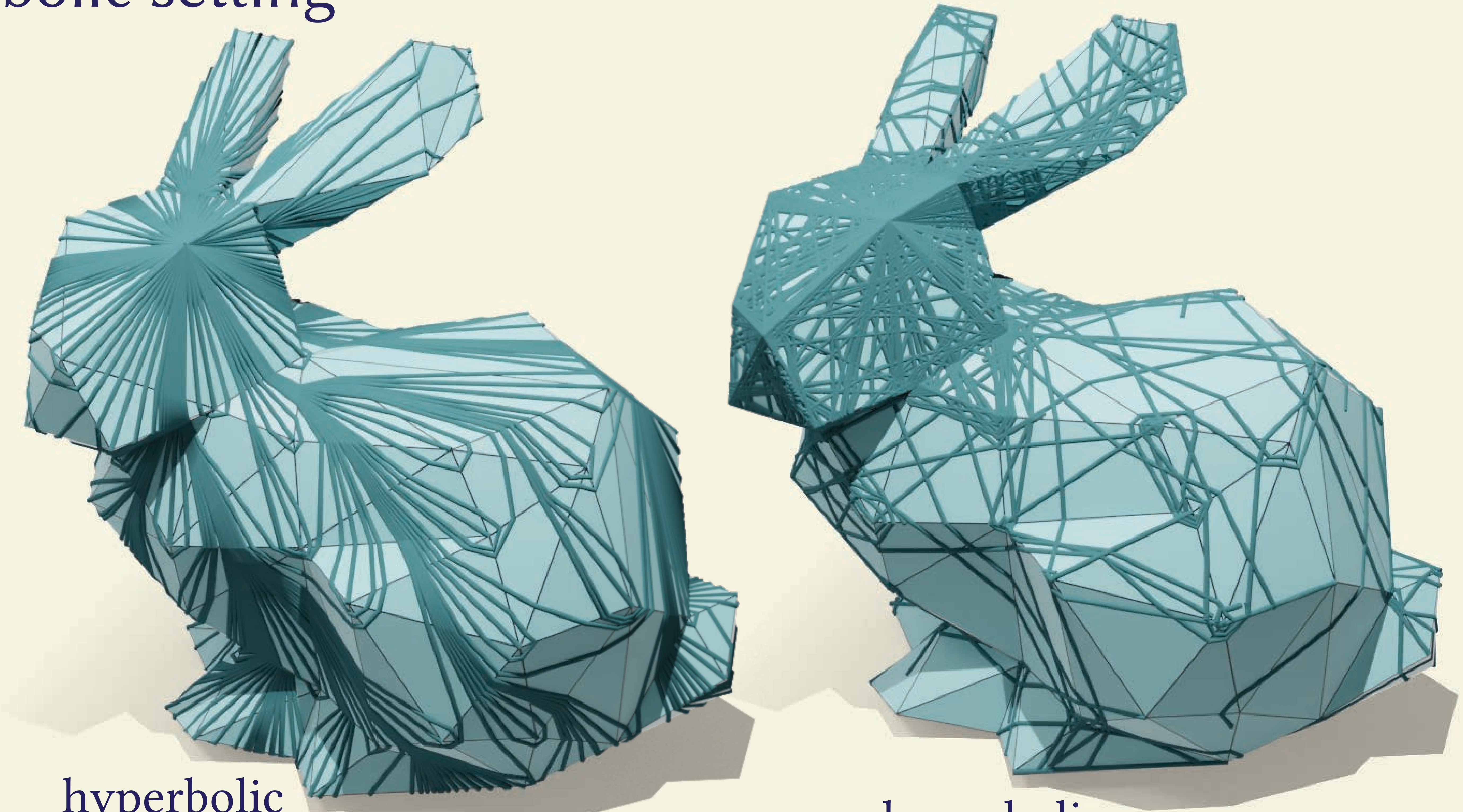


Hyperbolic correspondence problem



Correspondence between ideal polyhedra

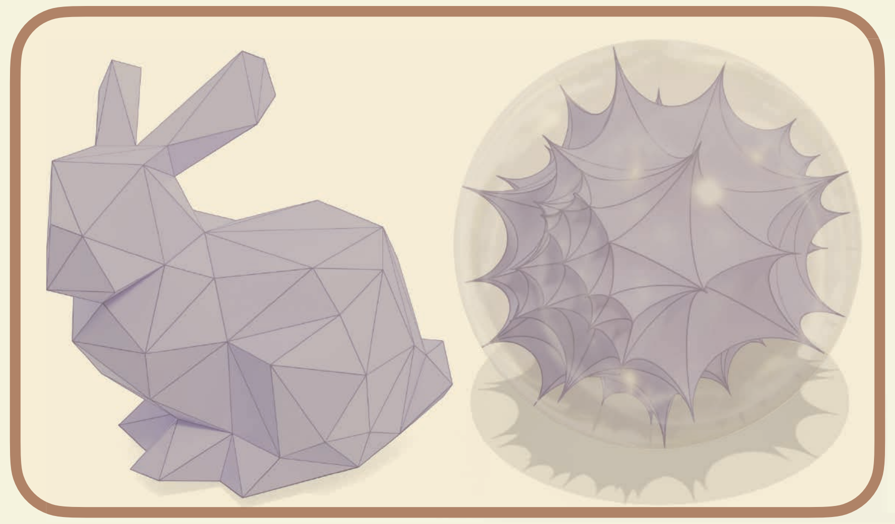
- Adapt Euclidean techniques to hyperbolic setting
- Integer coordinates essential



hyperbolic integer coordinates

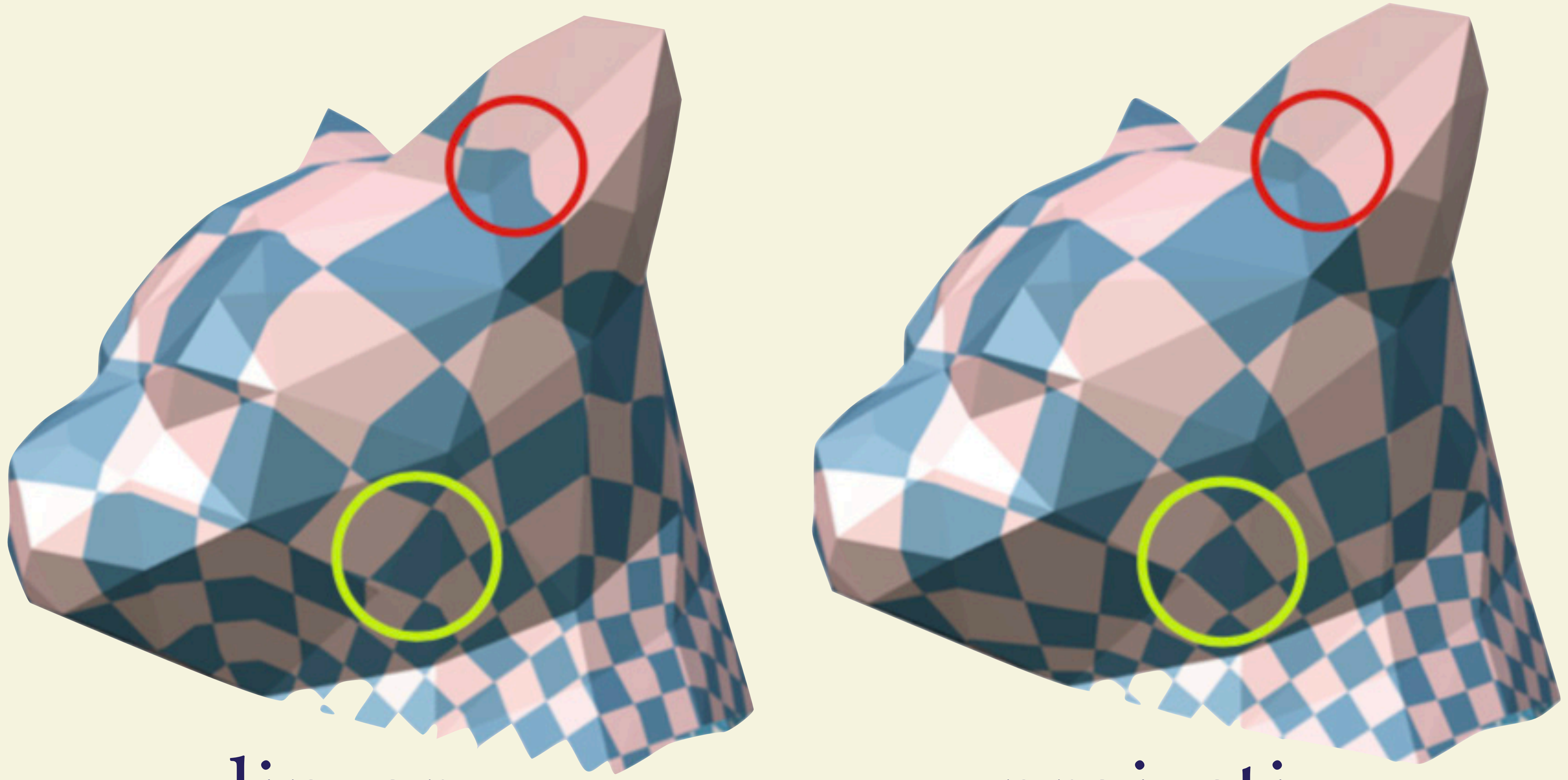
hyperbolic signposts

Projective interpolation

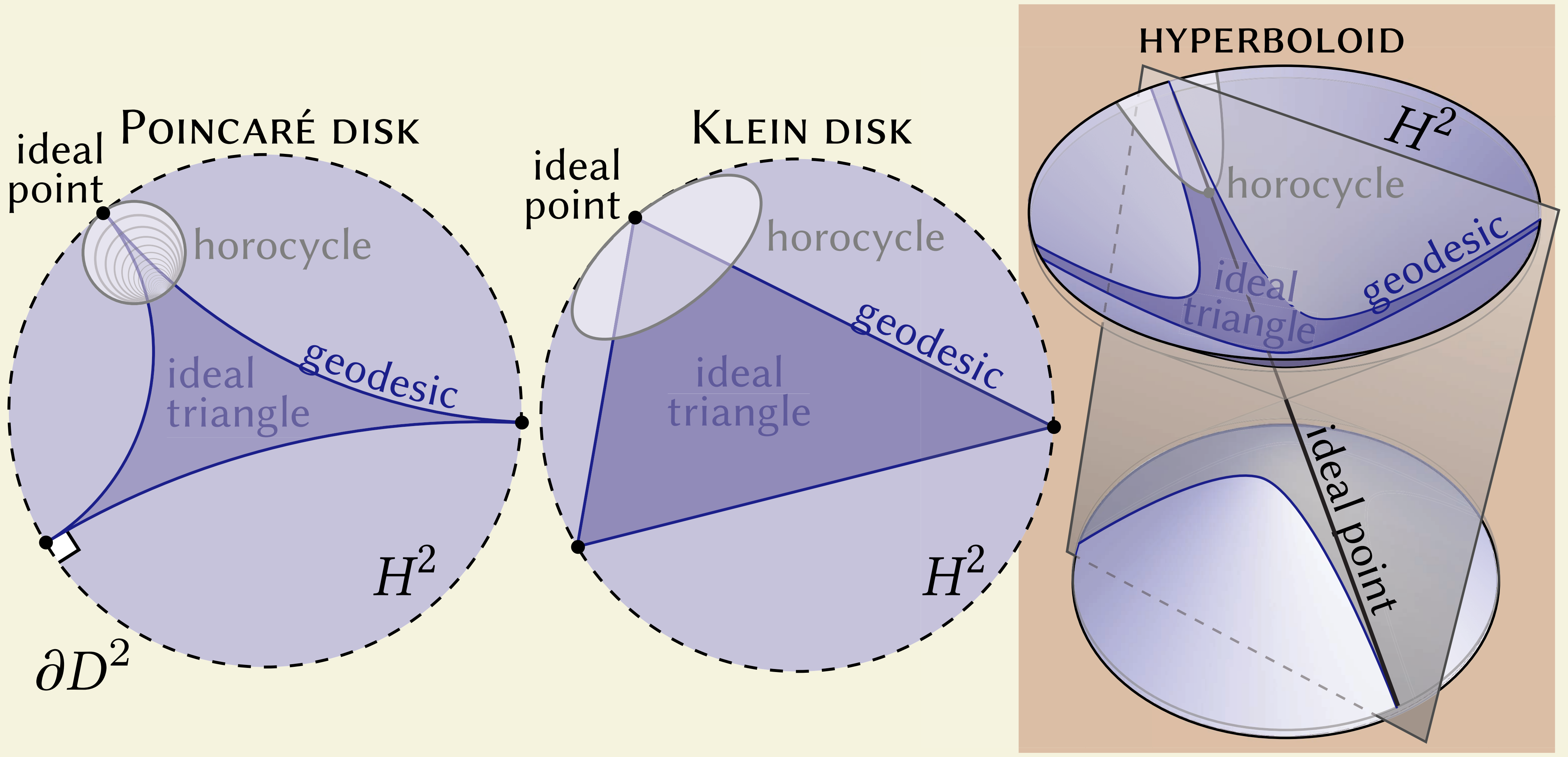


- [Springborn, Schröder & Pinkall 2008]: projective interpolation
  - Hyperbolic isometry
- In variable triangulation case, lay out triangles in hyperboloid model

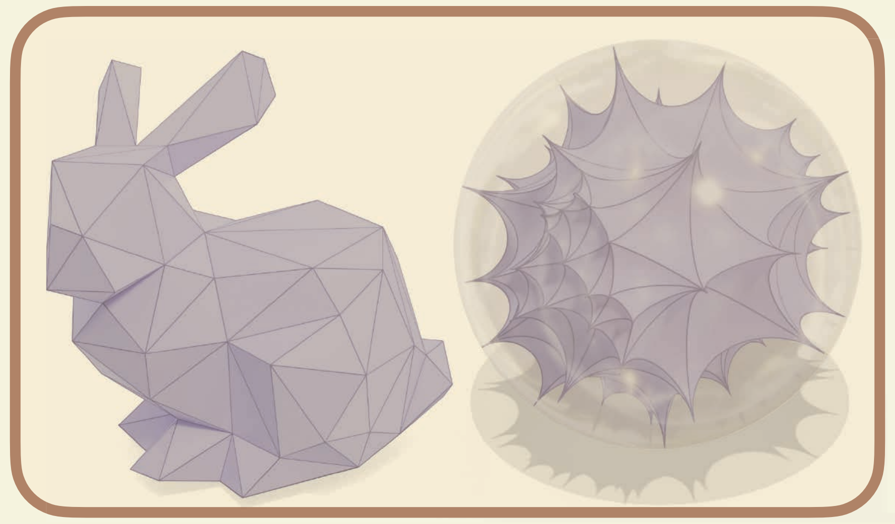
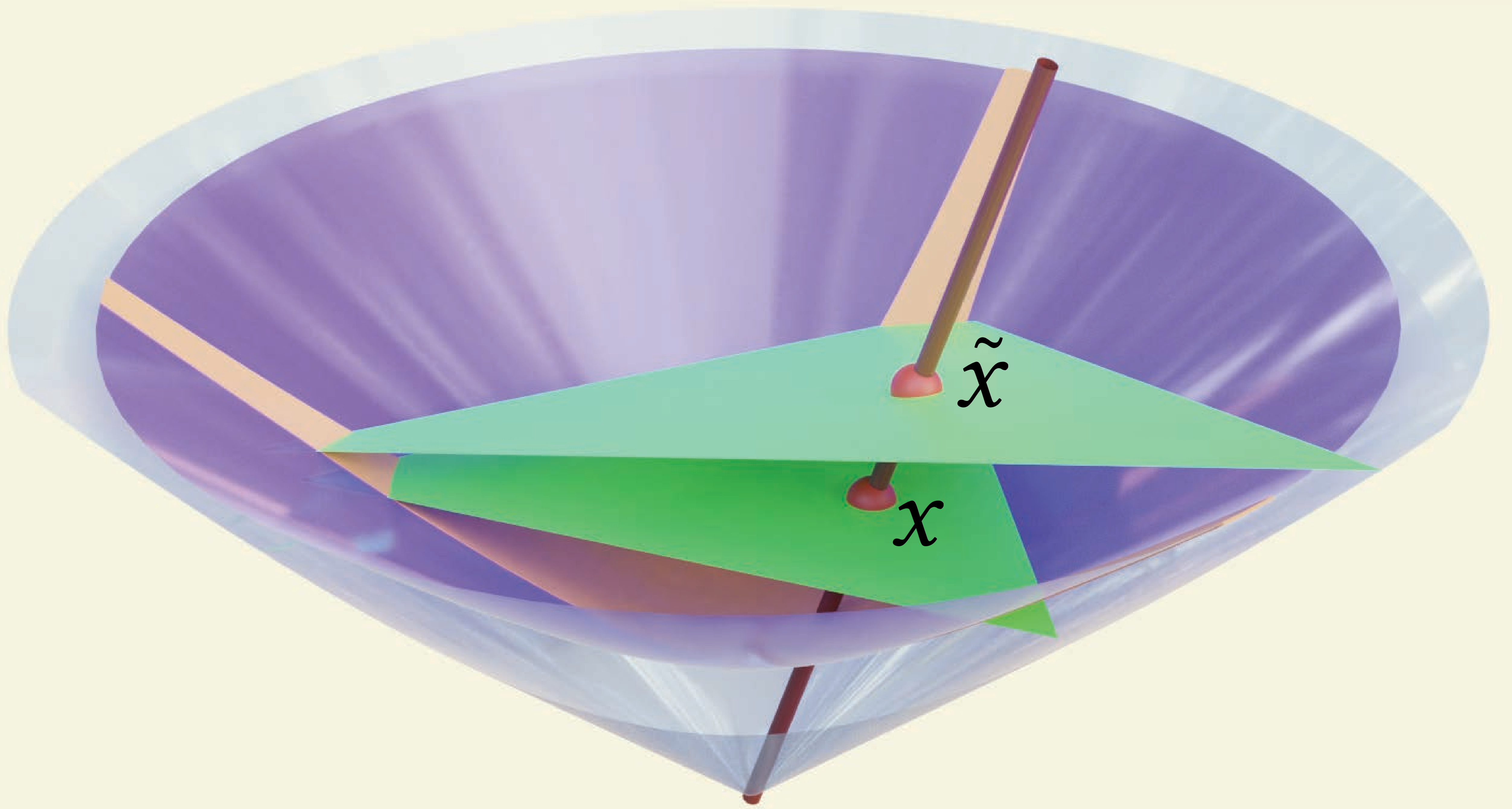
Image: [Springborn, Schröder & Pinkall 2008]



linear projective

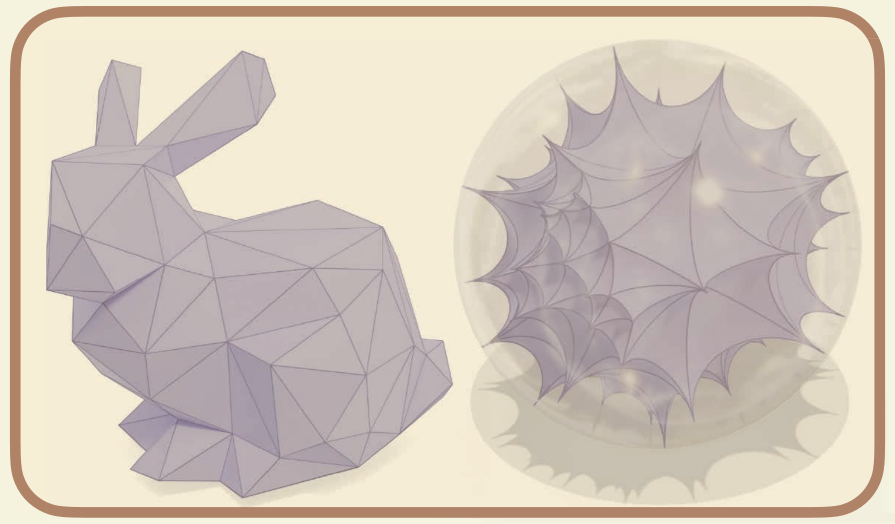


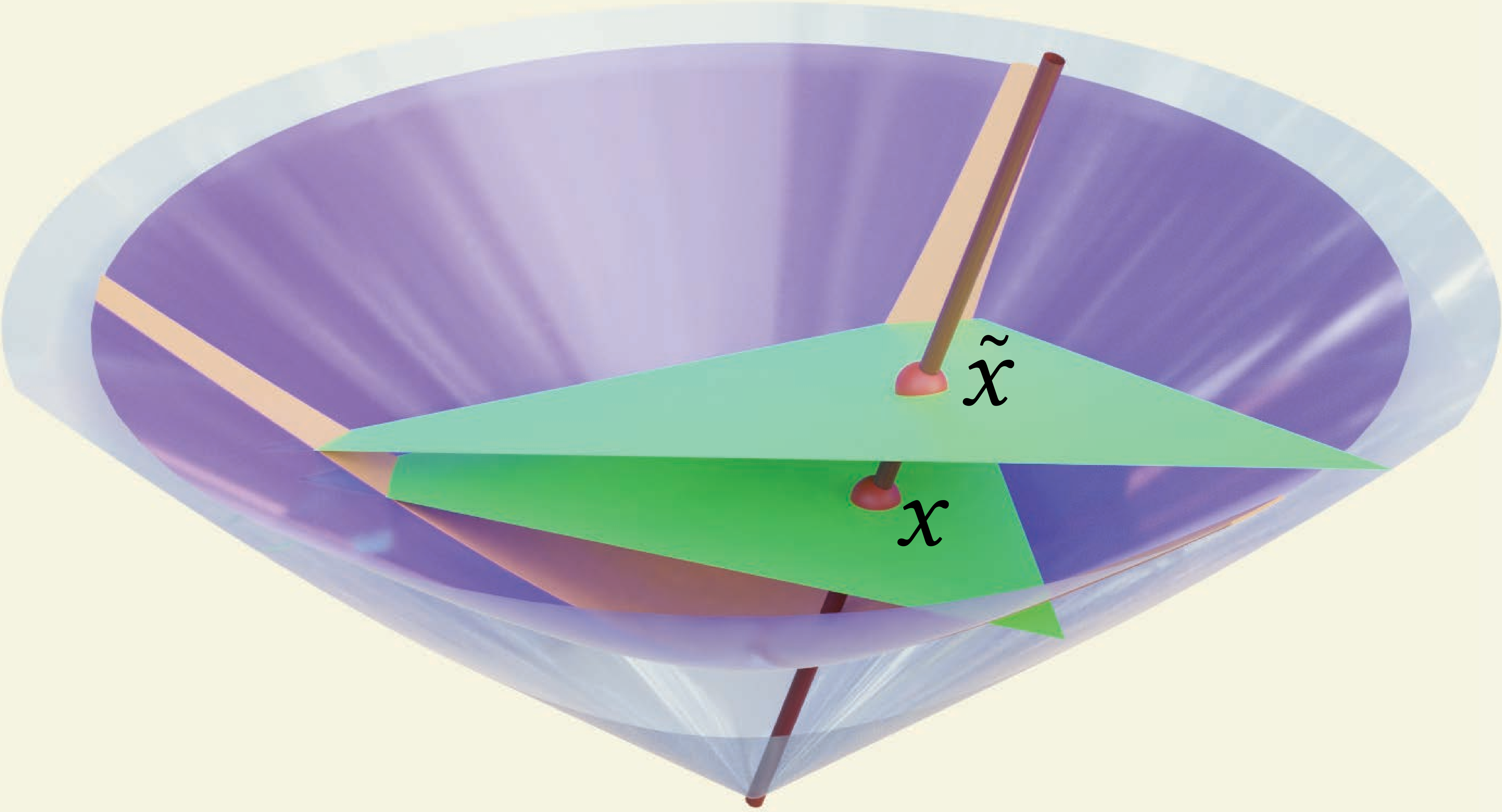
Interpolation in the hyperboloid model

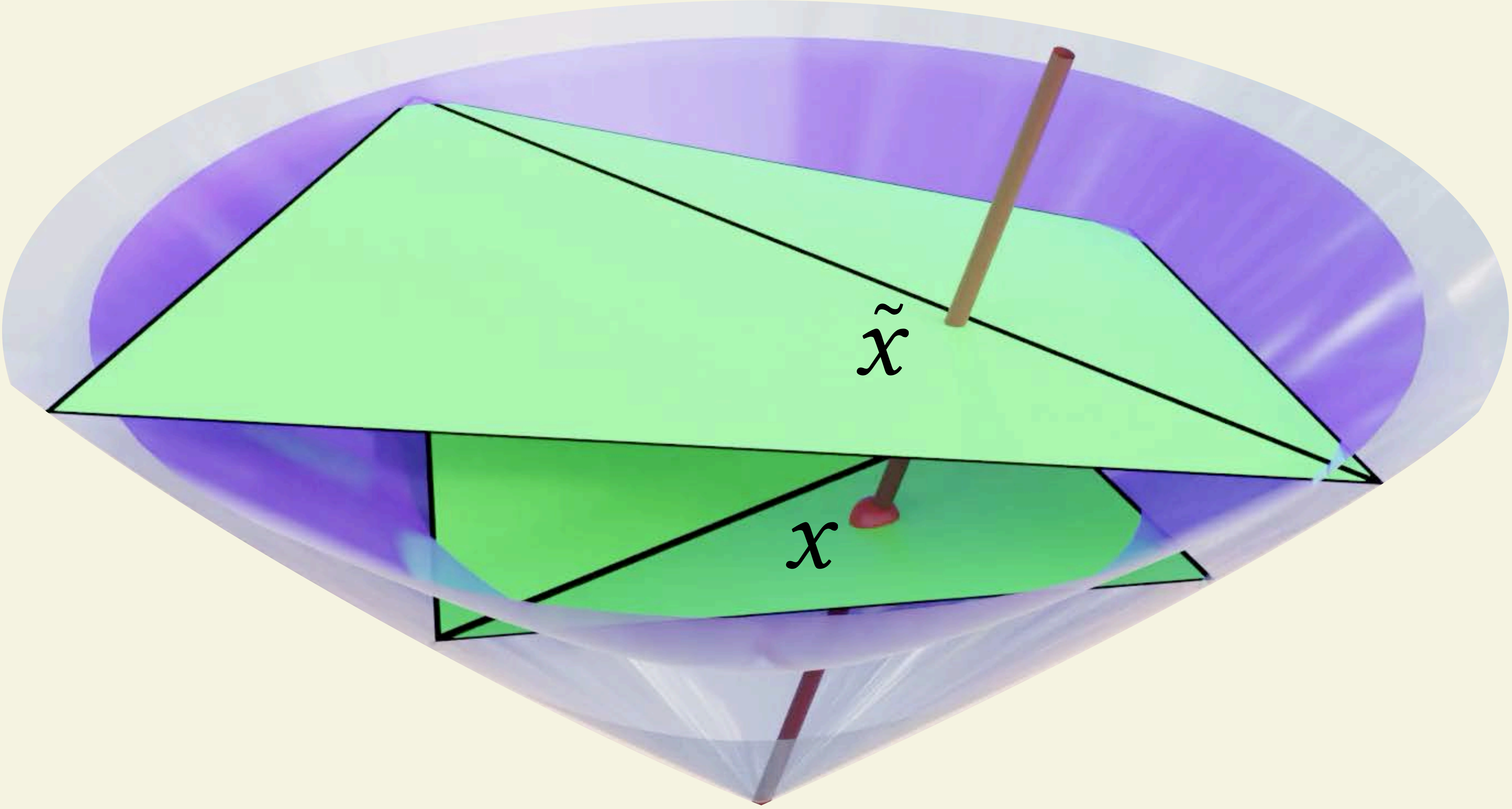
fixed triangulation

Interpolation in the hyperboloid model



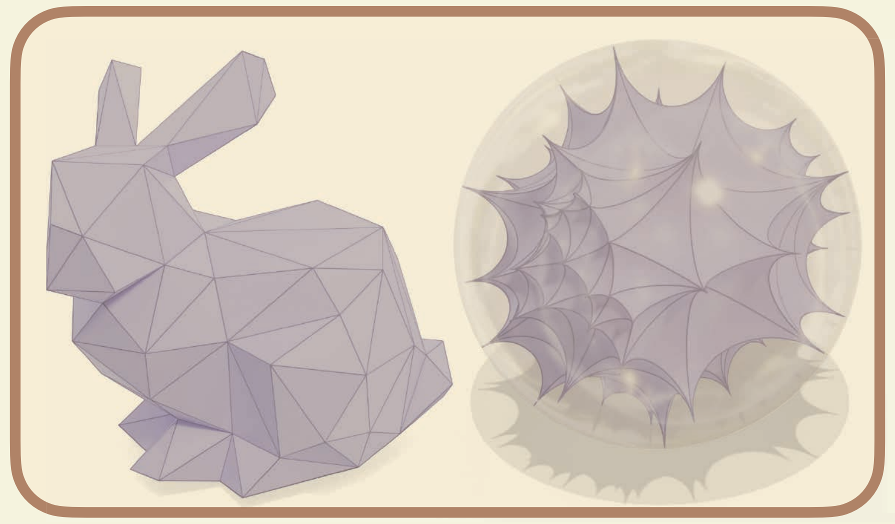


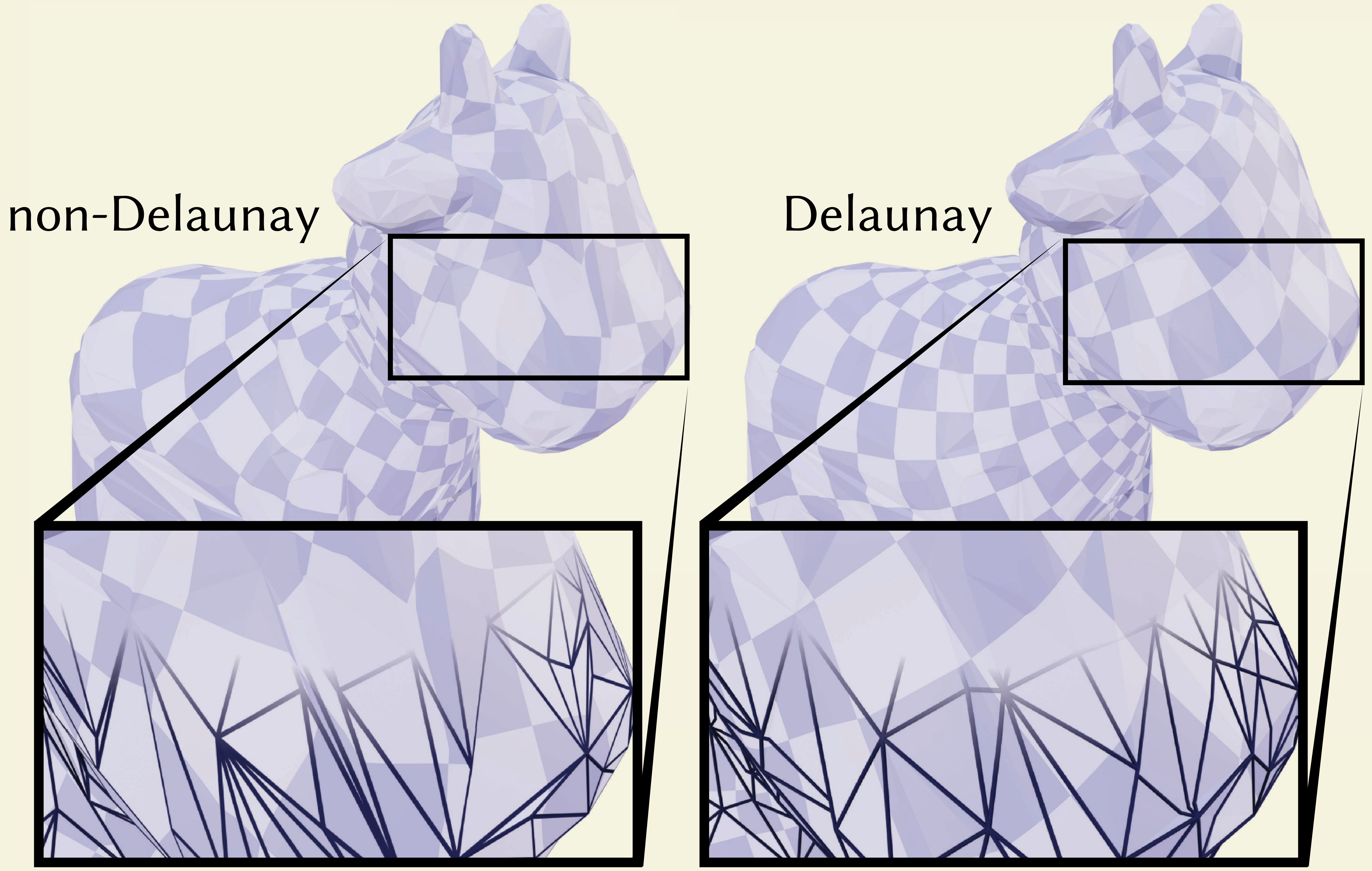
fixed triangulation

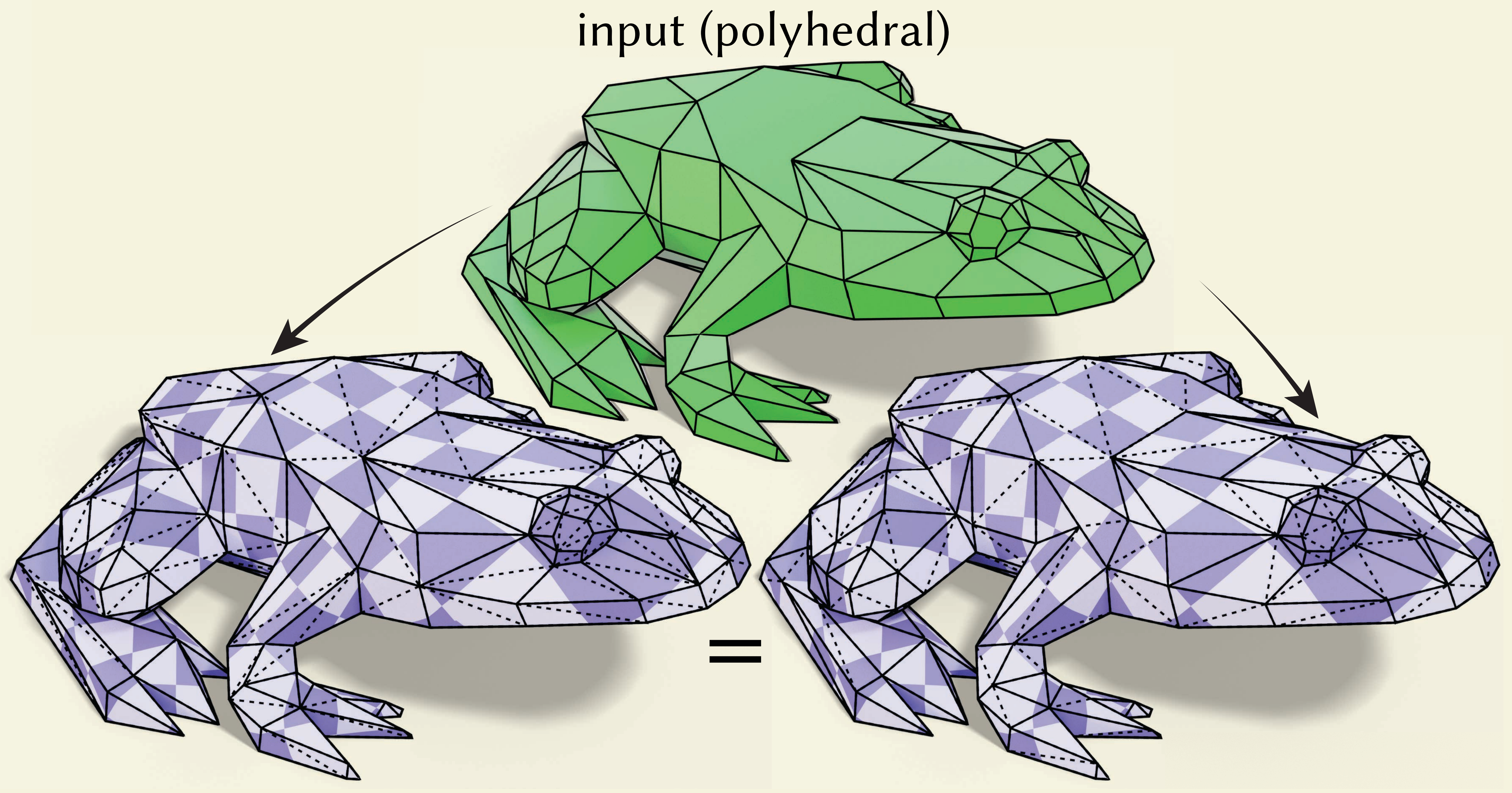


variable triangulation

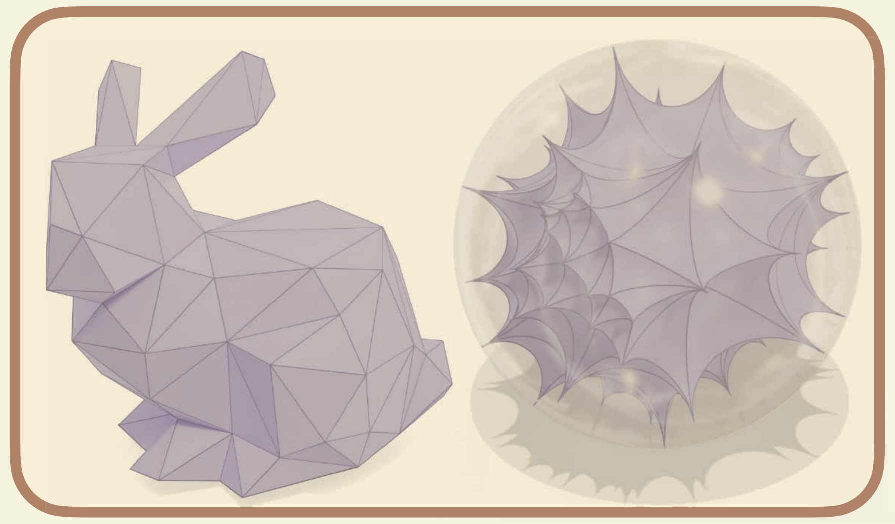
Starting from Delaunay







Final algorithm



flip to (Euclidean) Delaunay

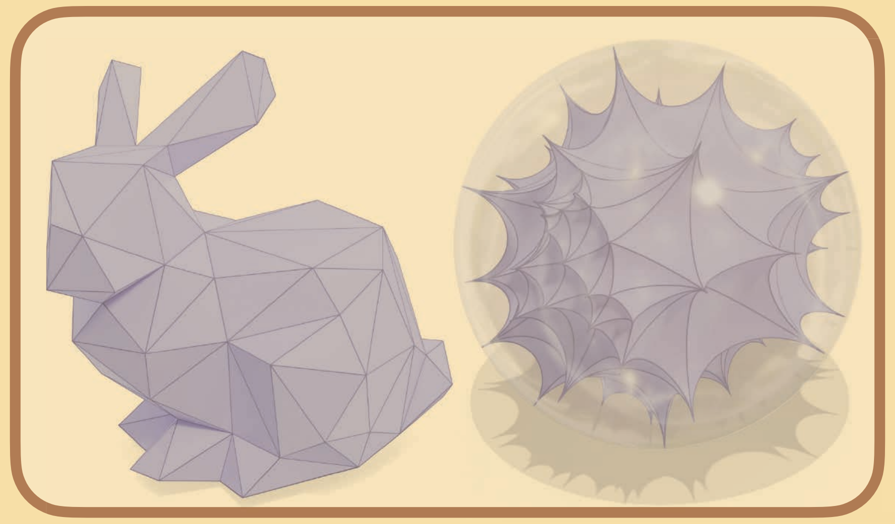
find scale factors

lay out in plane

extract correspondence

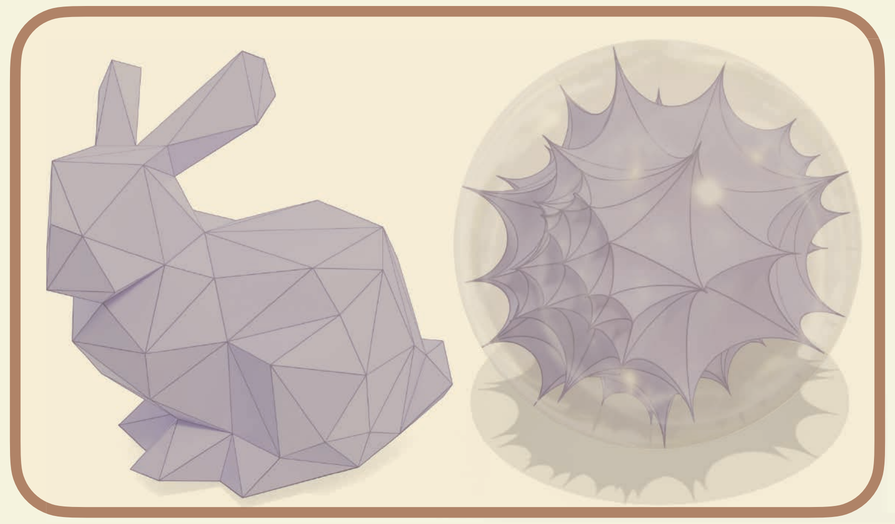
compute common subdivision

interpolate via hyperboloid

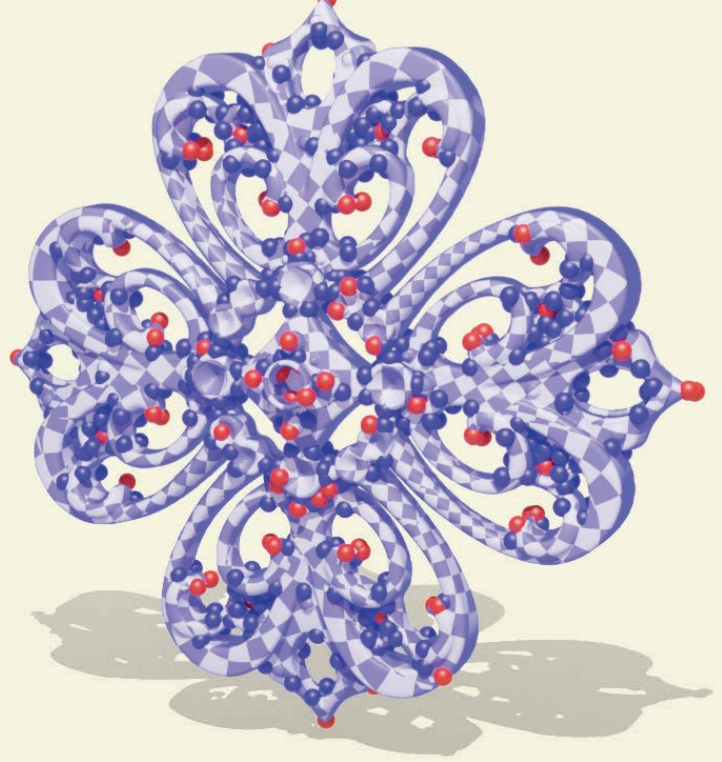


Uniformization results

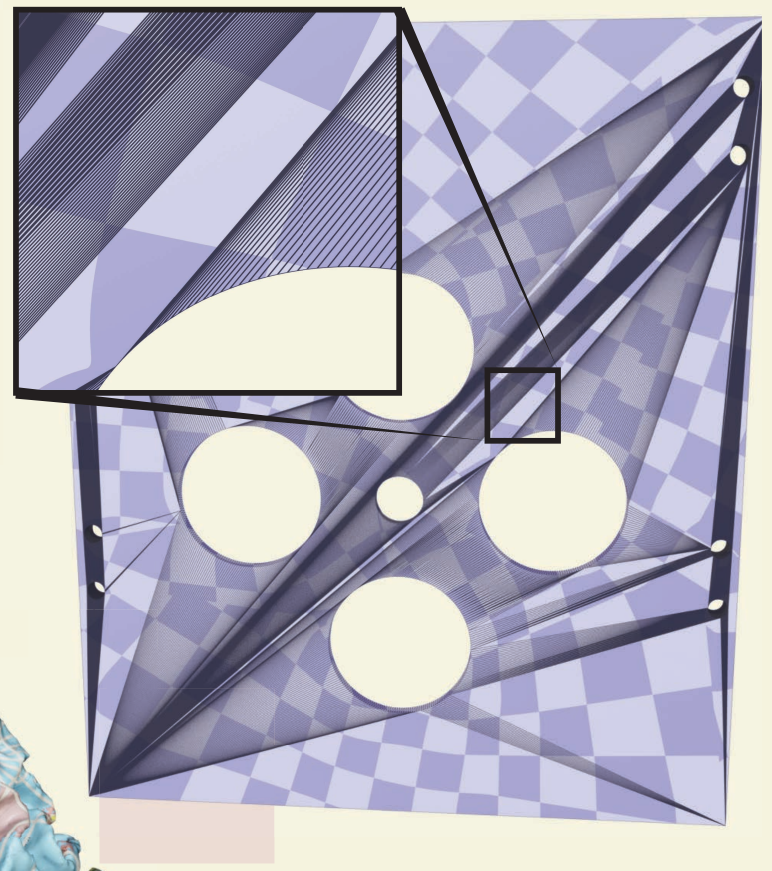
Challenging datasets



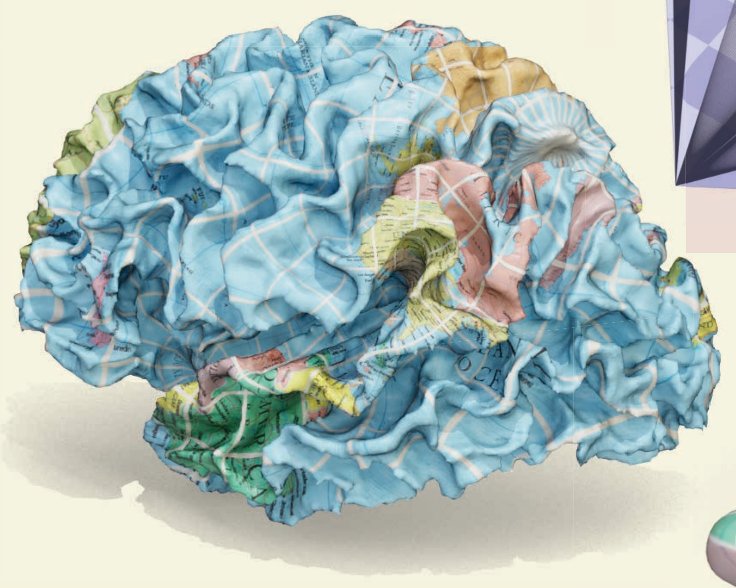
difficult cones



bad meshes



cone flattening

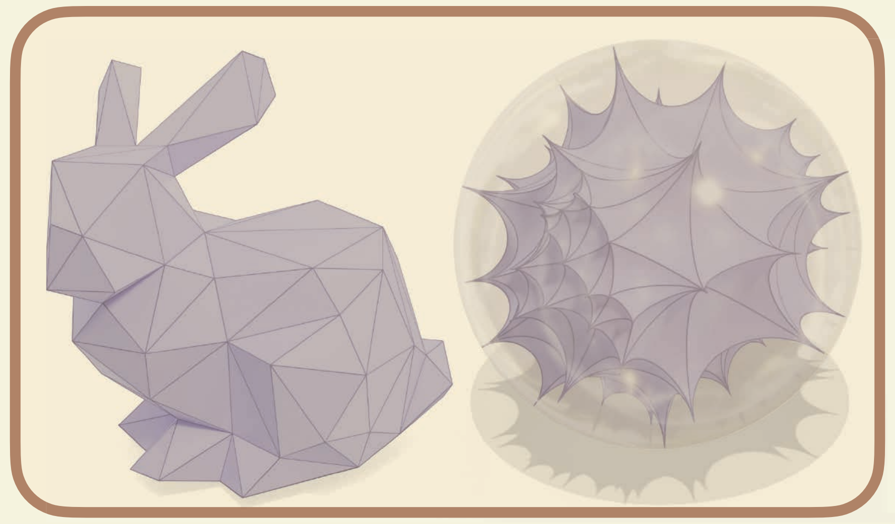


spherical uniformization

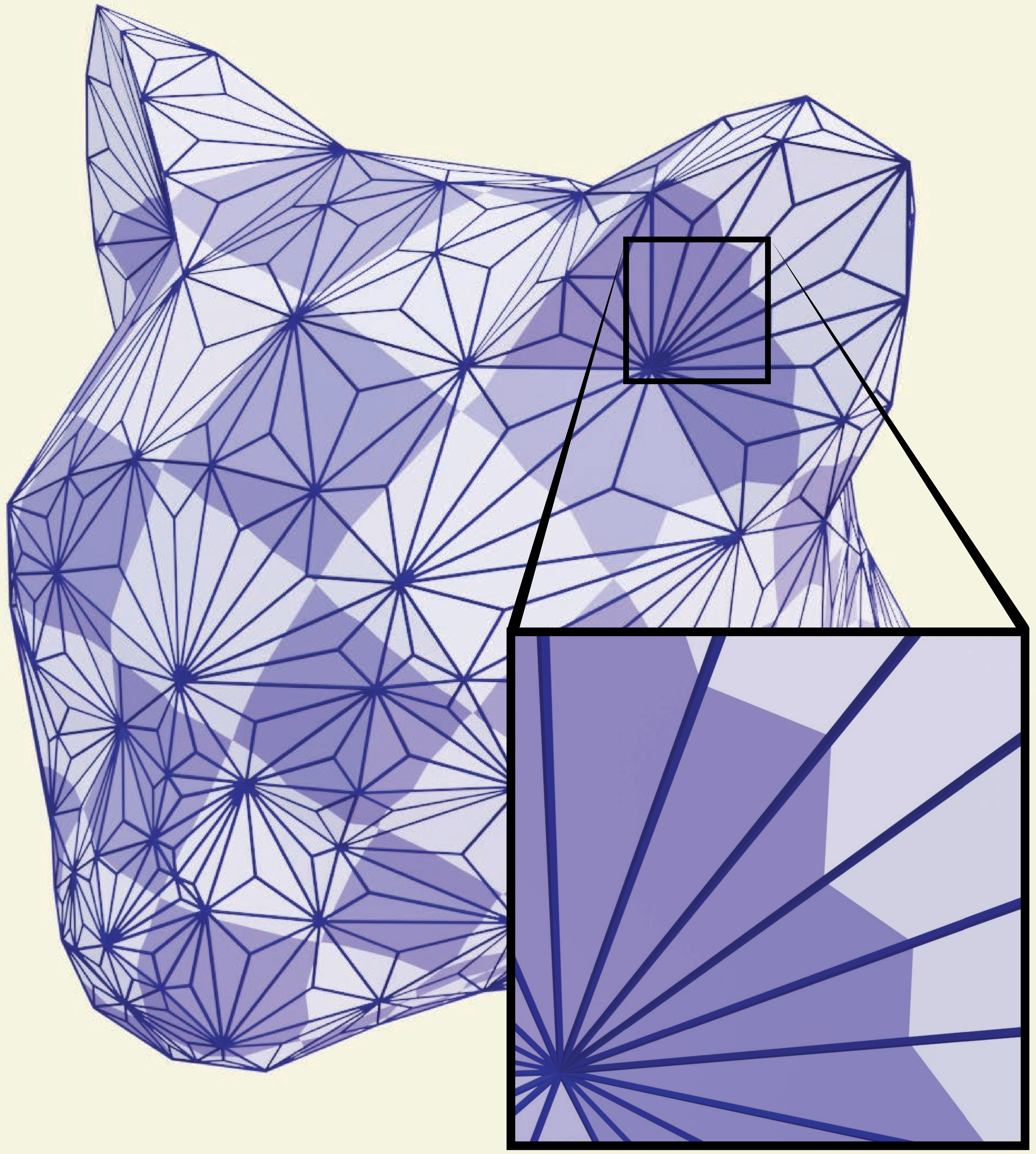
Dataset	# Models	Success rate	Average time
MPZ [Myles+ 2014]	114	100%	8s
Thingi10k [Zhou+ 2016]	32,744*	97.7%	57s†
brain scans [Yeo+ 2009]	78	100%	493s
anatomical surfaces [Boyer+ 2011]	187	100%	15s

* connected components of models from Thingi10k

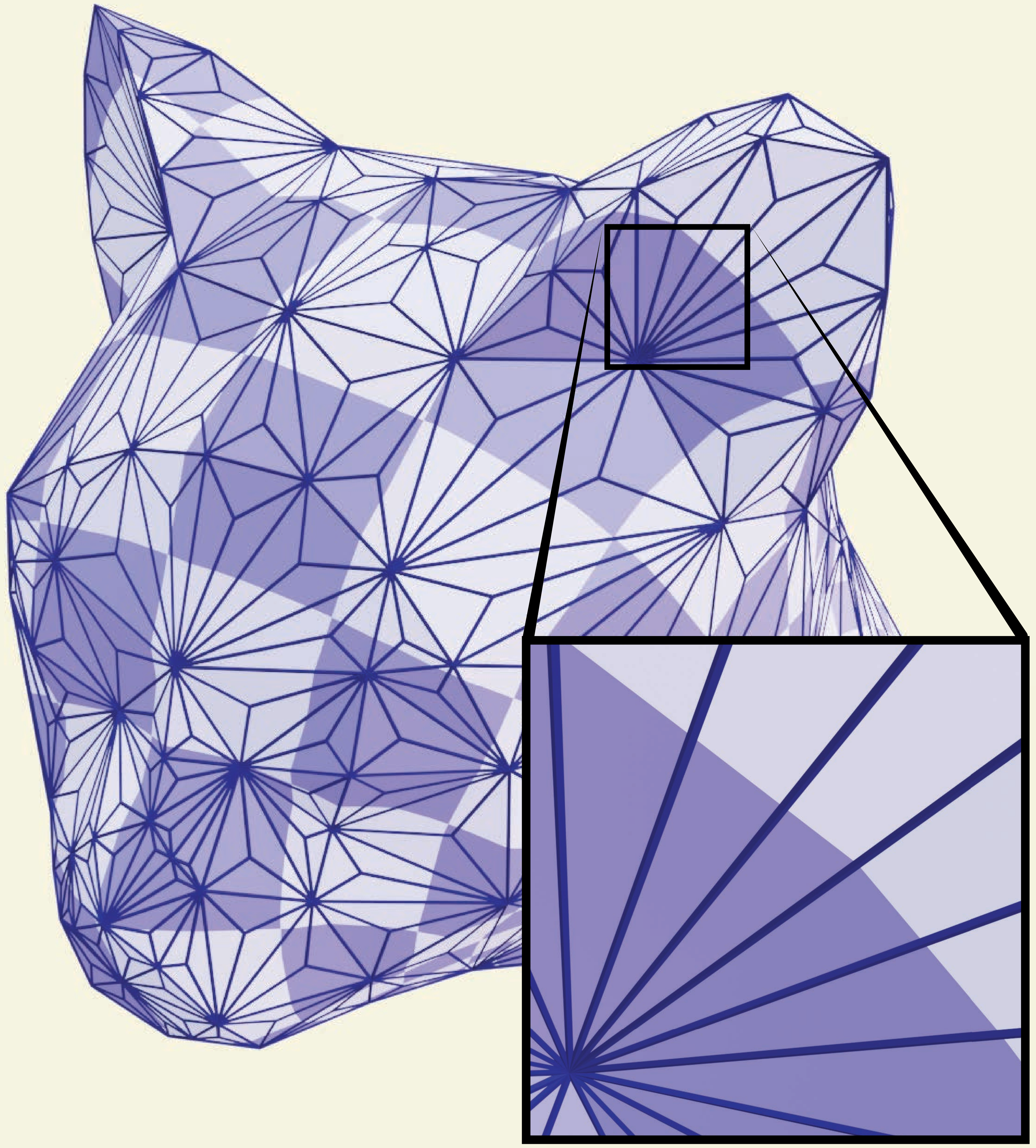
† average time on models with > 1000 vertices



Variable triangulation > fixed triangulation



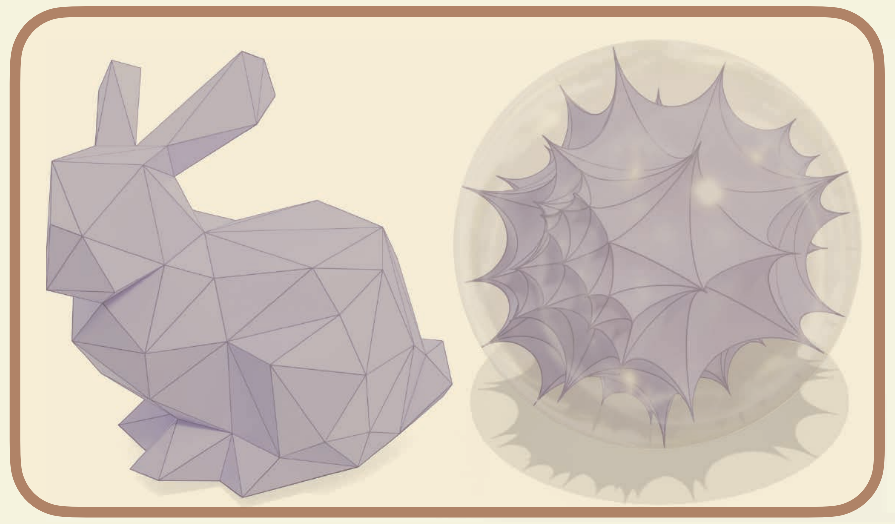
Fixed triangulation (CETM)

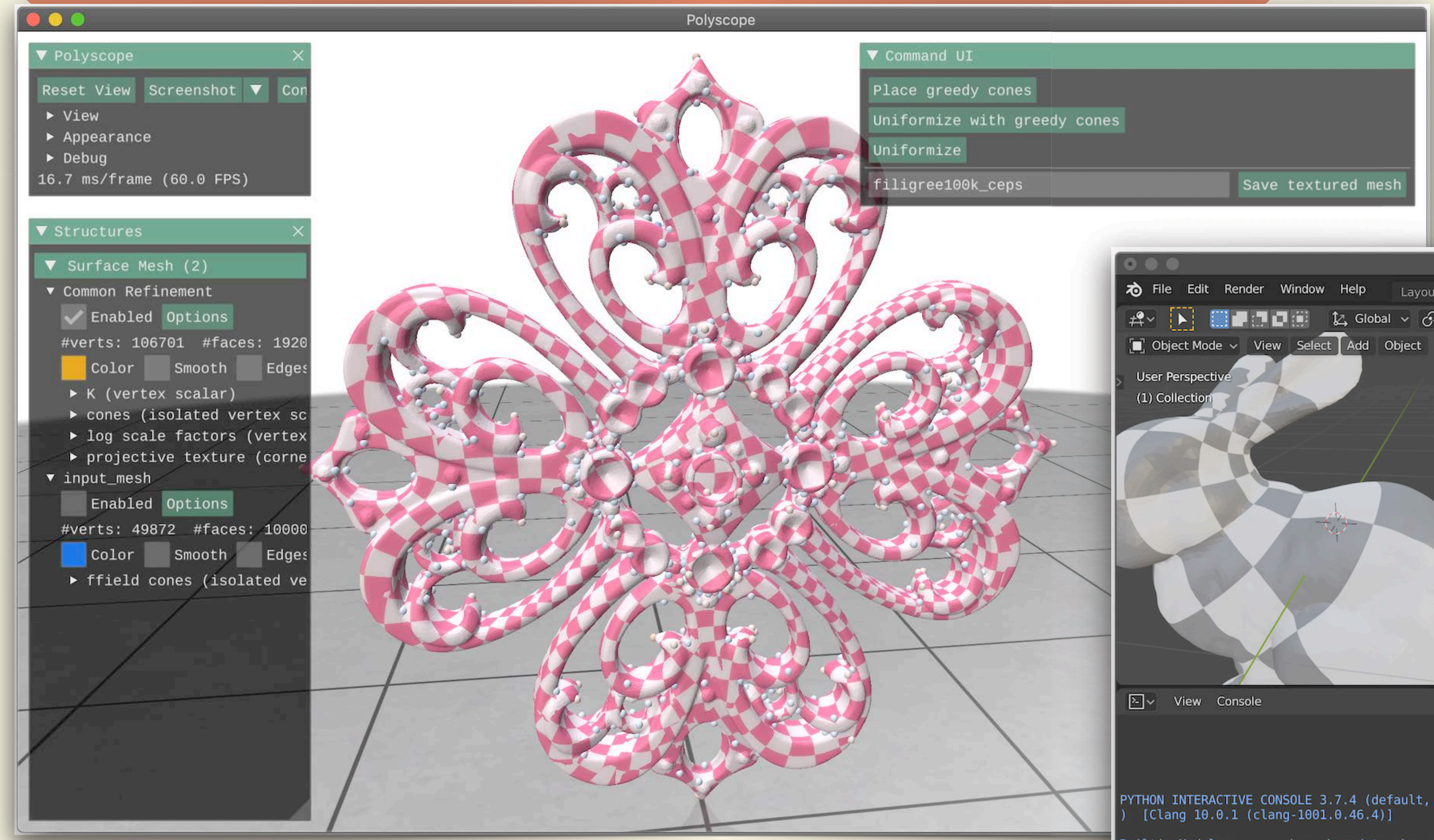


Variable triangulation (CEPS)

Boundary conditions

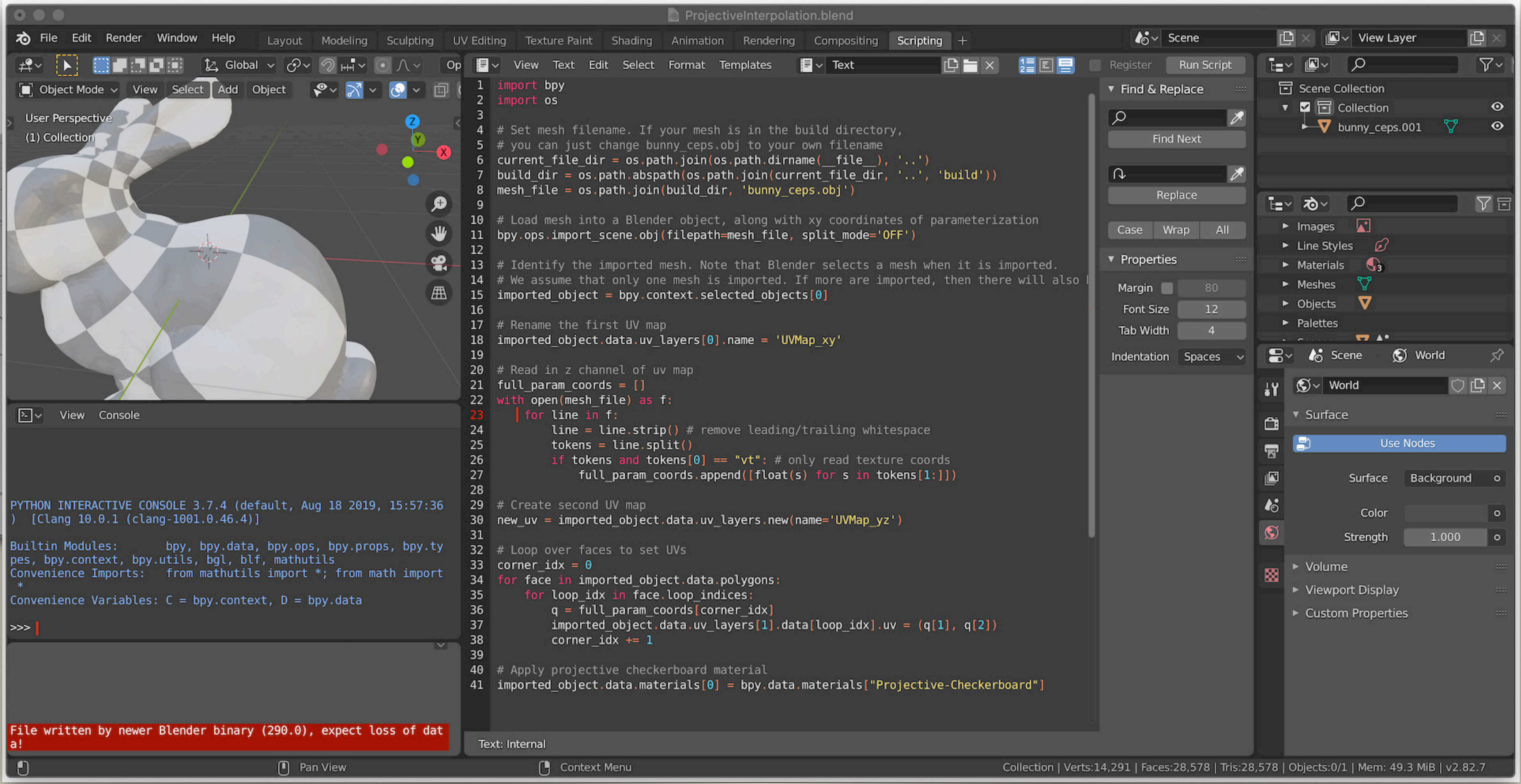
Try it out yourself





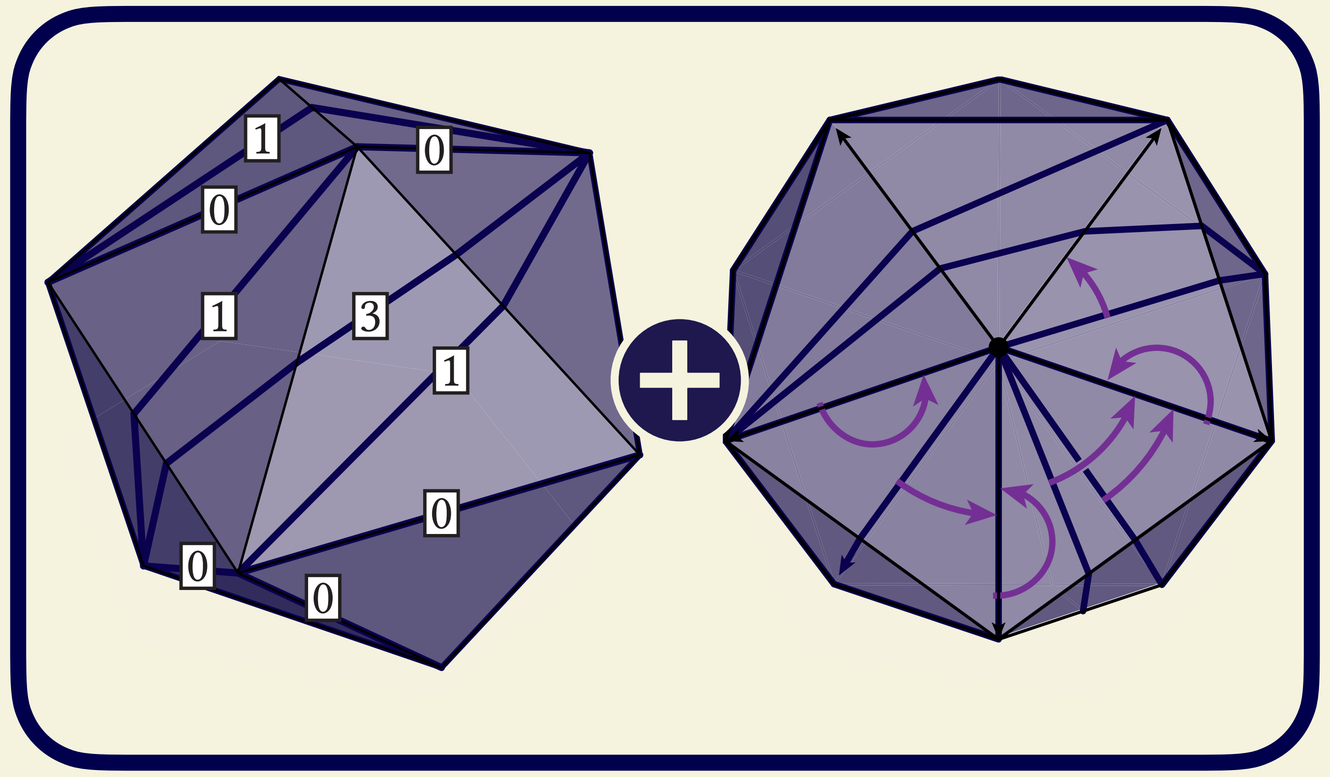
C++ application

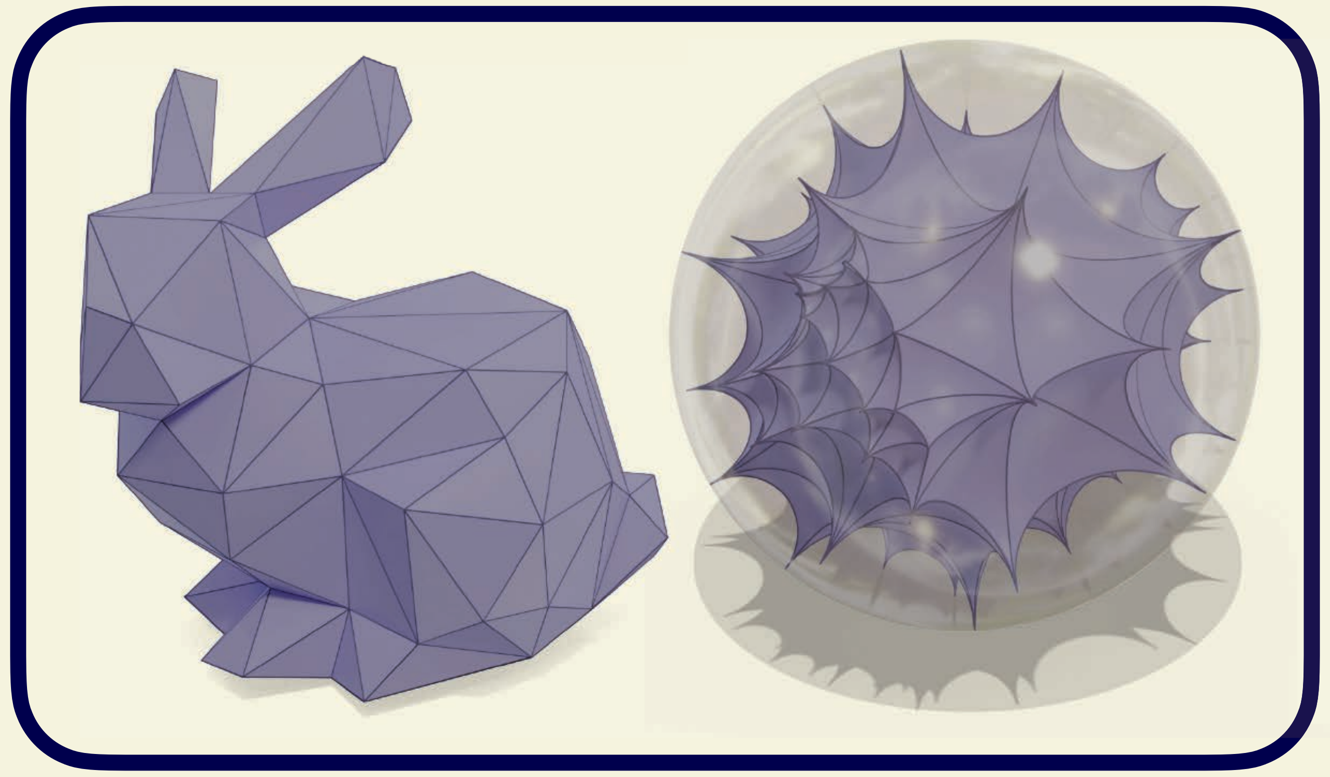
projective interpolation in Blender



https://github.com/MarkGillespie/CEPS

Thanks for listening





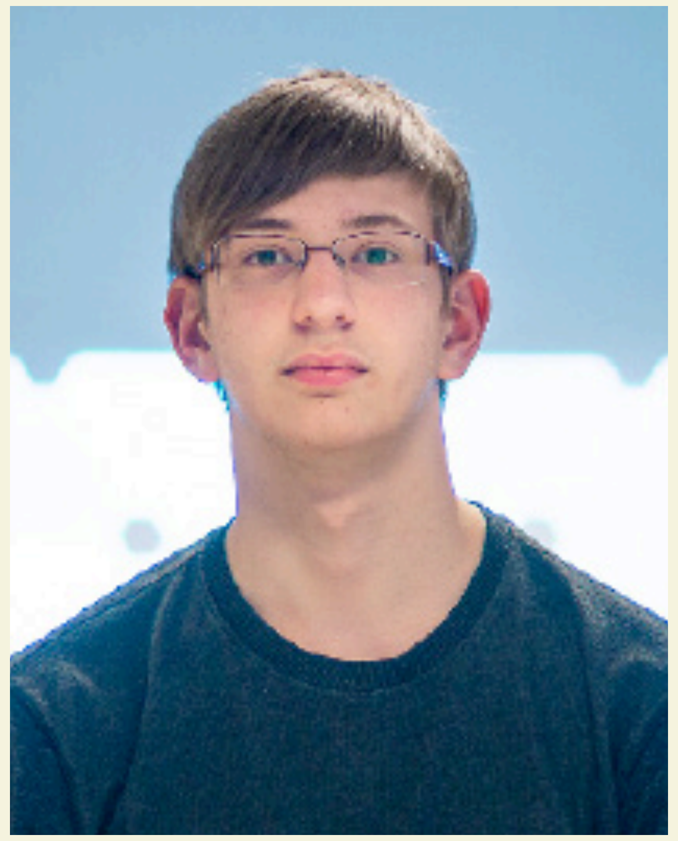
- And thank you to all of my great coauthors!

Boris Springborn

Keenan Crane

Nicholas Sharp

Hsueh-Ti Derek Liu

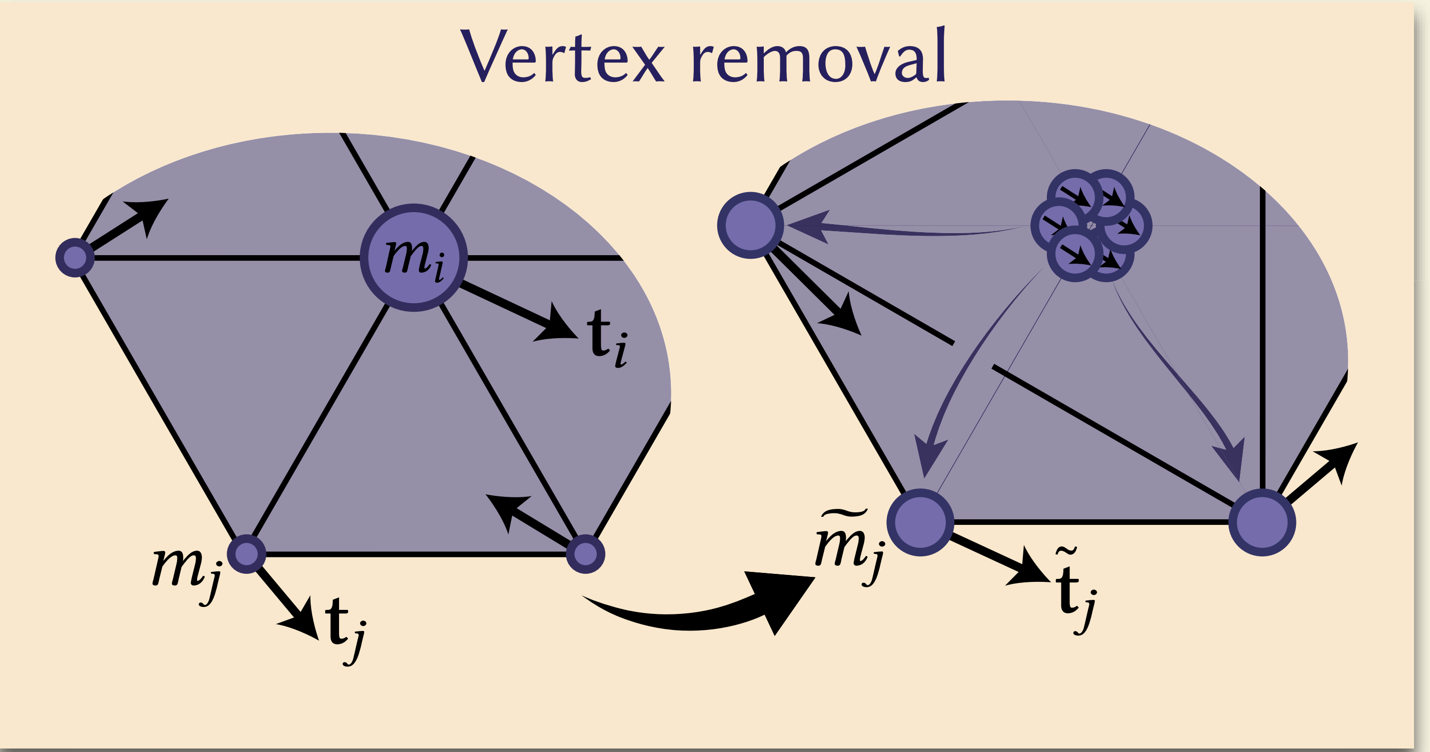


Benjamin Chislett

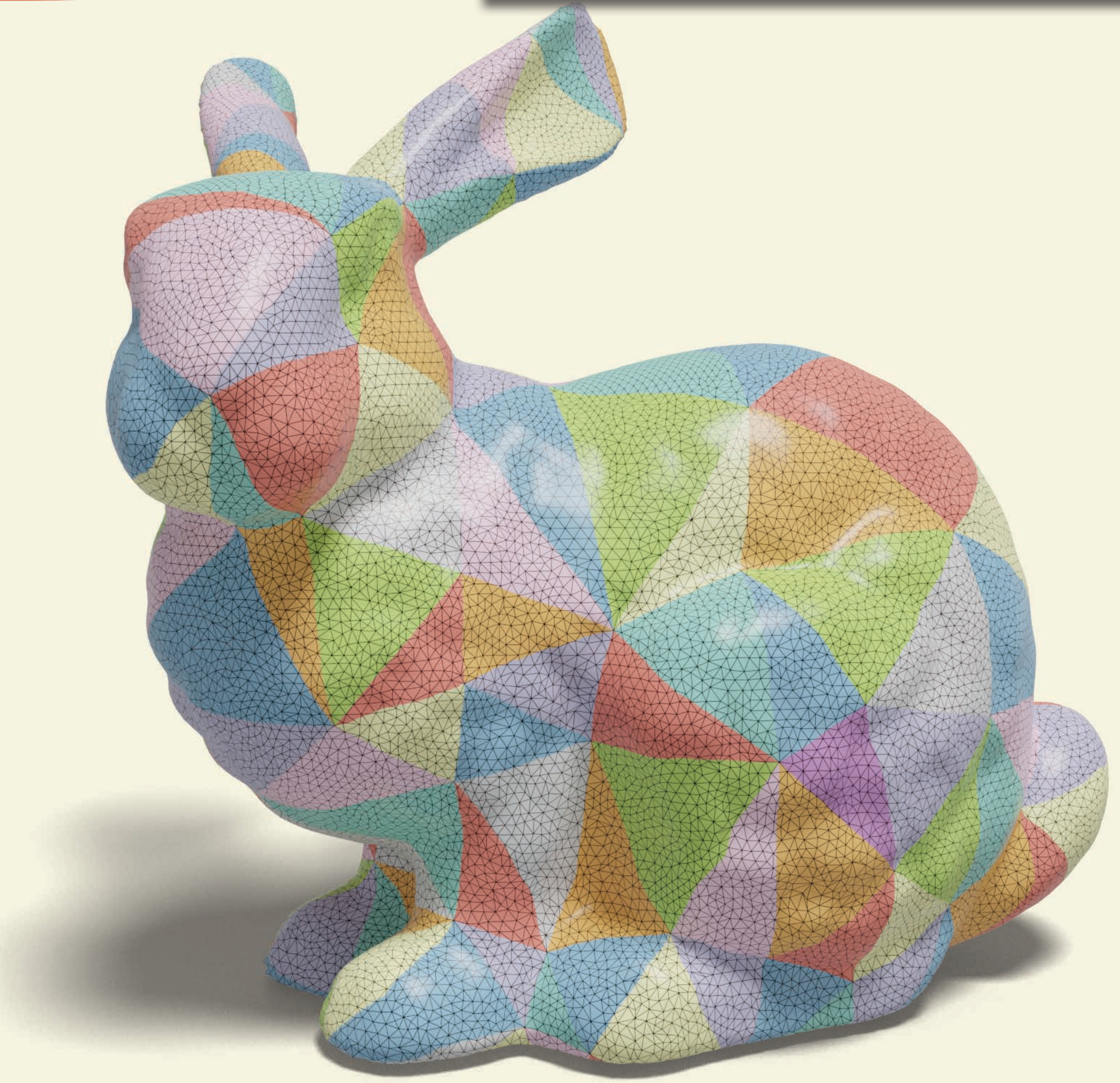
Alec Jacobson

Supplemental Slides

The importance of memory

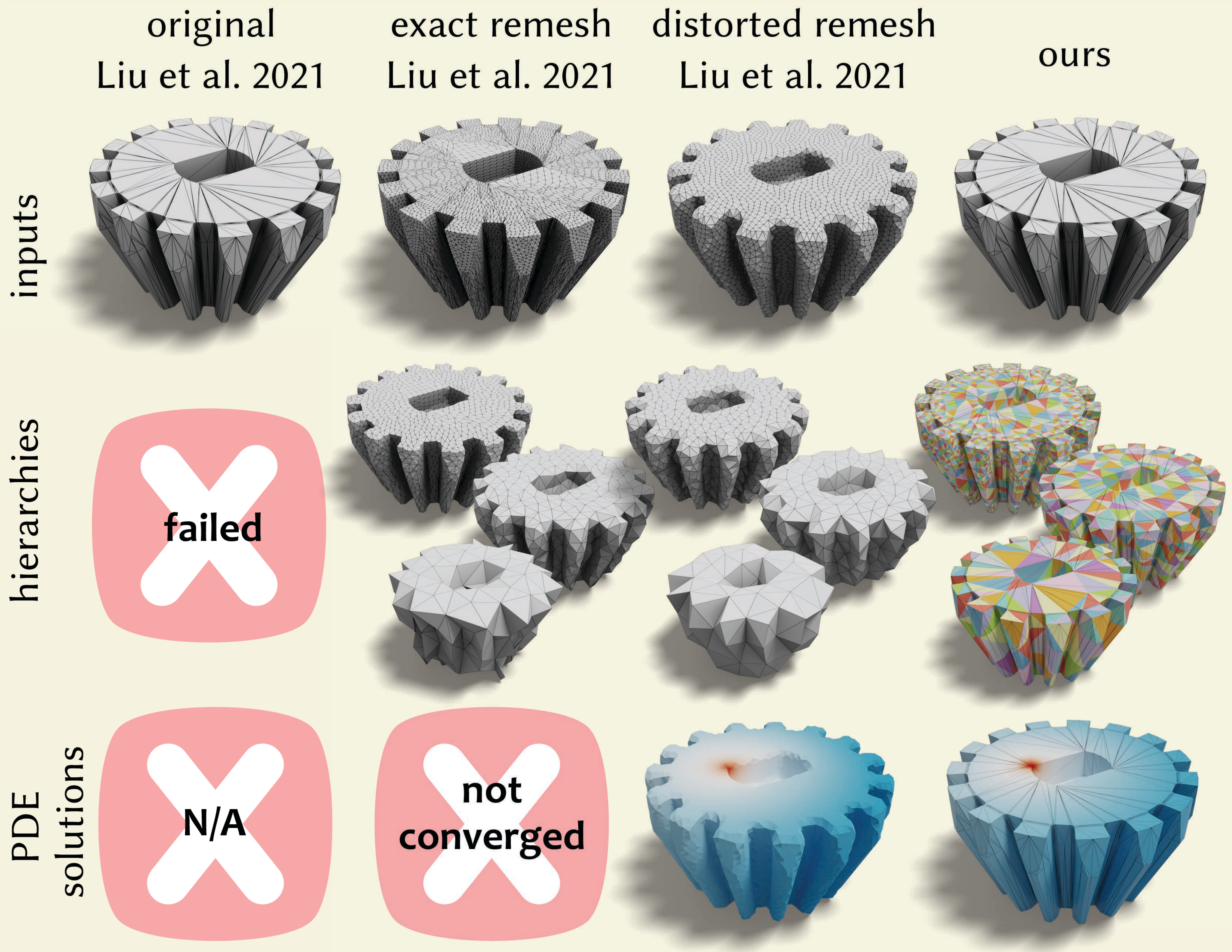


Memoryless transport cost

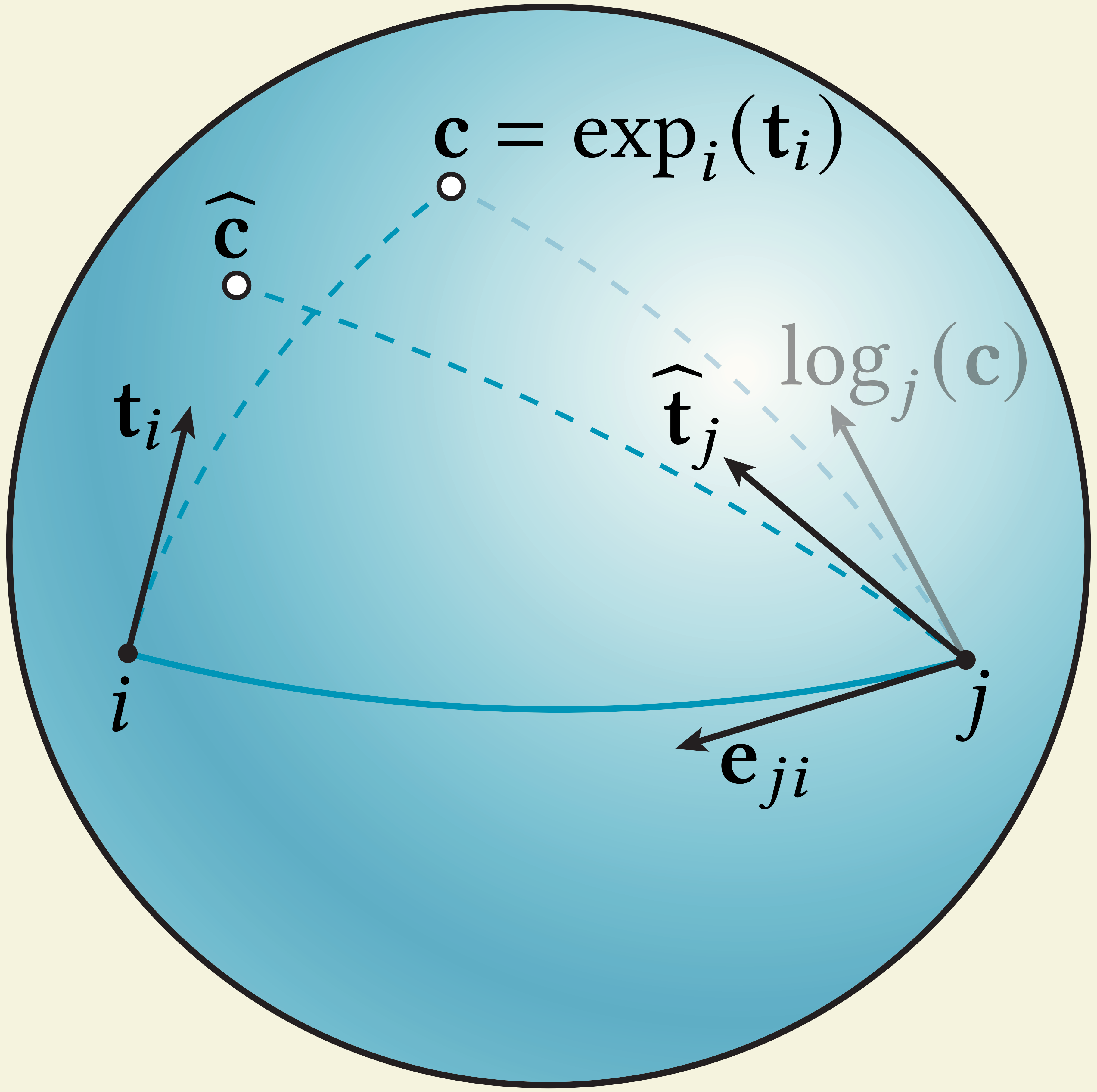


Full transport cost

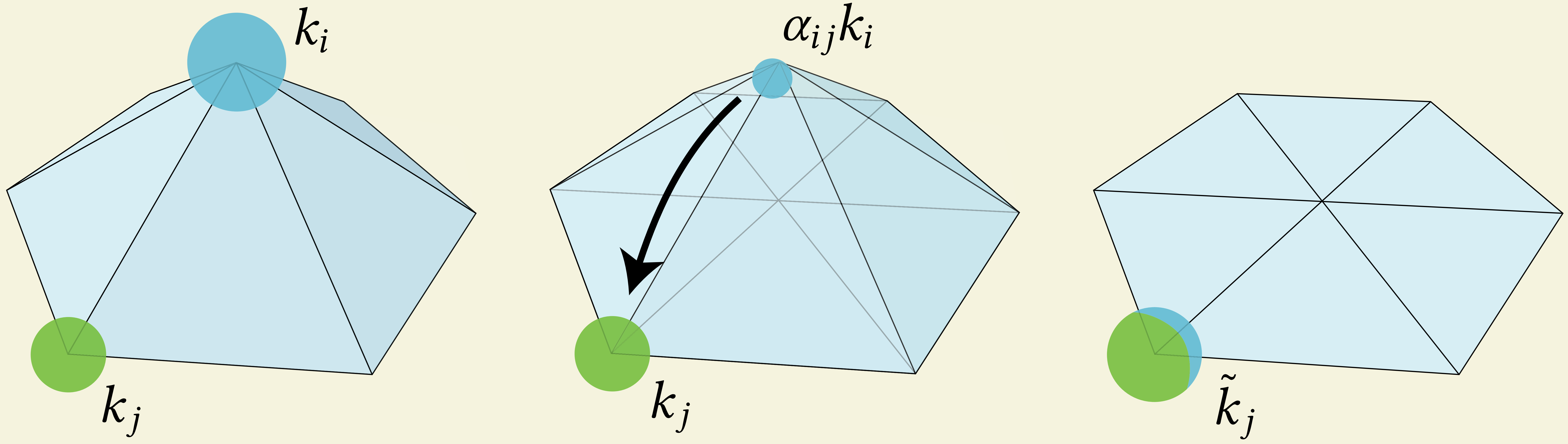
Robust hierarchies



Tangent space approximations



Signed curvature transport



$$\alpha_{ij} := \frac{|\tilde{K}_j - K_j|}{\sum_{l \in \mathcal{N}_i} |\tilde{K}_l - K_l|}.$$