

Implicit Minimal Surfaces for Bijective Correspondences

Mark Gillespie, INRIA & UNIVERSITY OF UTAH



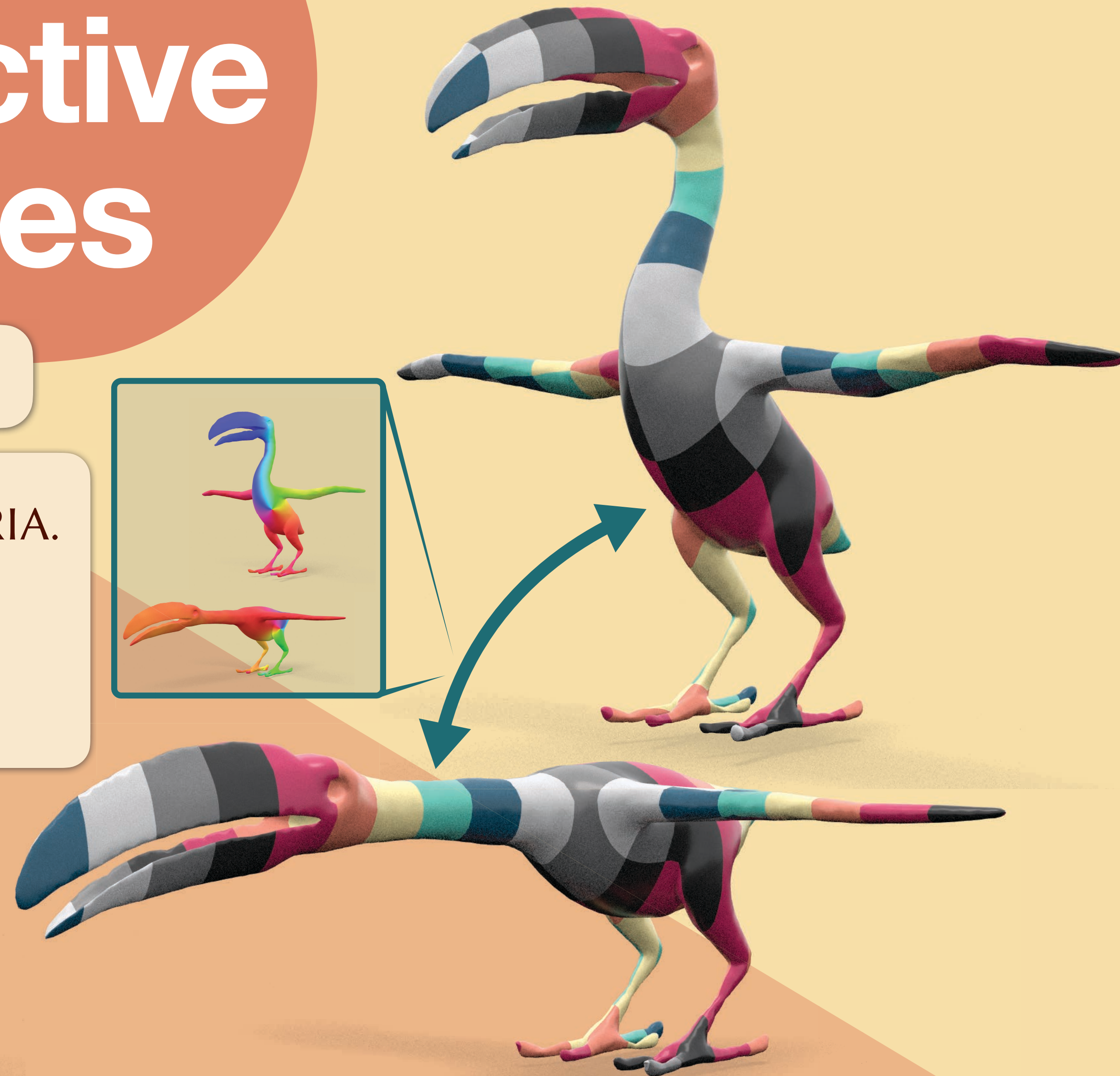
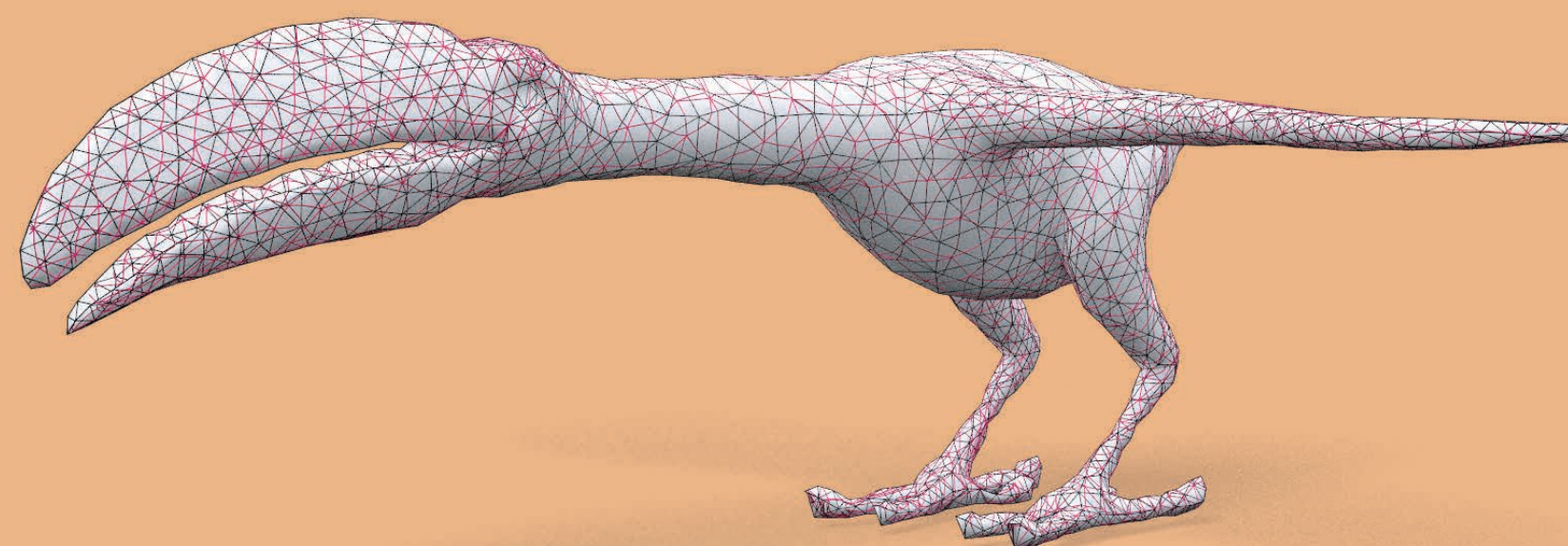
Etienne Corman, UNIVERSITÉ DE LORRAINE, CNRS, & LORIA.



Yousuf Soliman, SIDE EFFECTS SOFTWARE



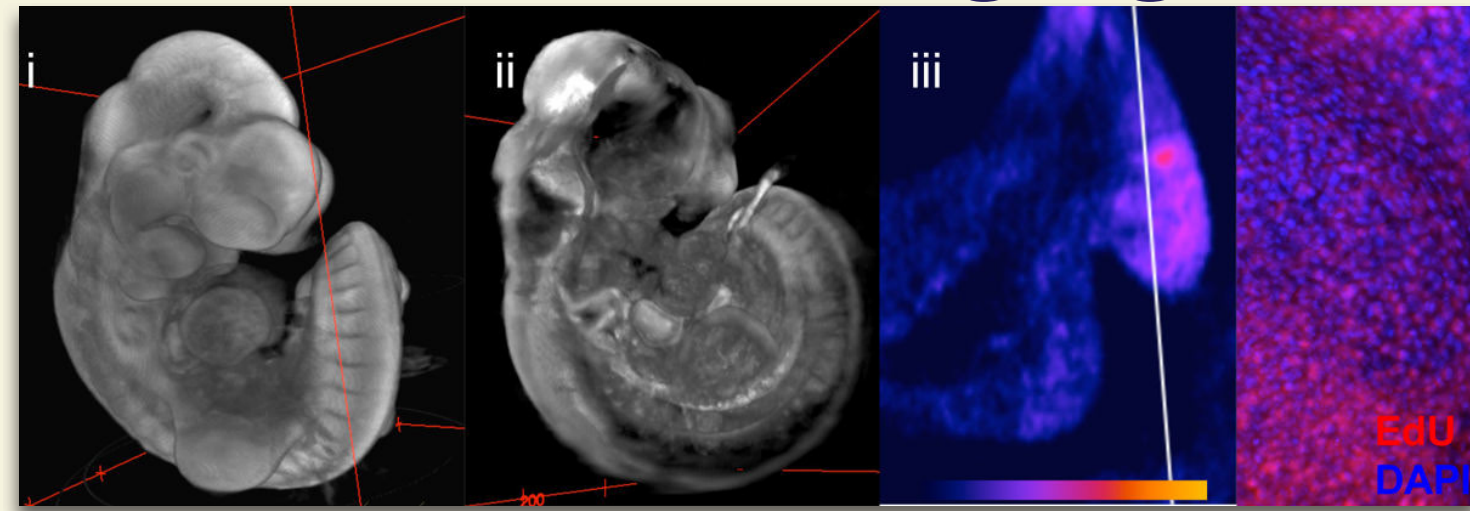
Robin Magnet, INRIA & UNIVERSITÉ PARIS CITÉ



Problem: Computing Bijections Between Surfaces

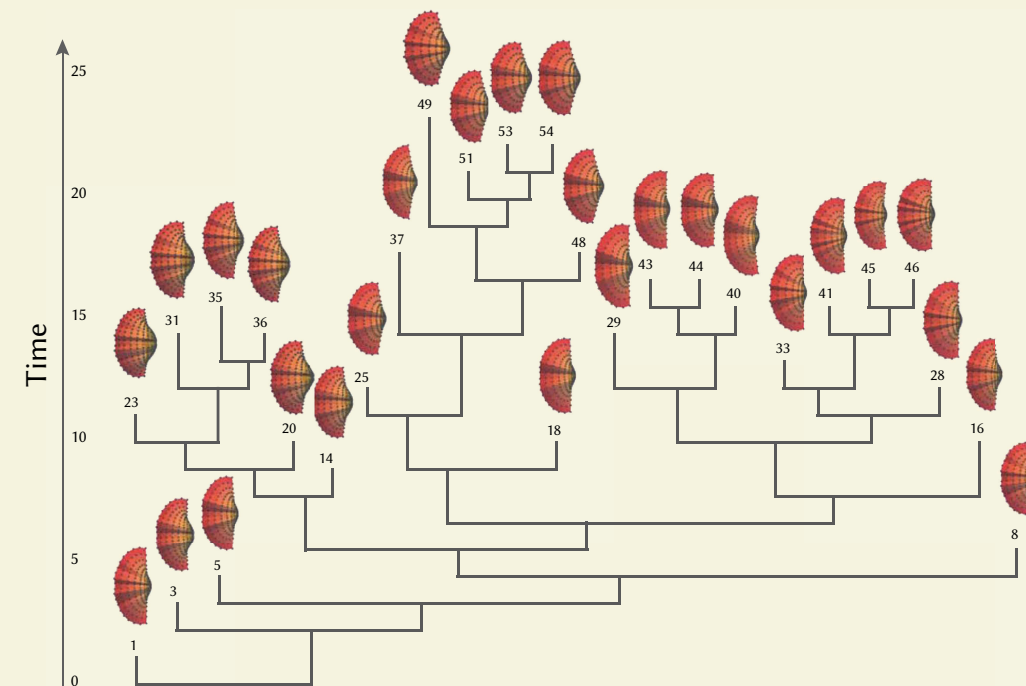
required to study data over collections of shapes

medical imaging

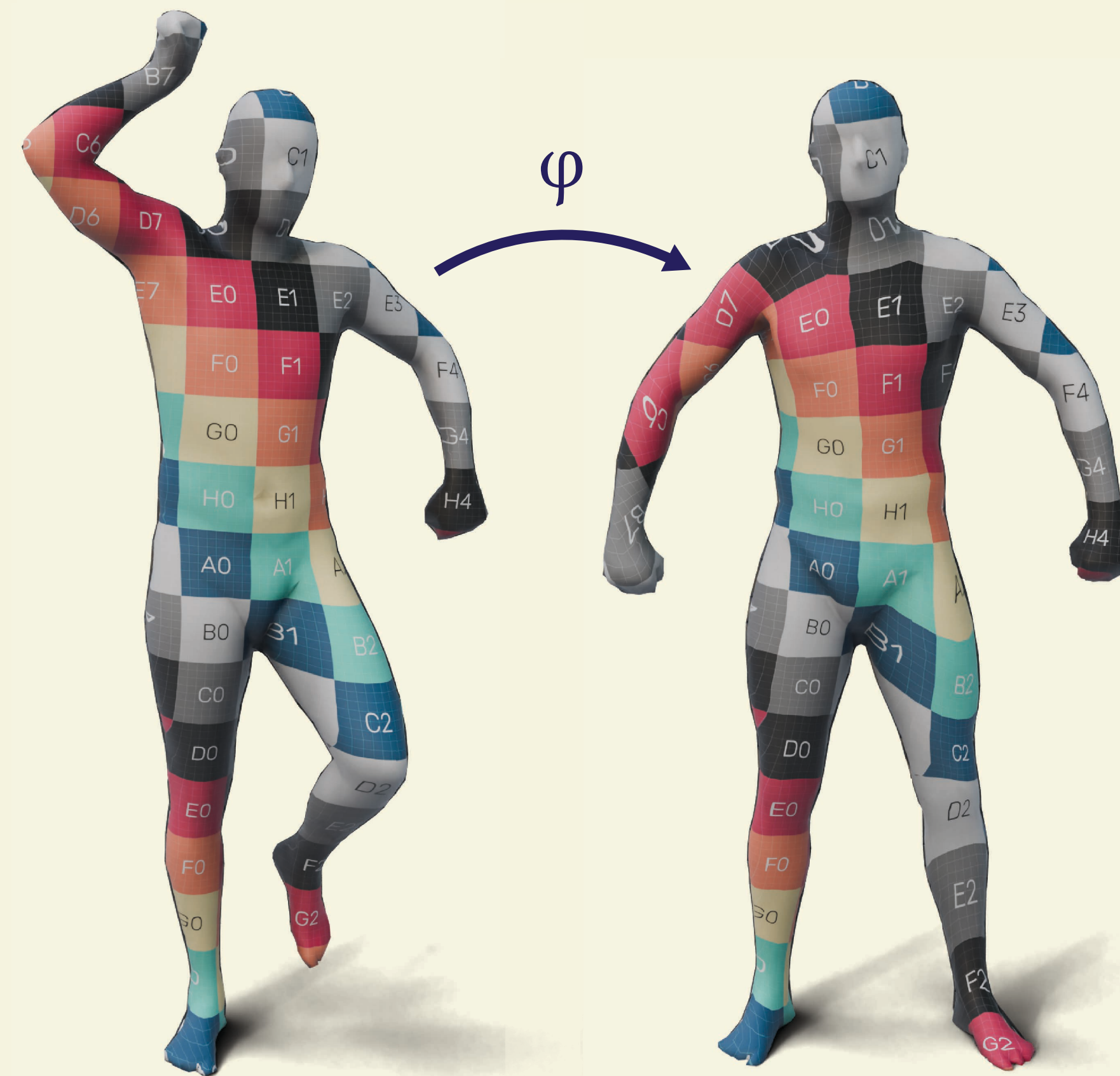


[Hallgrimsson et al. 2015]

paleontology



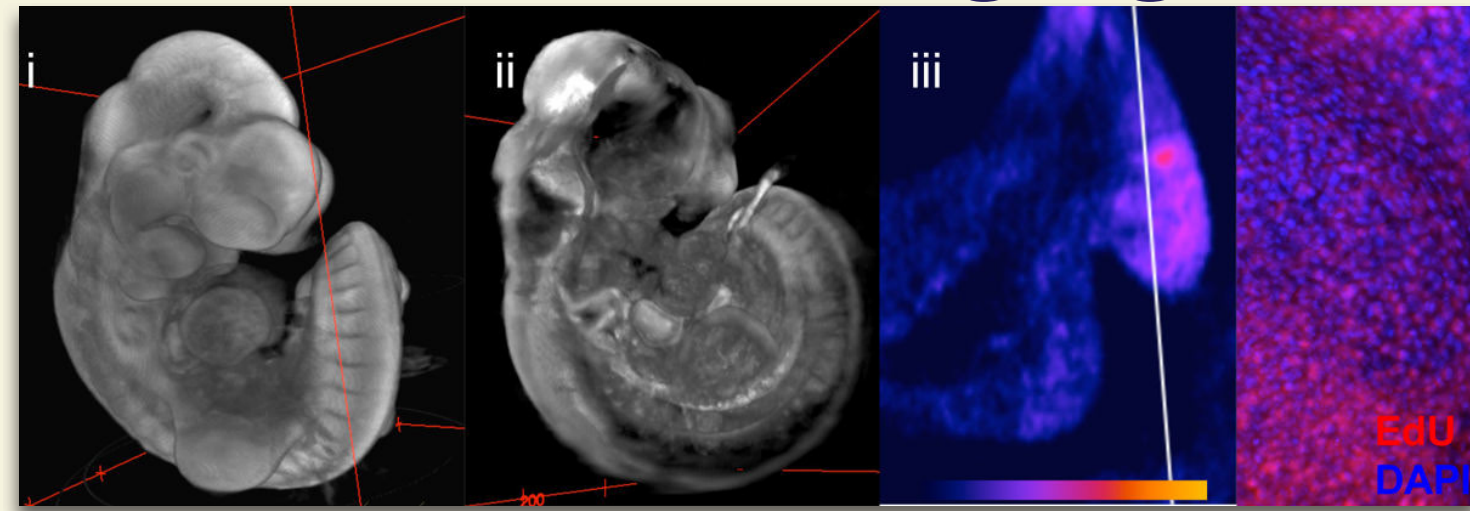
[Polly & Motz 2016]



Problem: Computing Bijections Between Surfaces

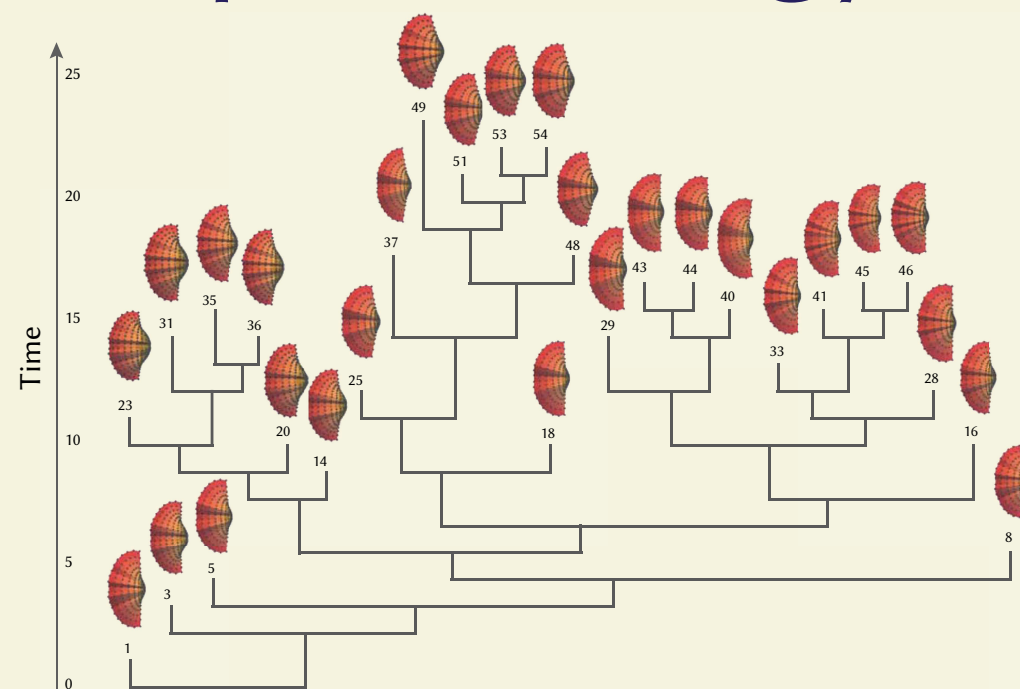
required to study data over collections of shapes

medical imaging

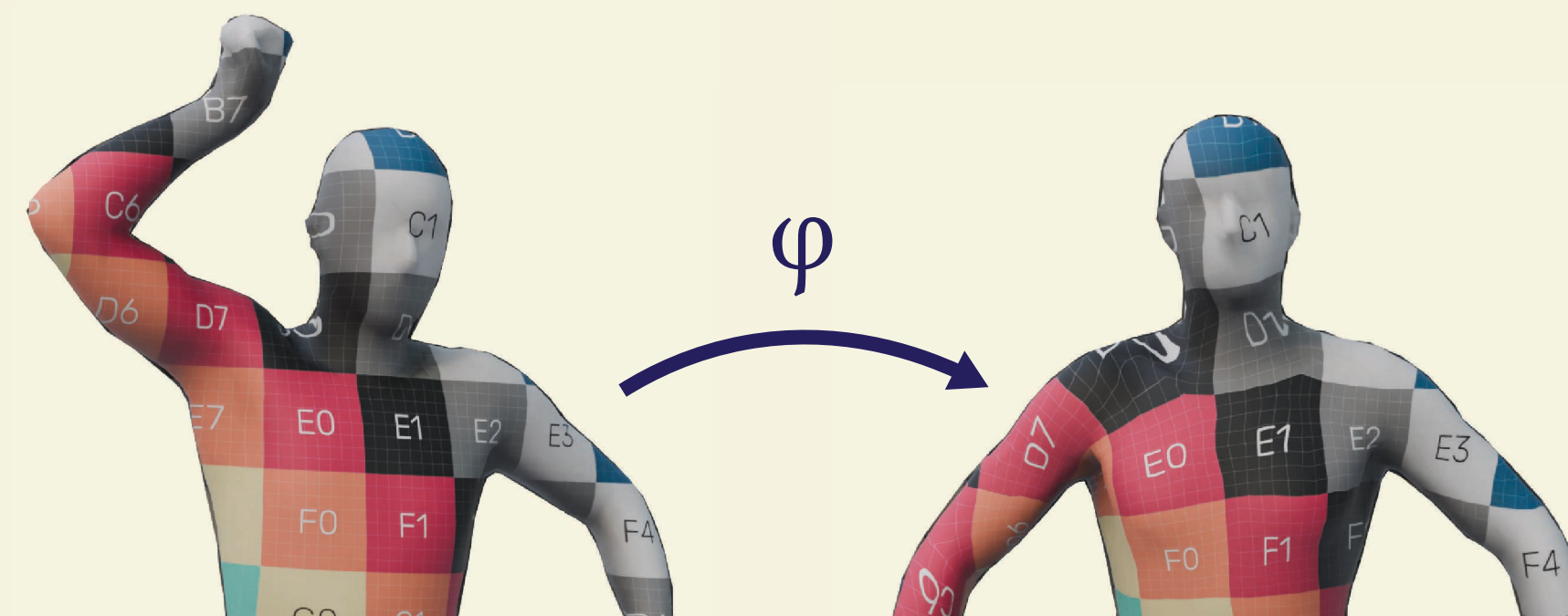


[Hallgrimsson et al. 2015]

paleontology

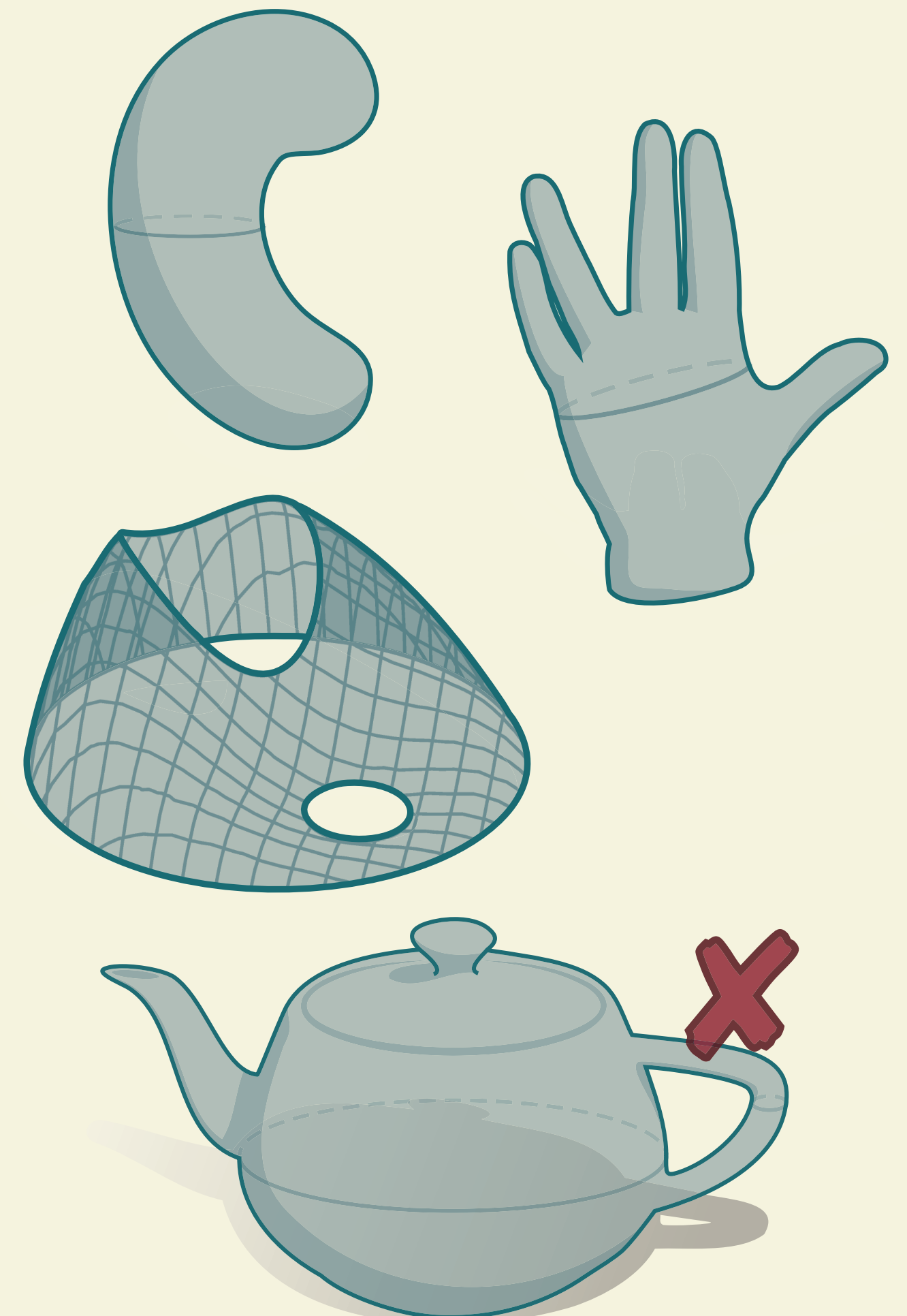


[Polly & Motz 2016]



SIMPLIFYING ASSUMPTION

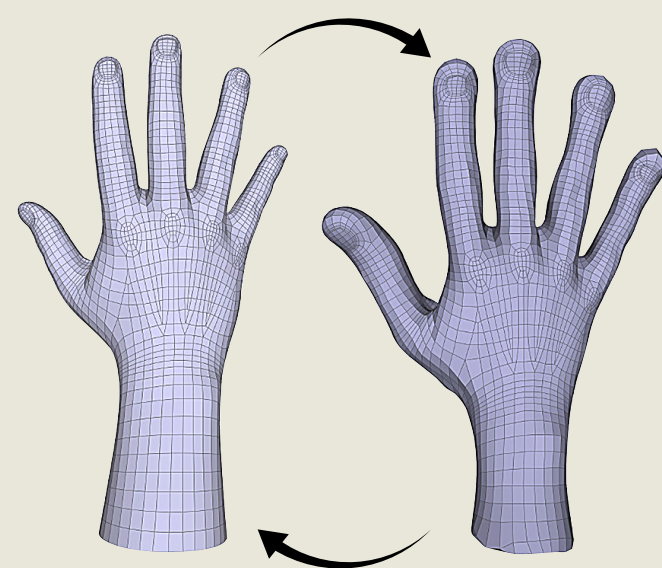
Surfaces are topological spheres
(will discuss boundary near the end)
but no handles (yet)



Existing Algorithms

SHAPE DEFORMATION

[Beg et al. 2005; Sharf et al. 2006; Huang et al. 2008; Li et al. 2008b; Tam et al. 2012; Eisenberger et al. 2019, 2020; Ezuz et al. 2019a]



Reversible Harmonic Maps (RHM)

[Ezuz et al. 2019b]

straightforward algorithms, *but*
requires good initialization
often requires user
intervention

“FUZZY” RELAXATIONS

[Ovsjanikov et al. 2012, 2017; Ren et al. 2021; Litany et al. 2016; Solomon et al. 2012; Mandad et al. 2017; Brifault et al. 2025]



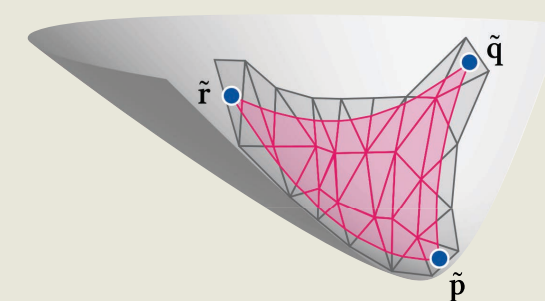
ZoomOut

[Melzi et al. 2019]

simpler to optimize, *but*:
can collapse features
may struggle with
symmetries

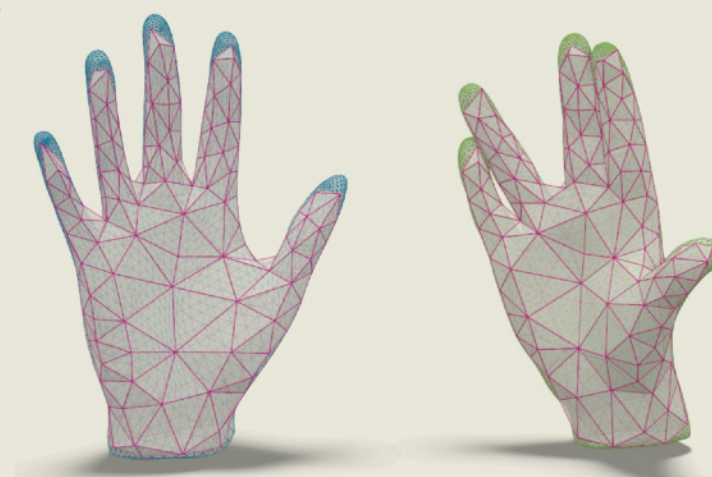
EXPLICIT MESH MAPS

[Kanai et al. 1997; Litke et al. 2005; Aigerman et al. 2014–2017; Schreiner et al. 2004; Kim et al. 2011; Michikawa et al. 2001; Peng & Timalaena 2016; Kraevoy & Sheffer 2004; Takayama 2022]



Inter-Surface Maps via Constant-Curvature Metrics (ISM)

[Schmidt et al. 2020]



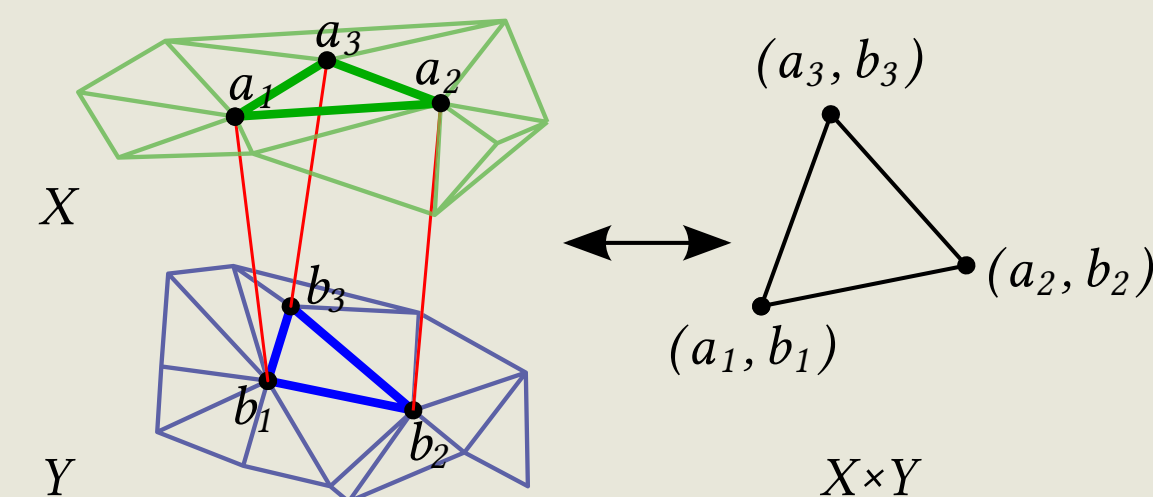
Surface Maps via Adaptive Triangulations

[Schmidt et al. 2023]

compute true mesh map, which is
bijective by construction, *but*:
requires bijective initialization
requires complicated mesh mutation

PRODUCT SPACE SEARCHES

[Vestner et al. 2017; Windheuser et al. 2011b; Roetzer et al. 2022; Amrani et al. 2025; Roetzer and Bernard 2025]



Geometrically consistent
elastic matching of 3D
shapes: A linear
programming solution

[Windheuser et al. 2011a]

stronger guarantees (e.g.
orientation preservation), *but*:
large-scale combinatorial
optimization problems (e.g.
integer programs)

Existing Algorithms

EXPLICIT MESH MAPS

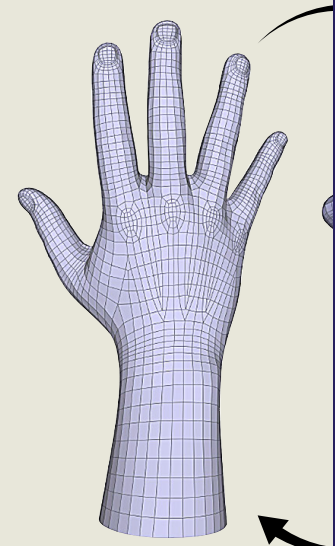
[Kanai et al. 1997; Litke et al. 2005; Aigerman et al. 2014–2017; Schreiner

PRODUCT SPACE SEARCHES

[Vestner et al. 2017; Windheuser et al. 2022; ... and

SHAPE DEFORMATIONS

[Beg et al. 2005; ... Huang et al. 2008; ... Tam et al. 2012; ... 2019, 2020; Ezuz



Our implicit product space method combines these strengths:

- fully intrinsic, but preserves orientation
- exact map between meshes without mesh mutation
- optimization can "untangle" low-quality, non-bijective initializations
- built on familiar operators from tangent vector field processing, (which can be used in factored form to avoid constructing explicit operators on the full product space)
- naturally supports landmark curves in addition to landmark points

$b_3)$
 (a_2, b_2)
 $X \times Y$
istent
of 3D
r
tion
2011a]

Reversible Harmonic Maps (RHM)

[Ezuz et al. 2019b]

straightforward algorithms, *but*
requires good initialization
often requires user
intervention

ZoomOut

[Melzi et al. 2019]

simpler to optimize, *but*:
can collapse features
may struggle with
symmetries

Surface Maps via Adaptive Triangulations

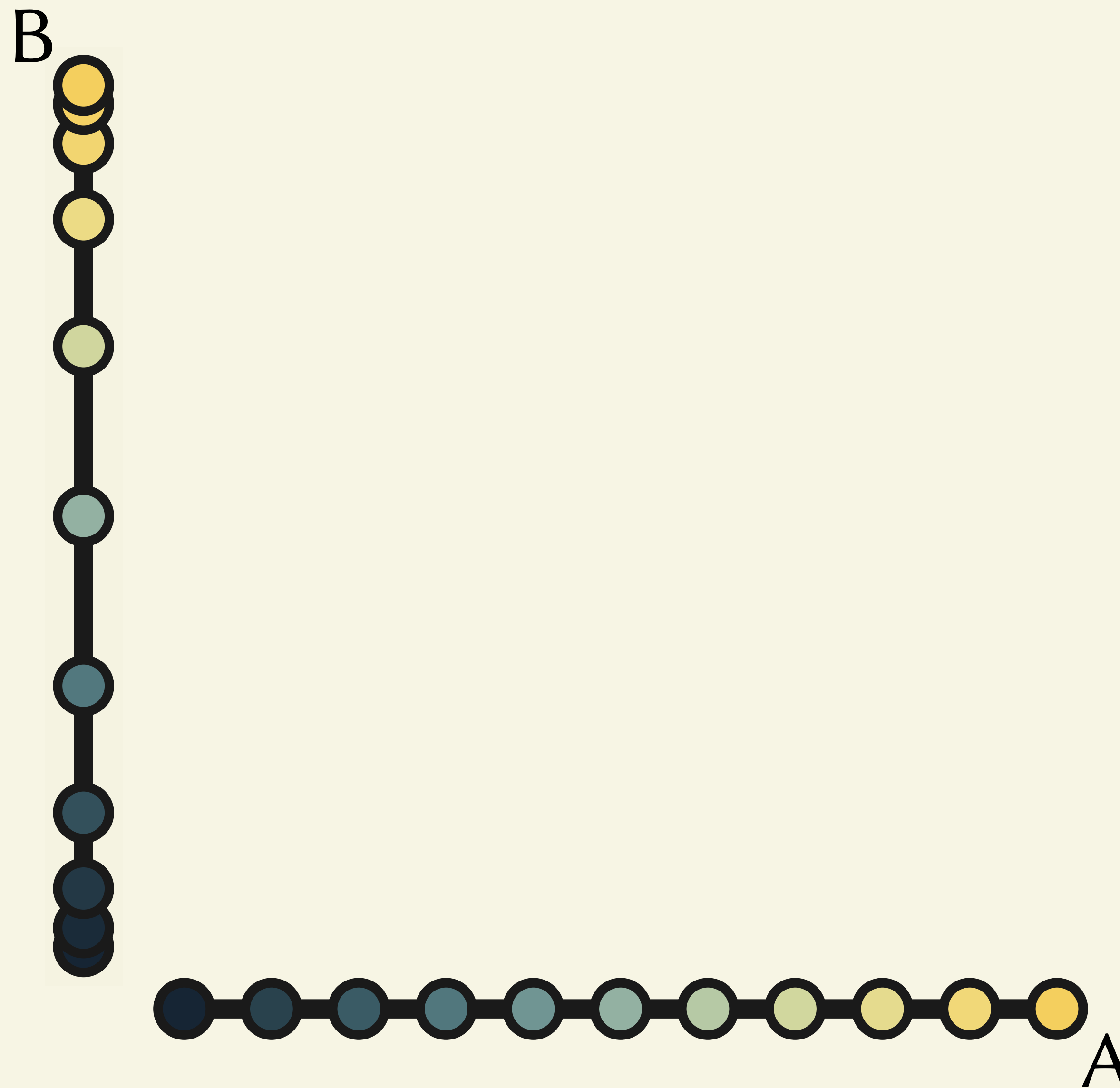
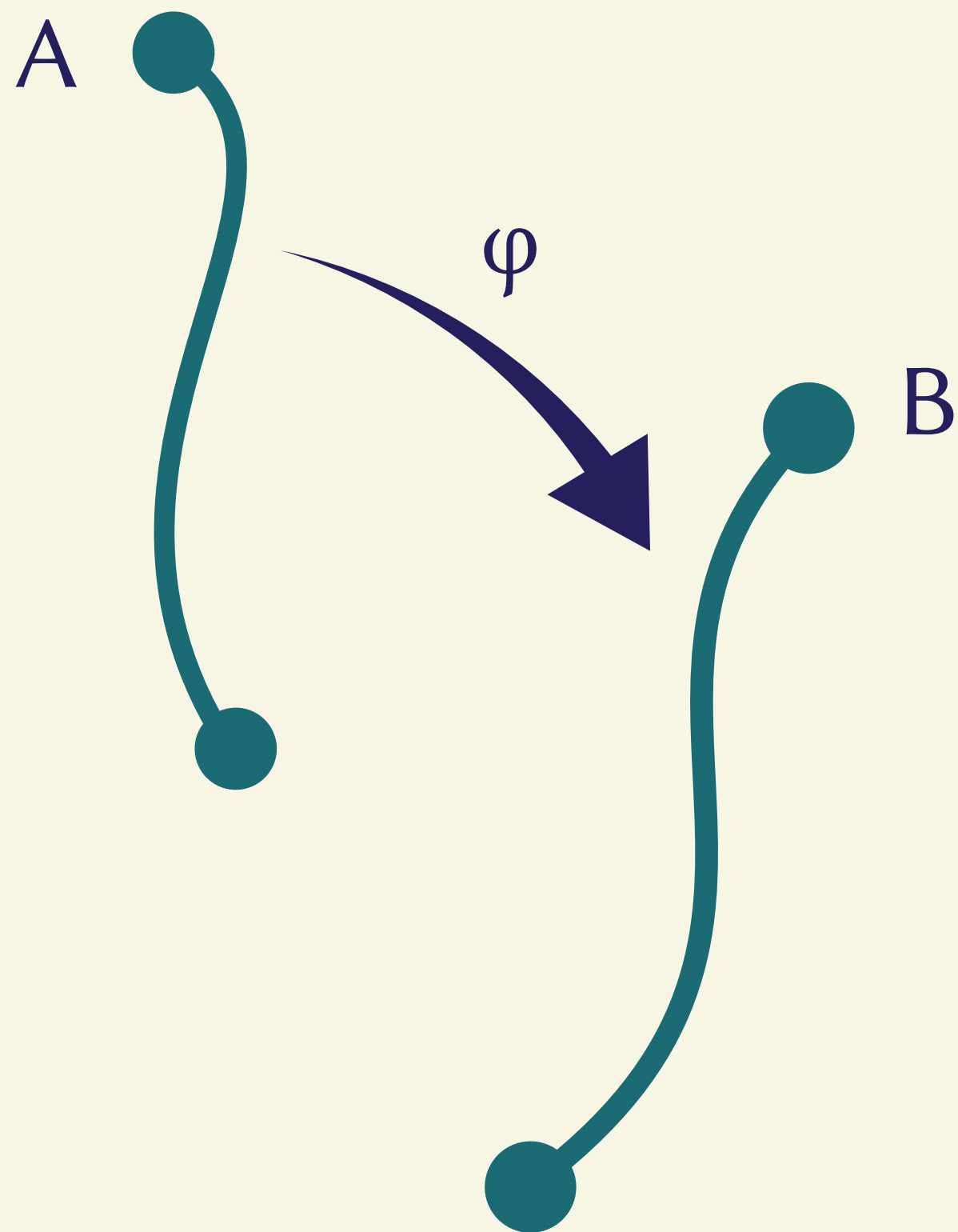
[Schmidt et al. 2023]

compute true mesh map, which is
bijective by construction, *but*:
requires bijective initialization
requires complicated mesh mutation

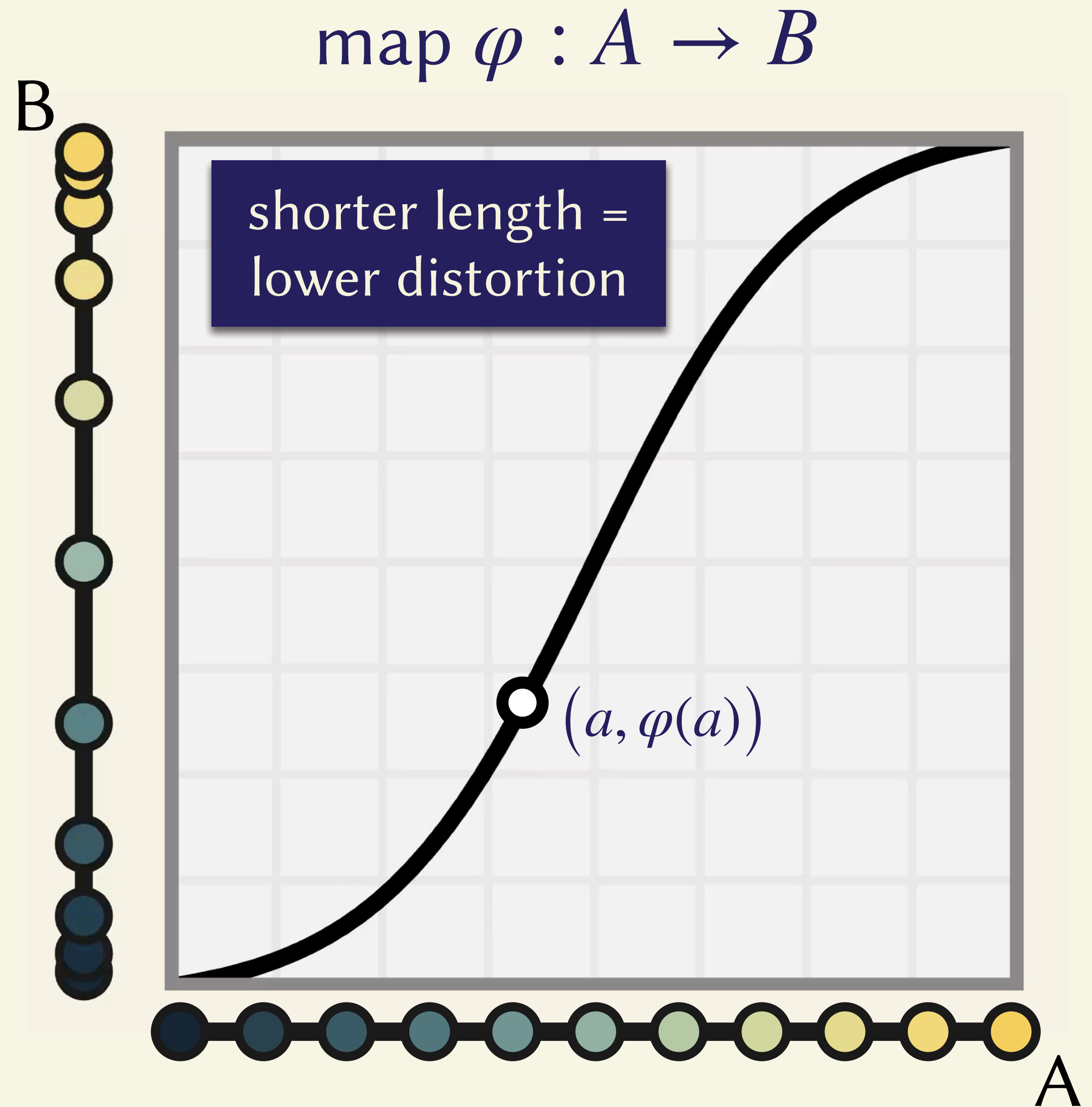
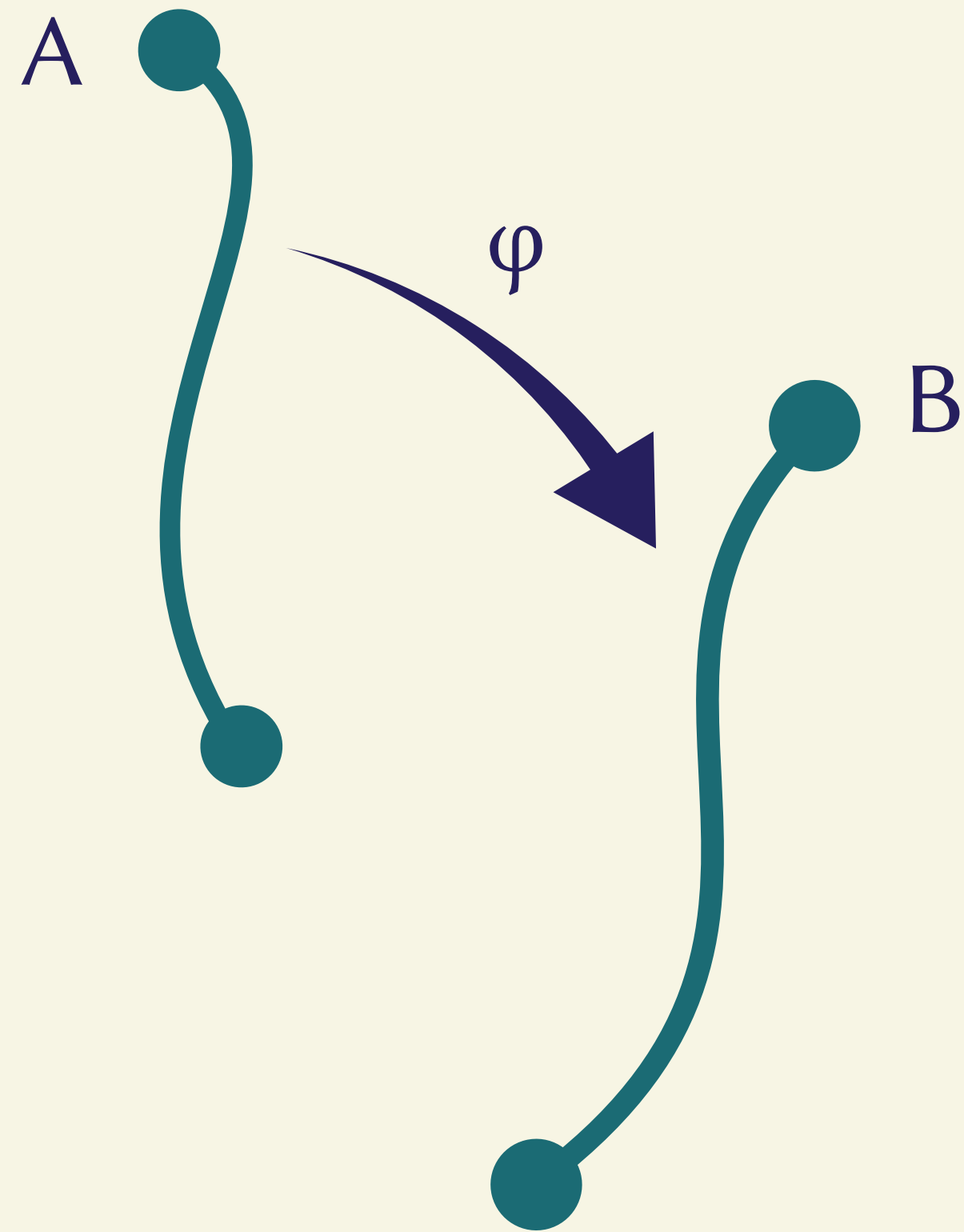
stronger guarantees (e.g.
orientation preservation), *but*:

large-scale combinatorial
optimization problems (e.g.
integer programs)

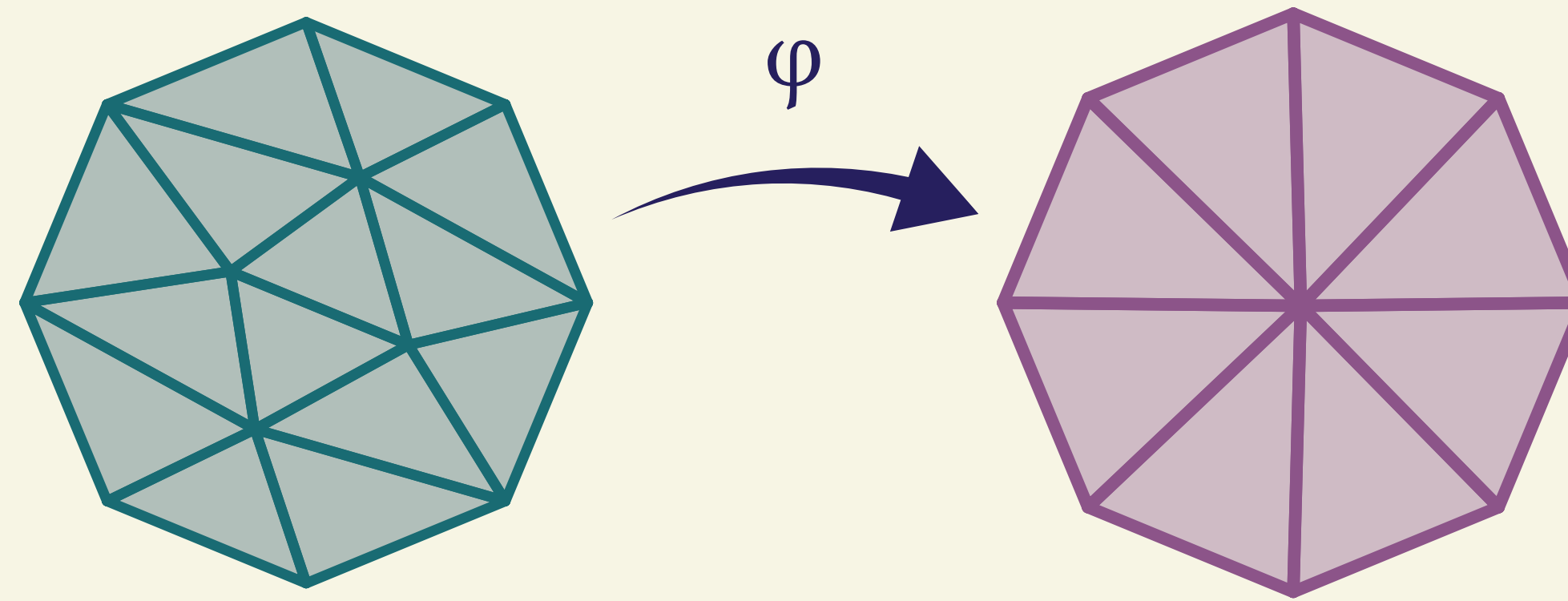
Example: 1D Curves



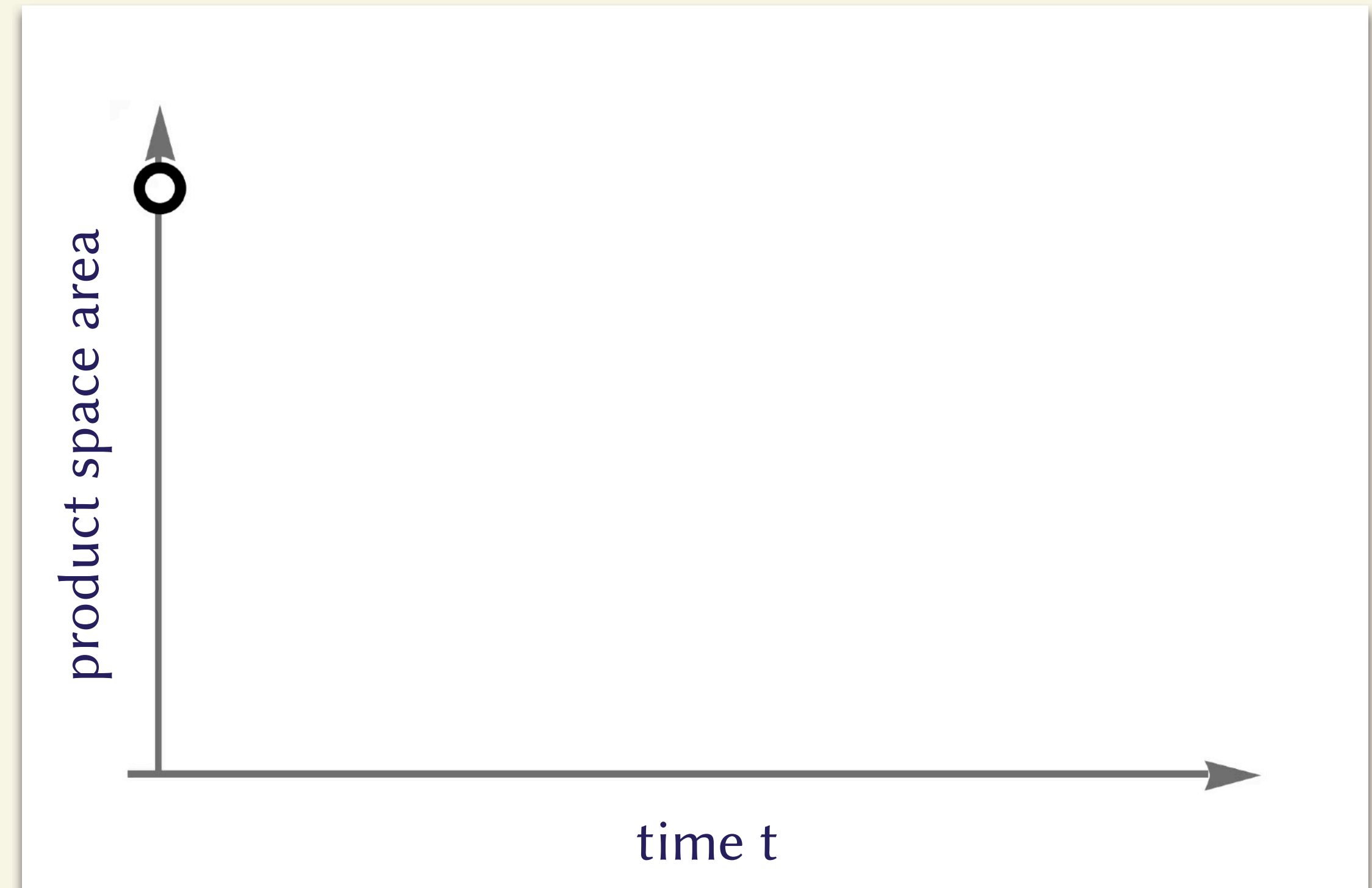
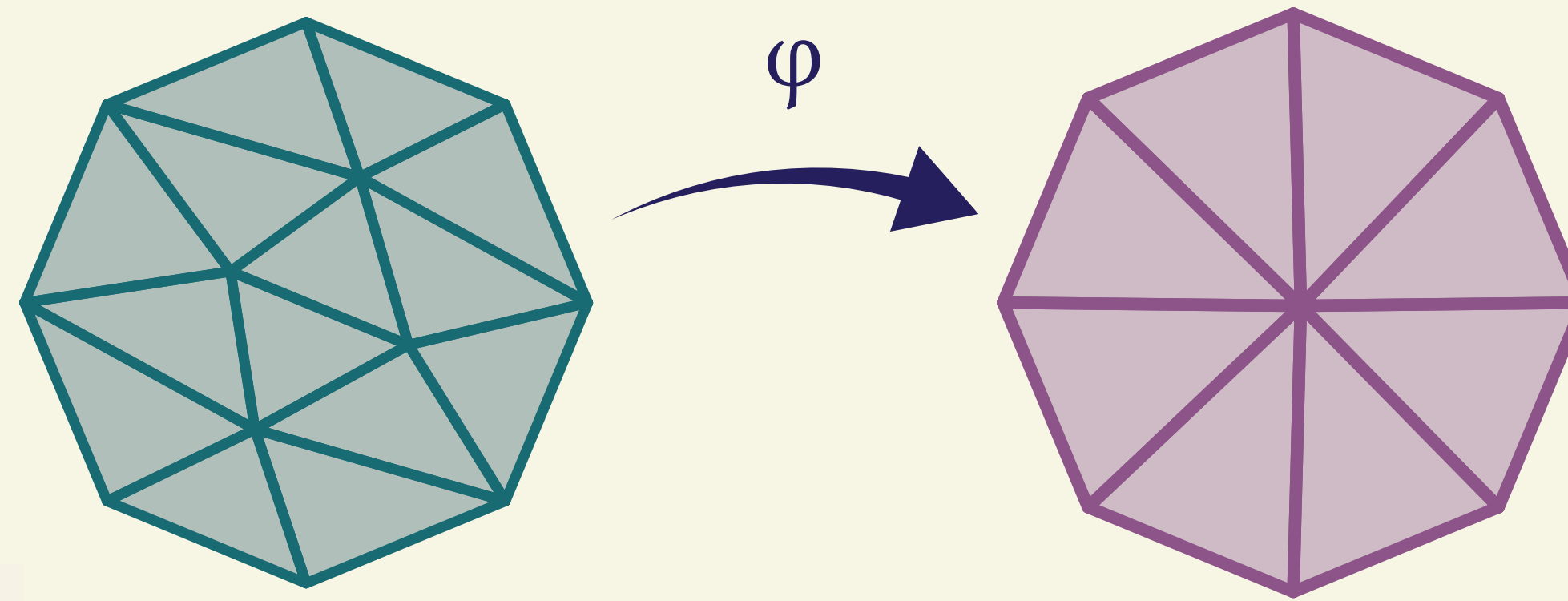
Example: 1D Curves



Example: 2D Disks



Example: 2D Disks



Find a good map by minimizing area in the product space

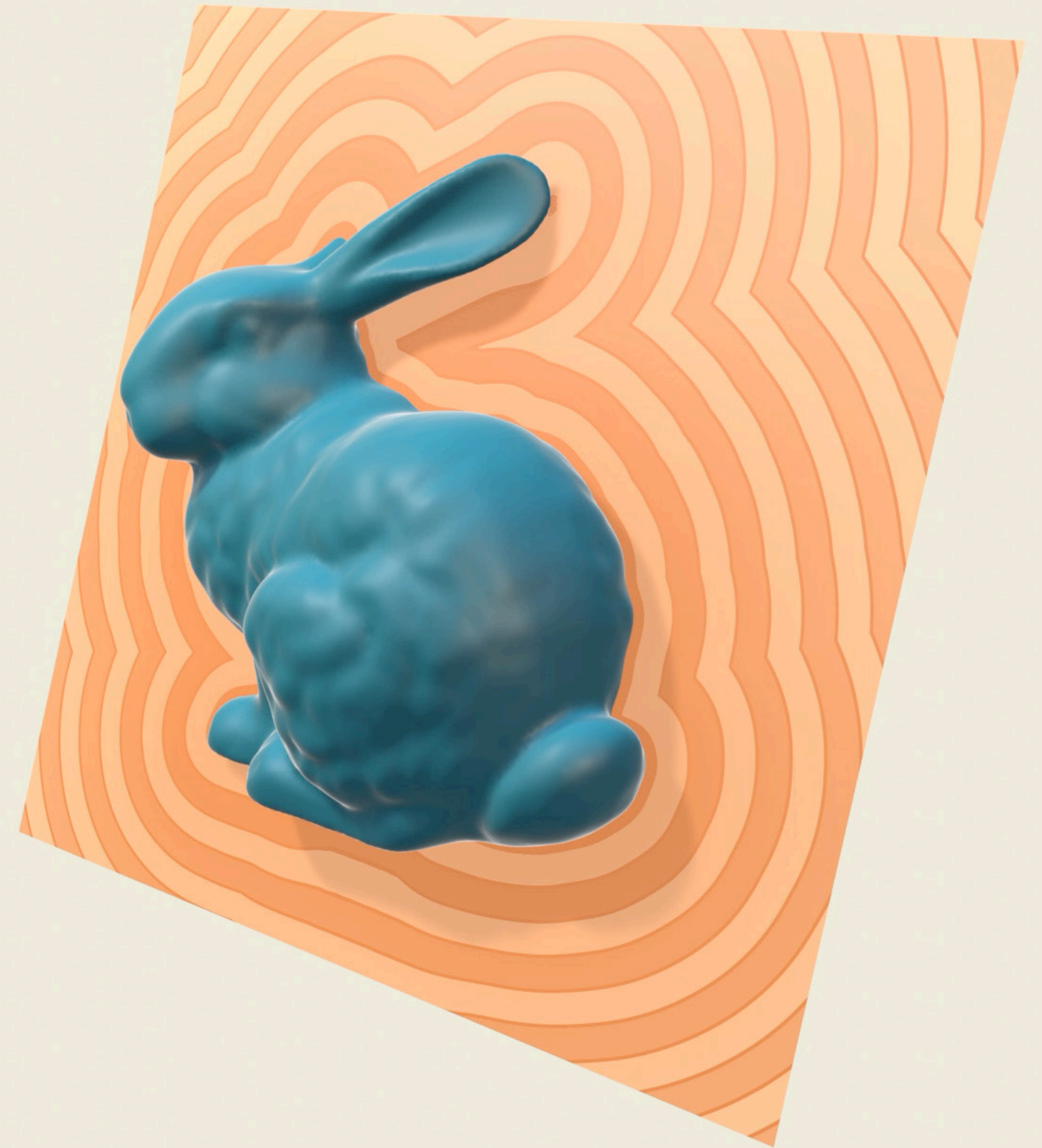
Implicit Encoding

Easy way of encoding a 2D surface in 3D space

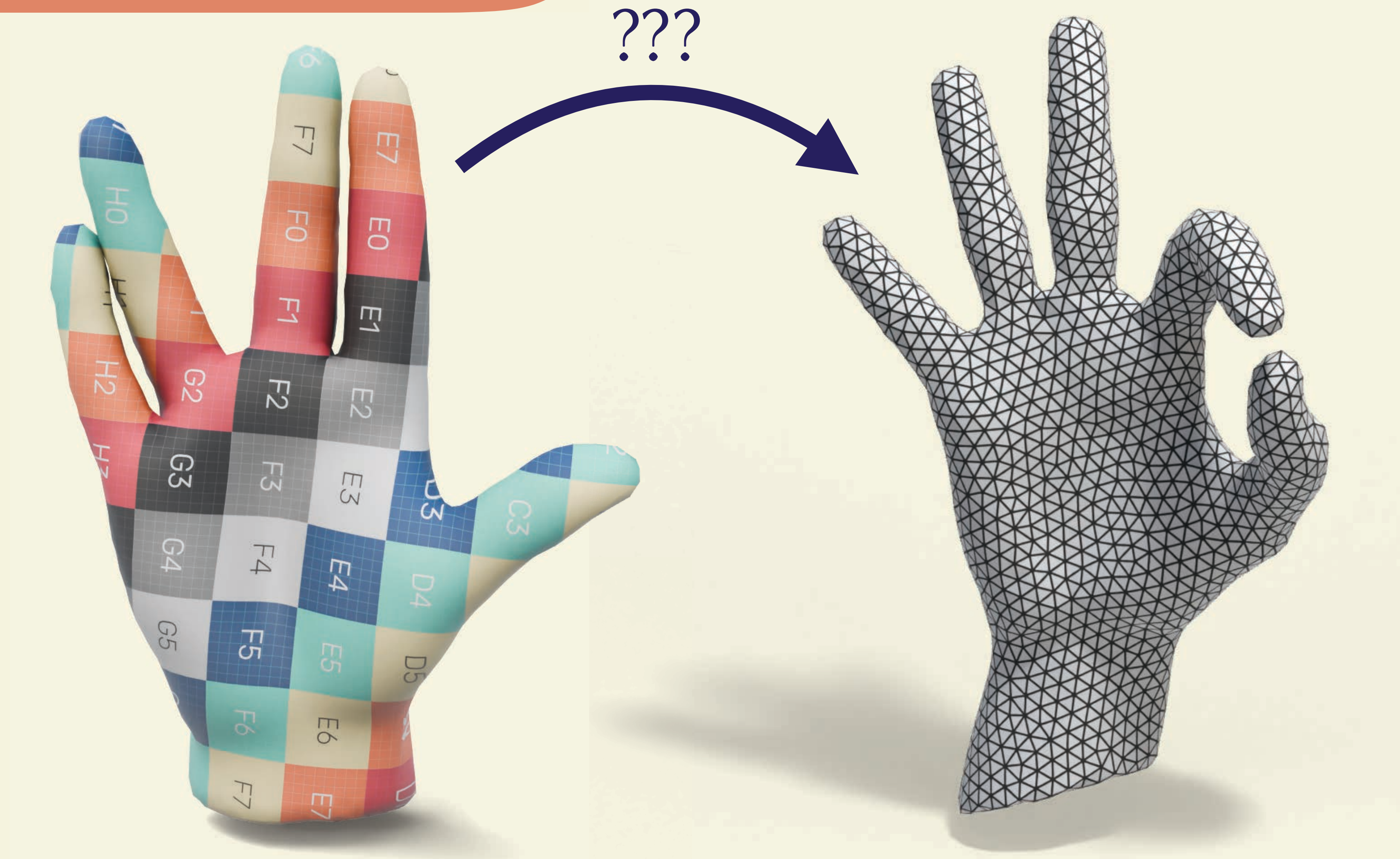
How about a 2D surface in a 4D space?

⇒ encode as the zero level set
of a *complex* function

$$f(x) = 0 \text{ on the surface}$$



Algorithm Sketch



Algorithm Sketch

I. CONSTRUCT
COMPLEX LINE BUNDLES



II. FIND AN
INITIAL MAP



III. FIT IMPLICIT
MAP

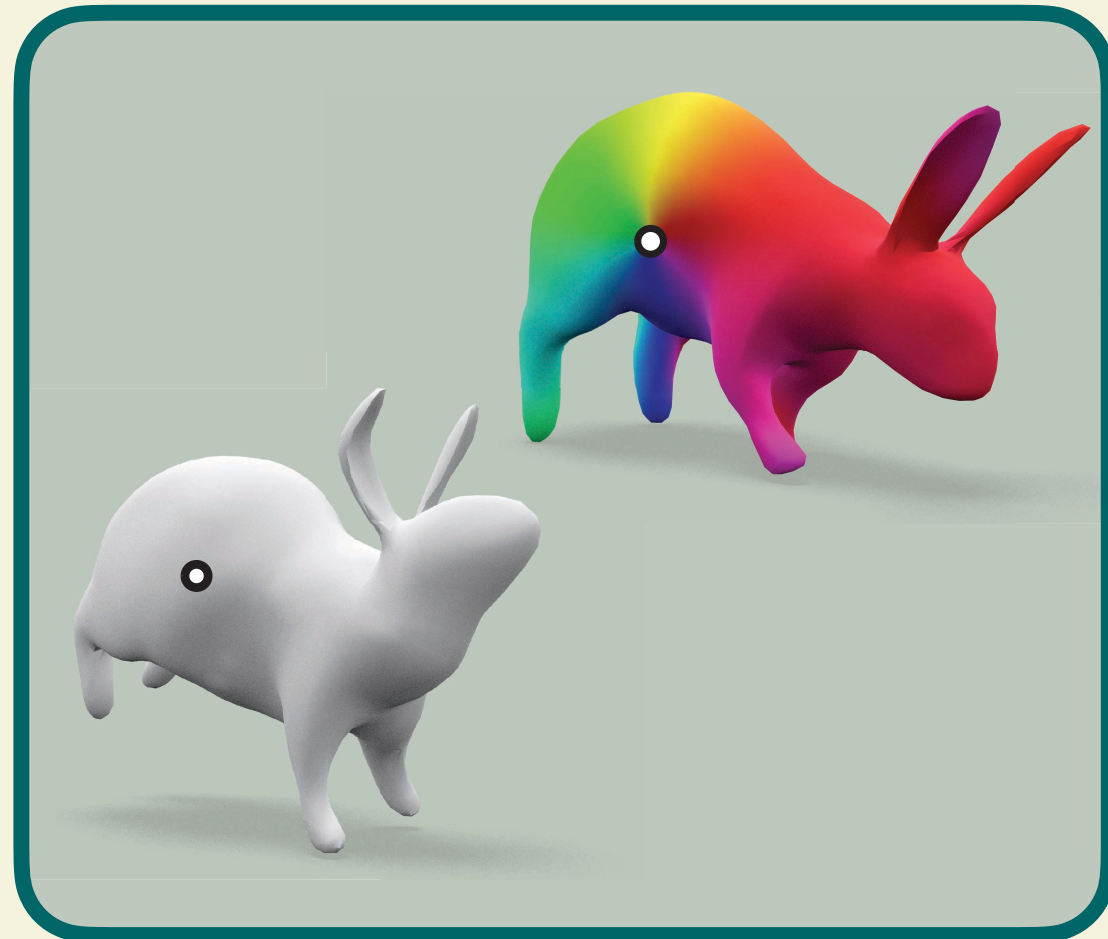


IV. OPTIMIZE
IMPLICIT MAP



Talk Outline

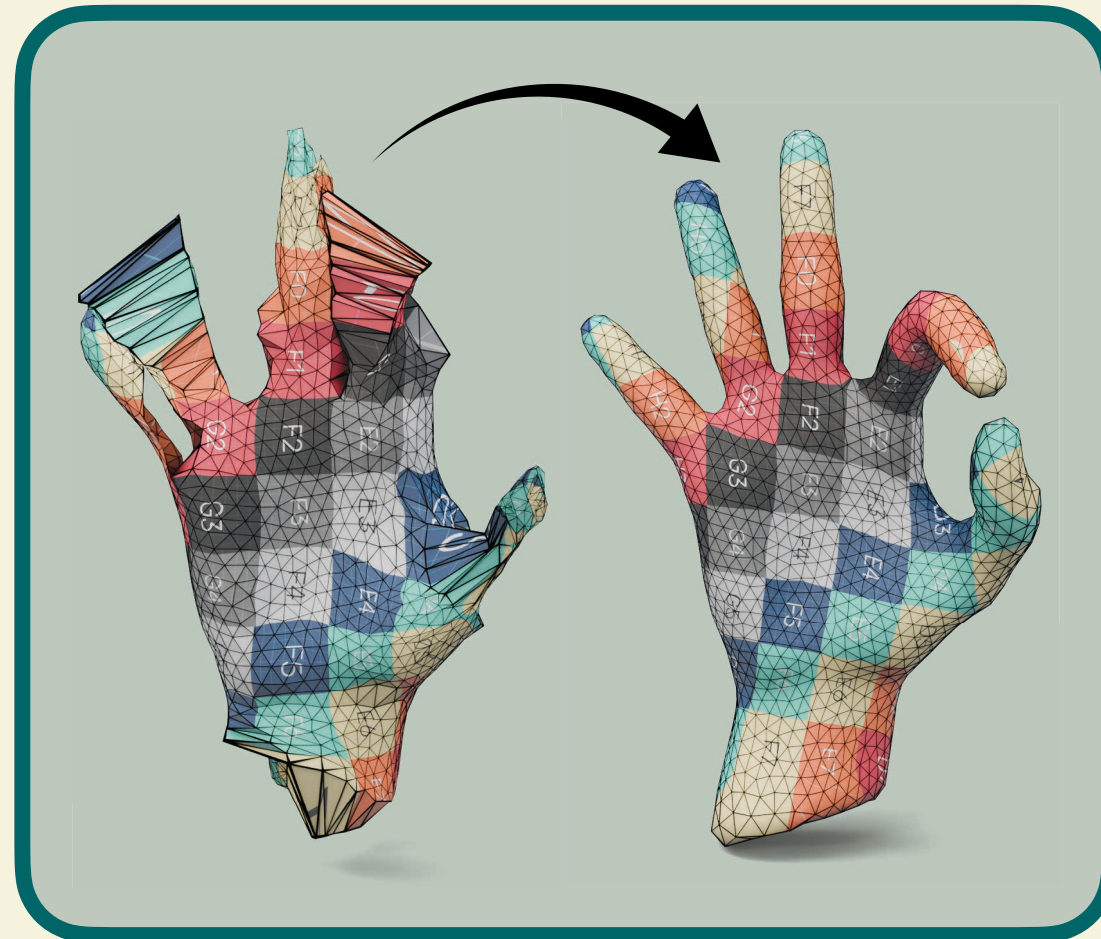
I. IMPLICIT ENCODING



encode mapping as a
complex function

(more formally: a section of
a complex line bundle)

II. MAP OPTIMIZATION



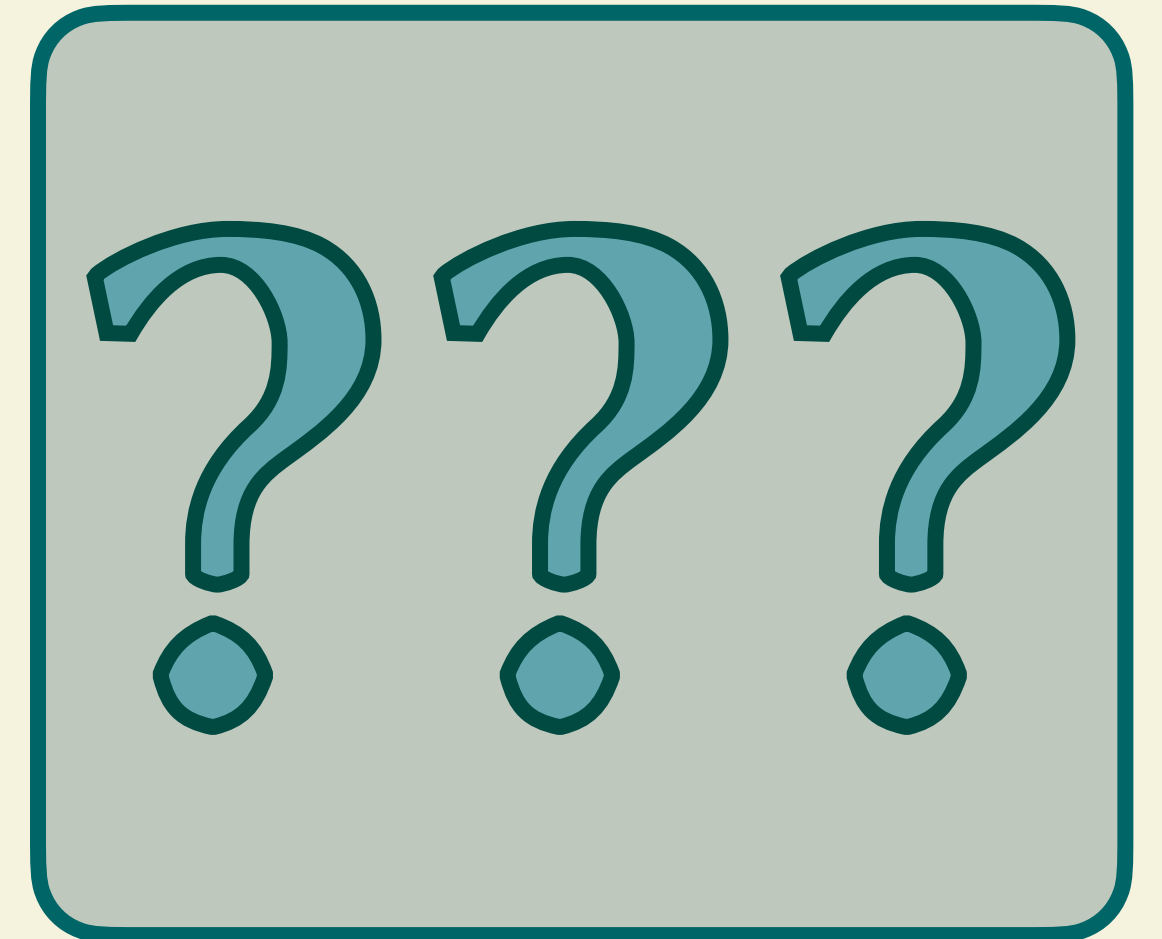
optimize complex function
to minimize distortion

III. EXAMPLES



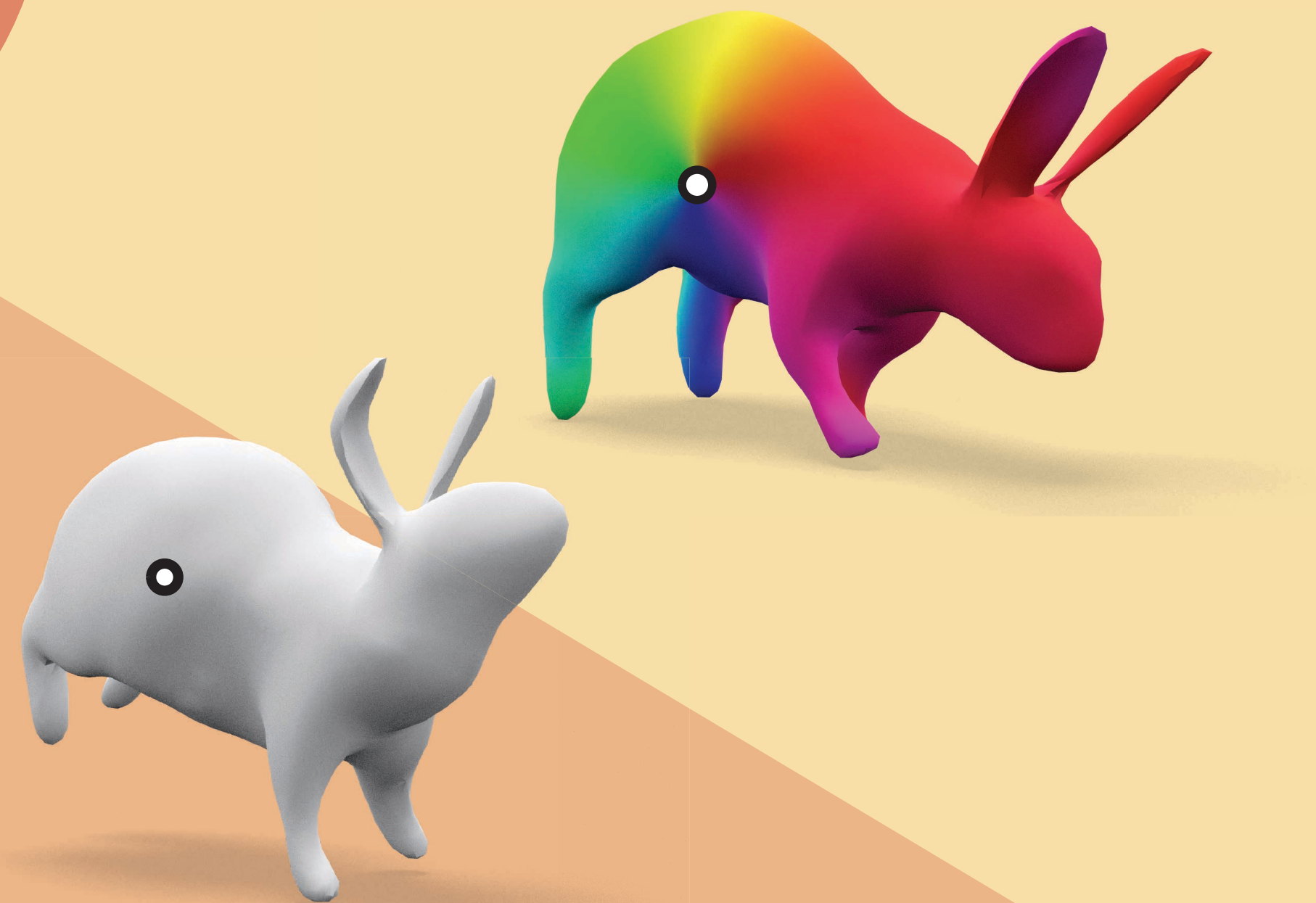
robustness & quality
advantages

IV. FUTURE WORK



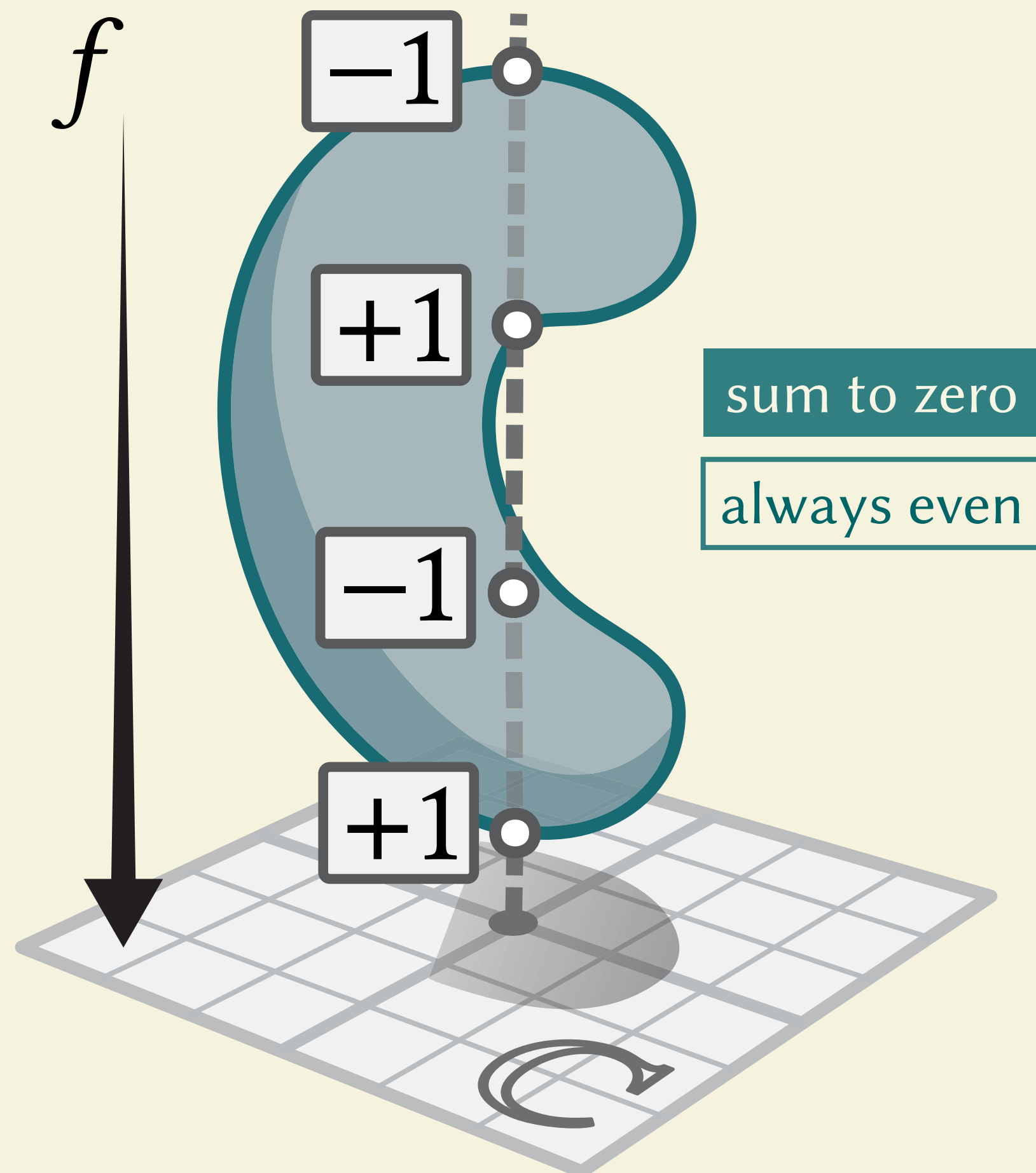
1

Implicit Encoding of Surfaces

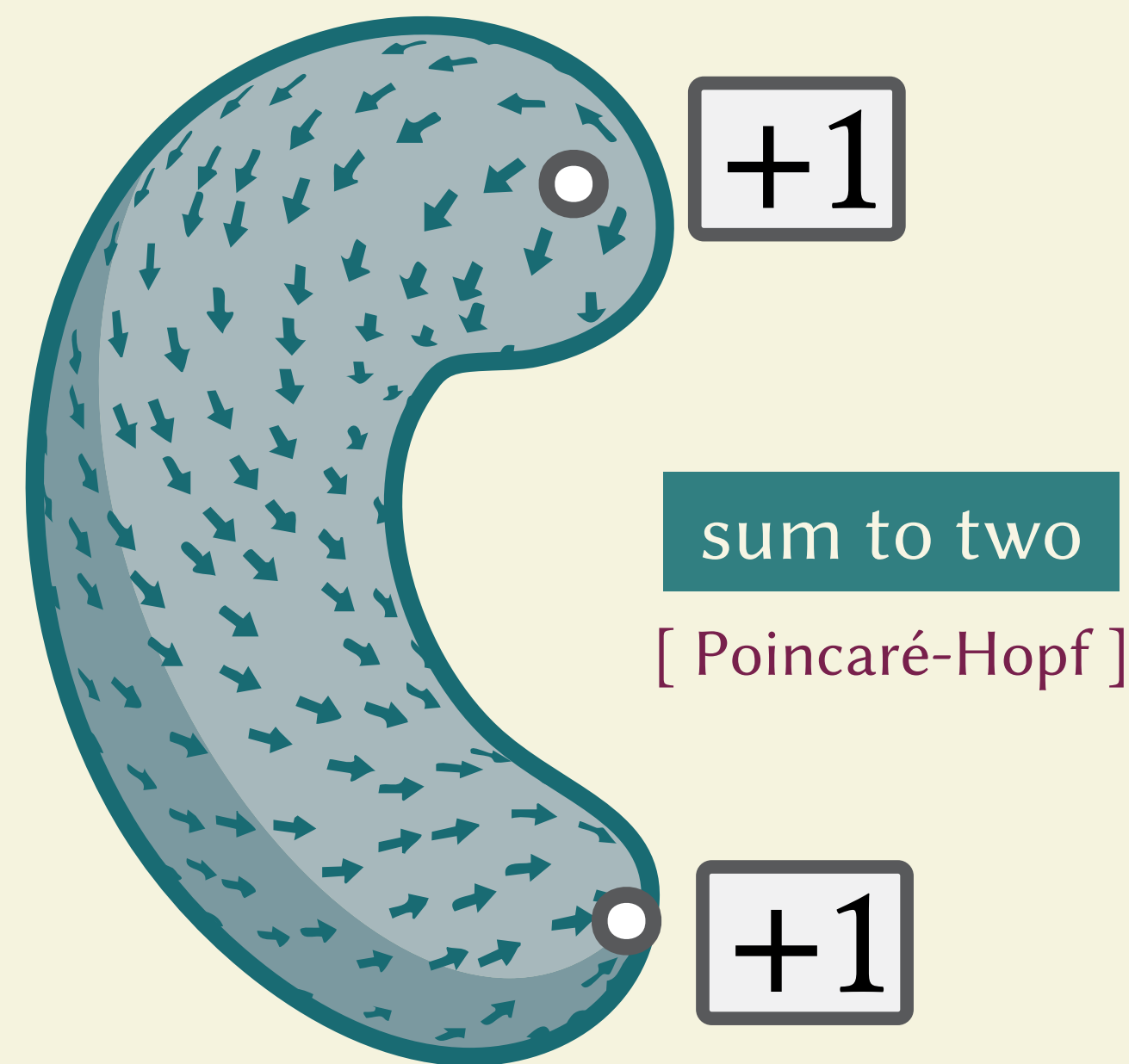


Prelude: Implicit Encoding of Points

Option 1: complex functions



Option 2: vector fields



Choice of representation
constrains which point
sets we can represent

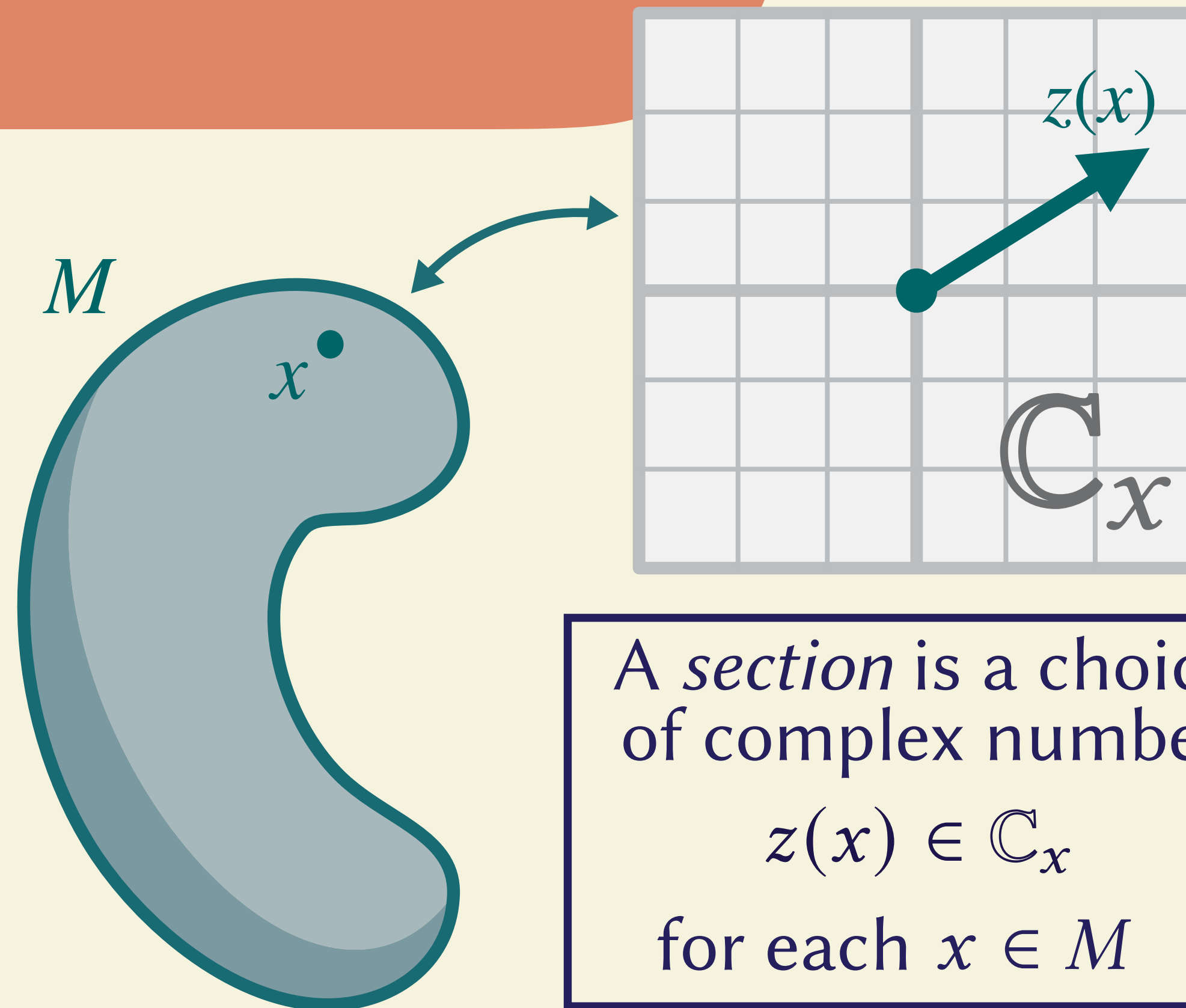
space of
complex functions space of
vector fields

examples of
complex line bundles

global structure of
complex line bundle

$\iff$
constraints on
point set

Complex line bundles



A *section* is a choice
of complex number

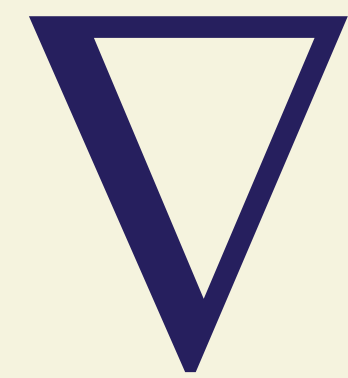
$$z(x) \in \mathbb{C}_x$$

for each $x \in M$

generalized
complex function /
vector field

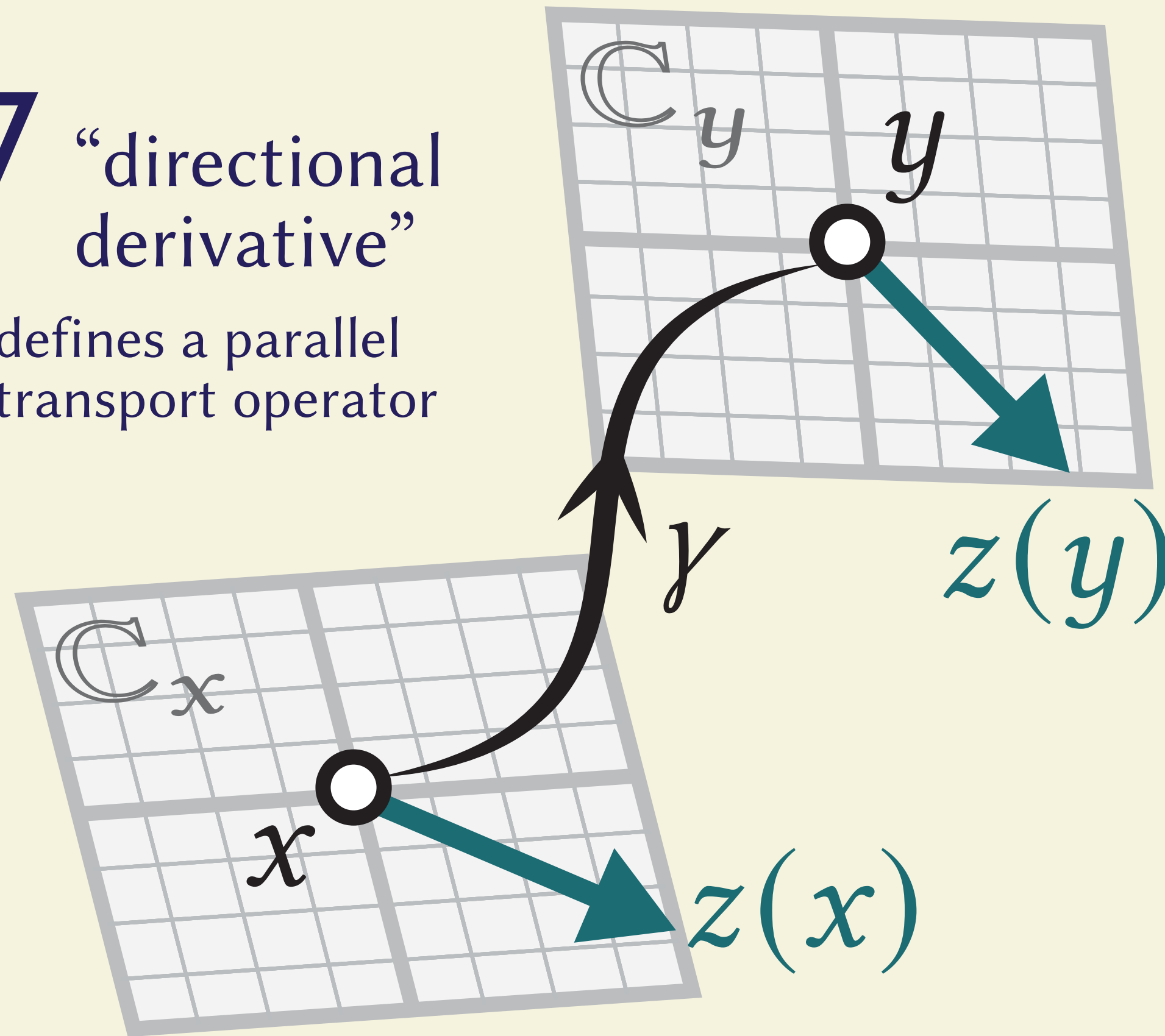
local structure: $M \times \mathbb{C}_x$
global structure may differ

Connections and Curvature



“directional derivative”

defines a parallel transport operator



“curvature 2-form”

failure to close up
around small loops

encodes the
topological class of the
complex line bundle

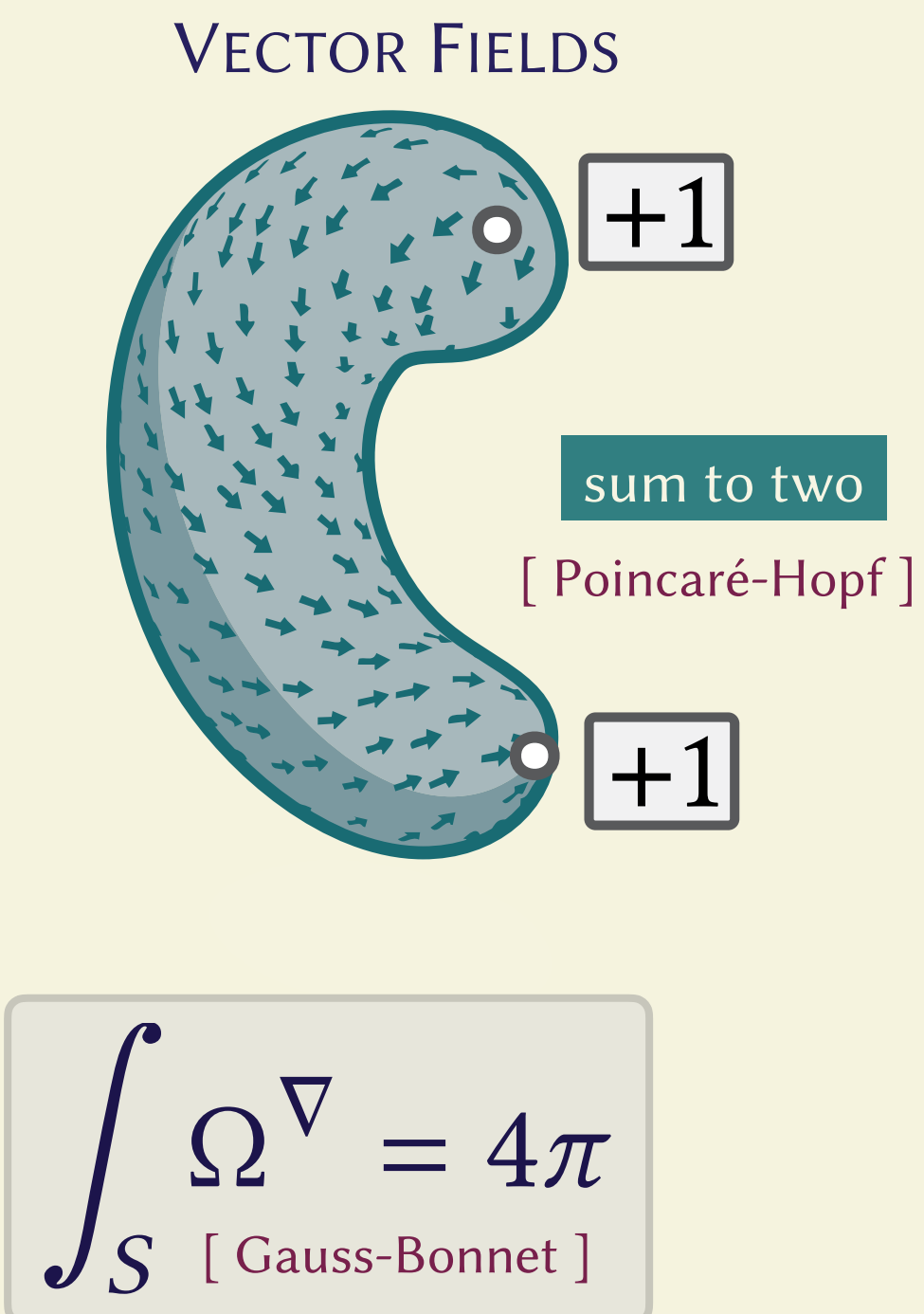
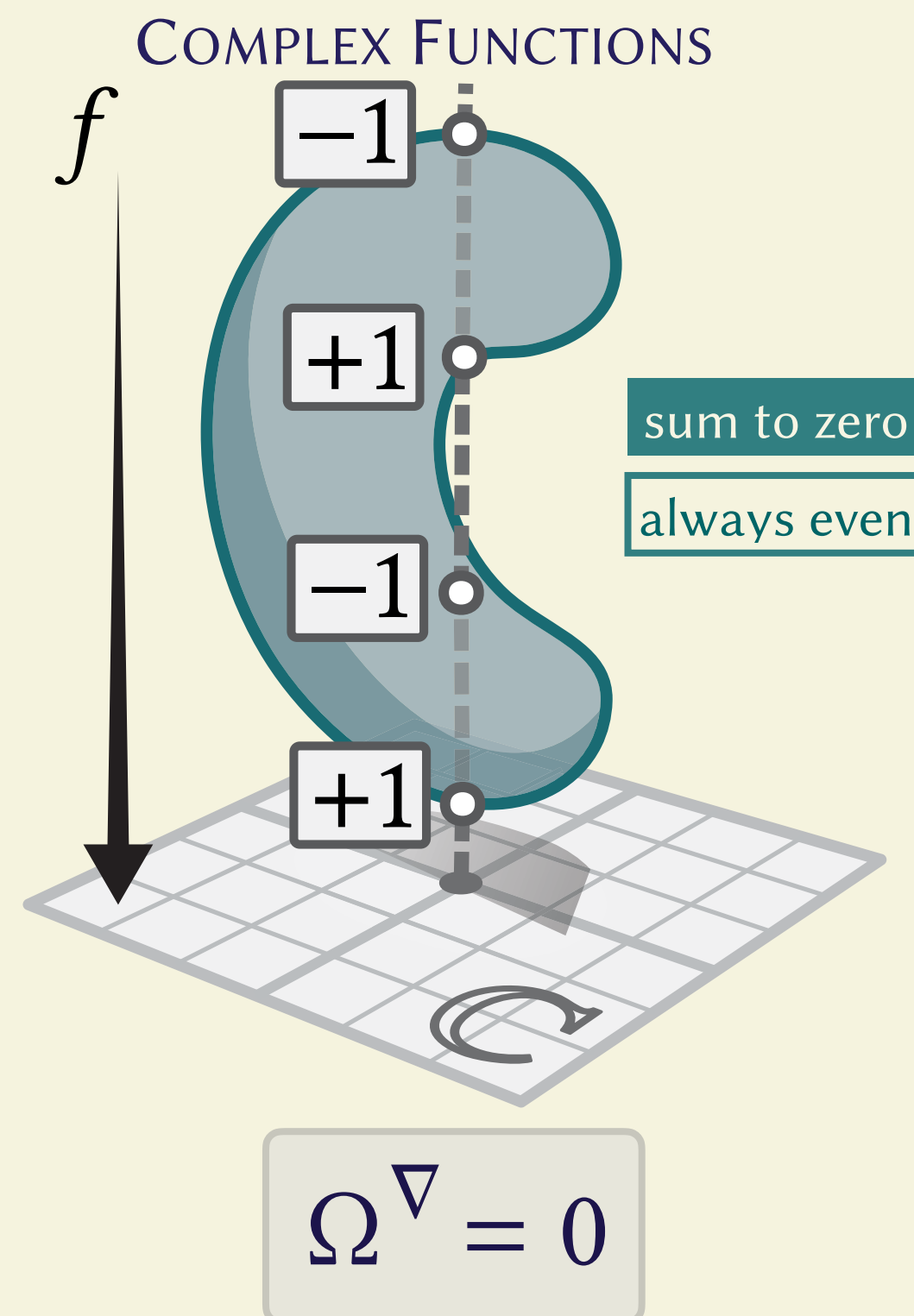
The Curvature 2-Form

Ω^∇ “curvature 2-form”
failure to close up
around small loops

encodes the
topological class of the
complex line bundle

for any complex line bundle on a surface S

$$\text{sum of zeros of a section} = \frac{1}{2\pi} \int_S \Omega^\nabla$$



on a d -dimensional domain M

consider any 2-dimensional subset S

$$\frac{1}{2\pi} \int_S \Omega^\nabla = \# \text{ oriented intersections between } S \text{ and the zero set of a section}$$

Ω^∇ determines the
topology of the zero
set of *any* section

Discrete Complex Line Bundles

[Knöppel & Pinkall 2016]

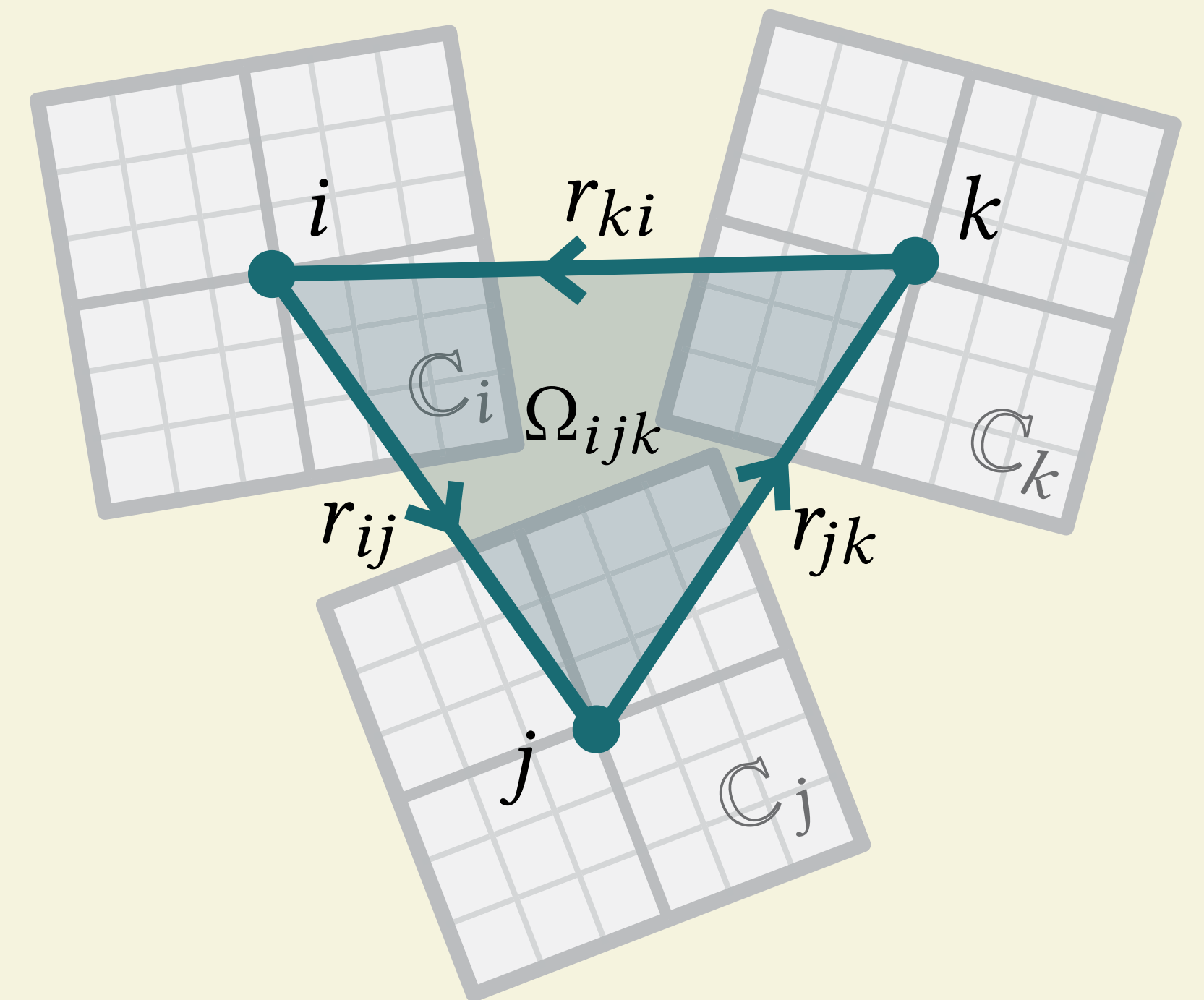
on a d -dimensional mesh $M = (V, E, F, \dots)$

1. A copy $\mathbb{C}_i$ of the complex plane for each vertex $i \in V$
2. A parallel transport map $r_{ij} \in \mathbb{C}$ for each edge $ij \in E$
3. A curvature $\Omega_{ijk} \in \mathbb{R}$ for each face $ijk \in F$

$$r_{ij} r_{jk} r_{ki} = \exp(\imath \Omega_{ijk})$$

(Ω_{ijk} measures rotation around face ijk)

Ω determines the topology of the zero set of any discrete section

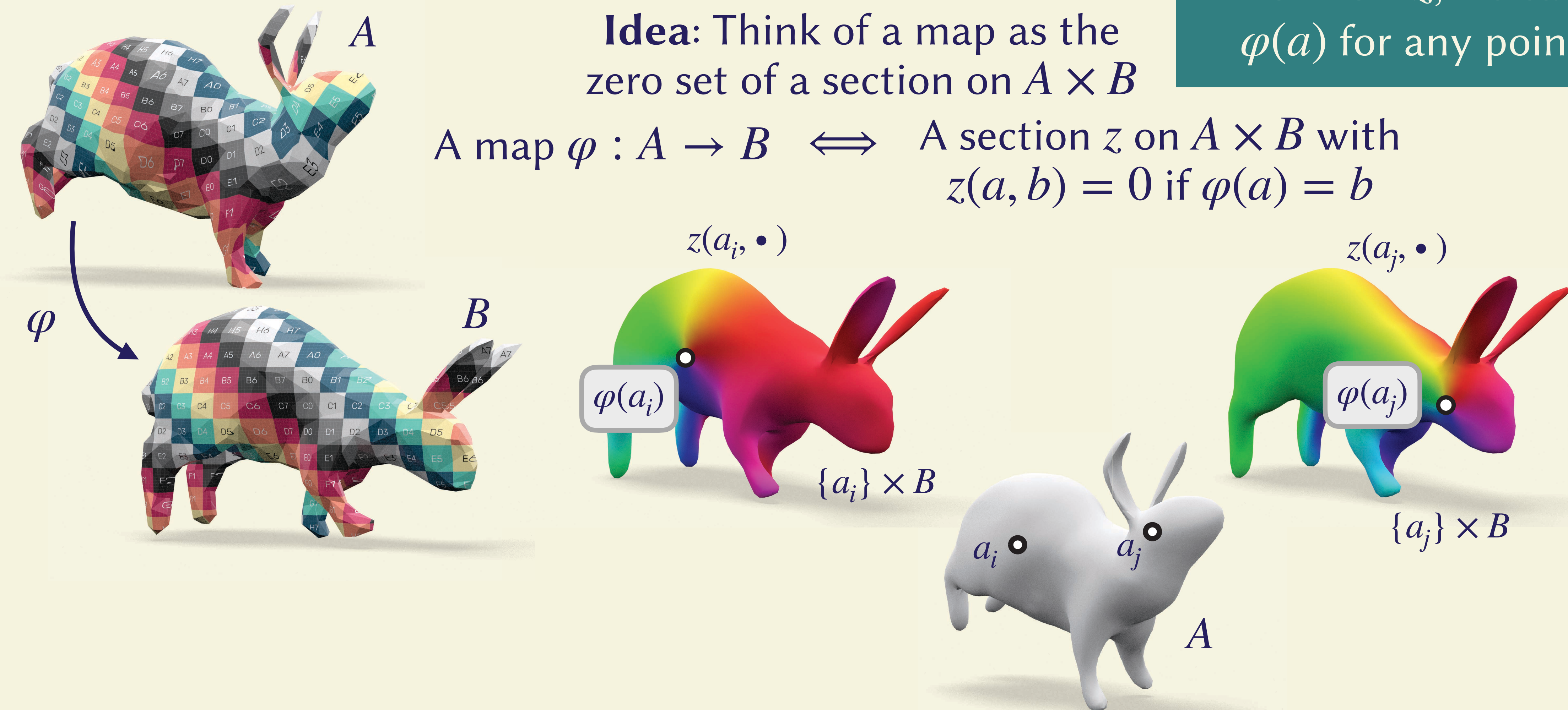


Representing a Map as a Complex Section

If we know z , we can evaluate $\varphi(a)$ for any point $a \in A$

Idea: Think of a map as the zero set of a section on $A \times B$

A map $\varphi : A \rightarrow B \iff$ A section z on $A \times B$ with $z(a, b) = 0$ if $\varphi(a) = b$



Extra Information: Edge Crossings

Does not require any combinatorial data or mesh modification — *only* uses complex values z on $A \times B$

Faces in $A \times B$

$v_A \times f_B$
does v_A
land in f_B ?

$f_A \times v_B$
does v_B
land in f_A ?

$e_A \times e_B$
does e_A
intersect e_B ?

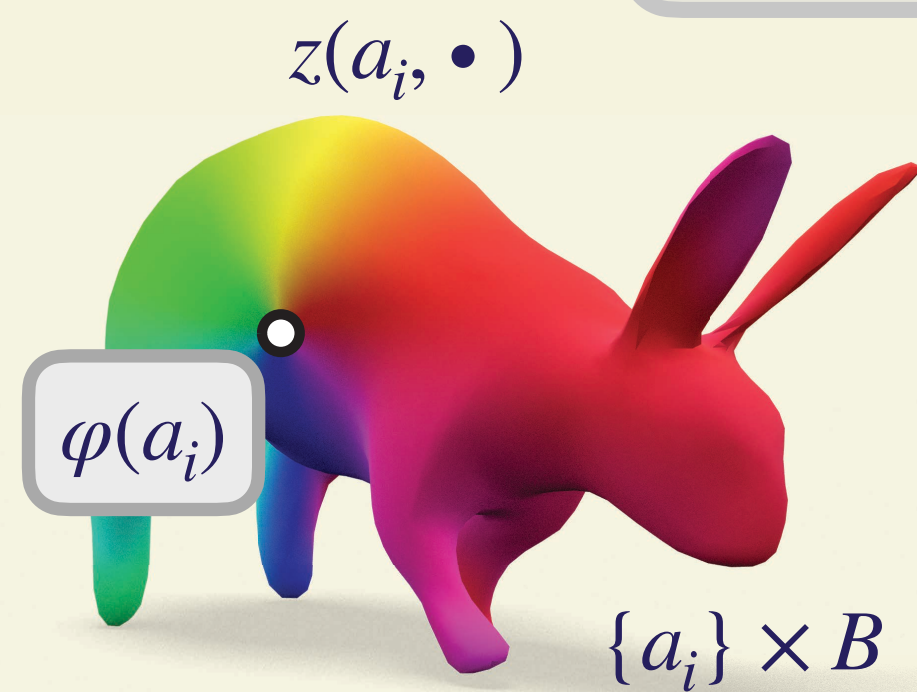
φ

We can read off the *overlay mesh* directly from the section z

How do we build the complex line bundle?

Need to pick:

- A parallel transport map $r_{ij} \in \mathbb{C}$ for each edge $ij \in E$
- A curvature $\Omega_{ijk} \in \mathbb{R}$ for each face $ijk \in F$



want to one zero per slice

oriented intersections between S and the zero set of a section

$$= \frac{1}{2\pi} \int_S \Omega^\nabla$$

pick a connection with total curvature 2π on each slice

Easy solution:

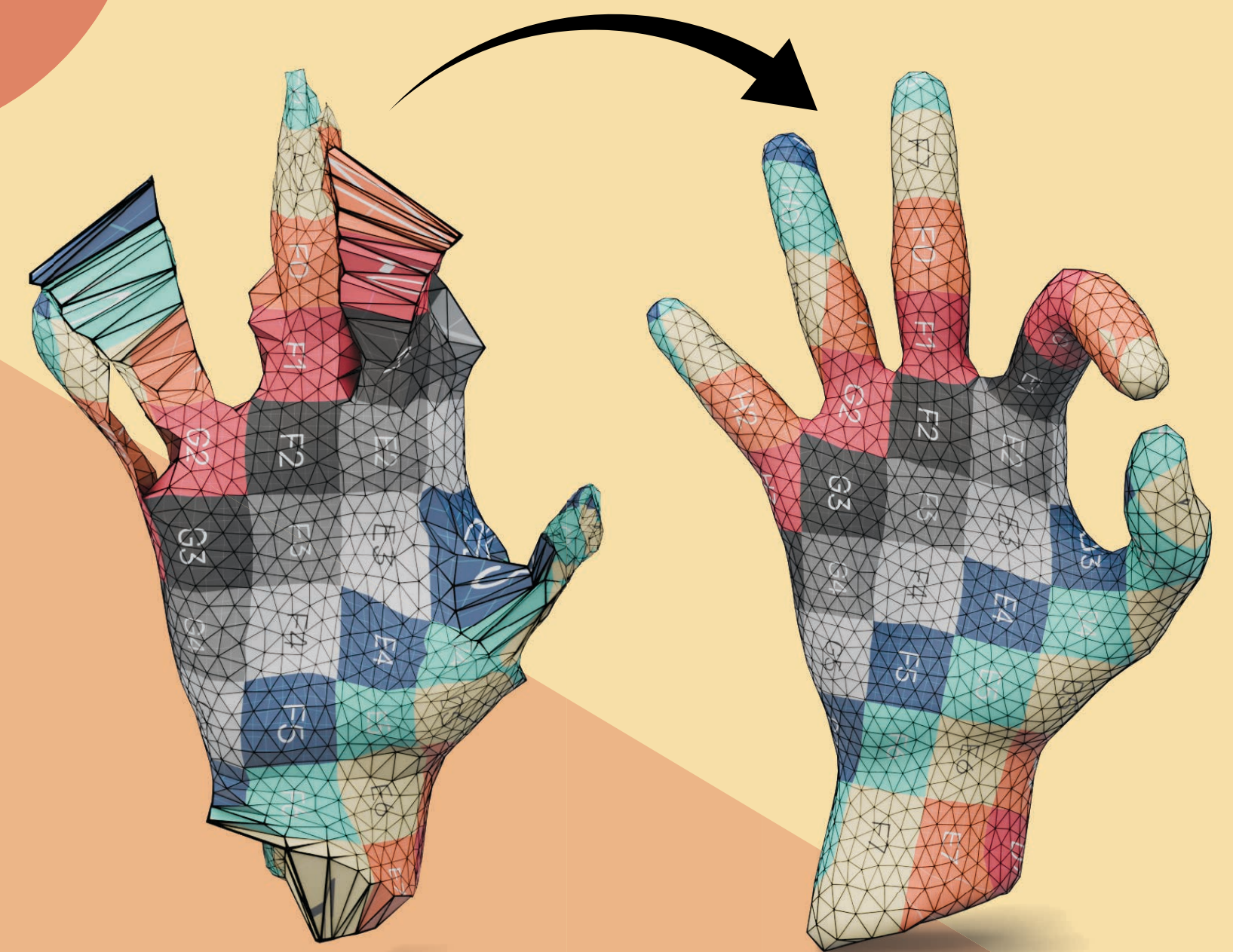
- Build a connection on A with curvature 2π
- Build a connection on B with curvature 2π
- Take *product connection* on $A \times B$

(solve via *trivial connections*)

[Crane et al. 2010]

guarantees that map will lie in correct topological class

2 Map Optimization



The Ginzburg-Landau Energy

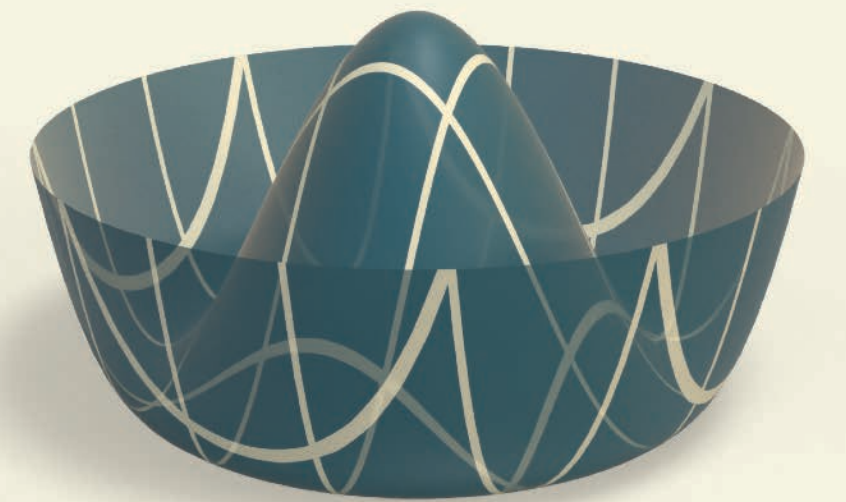
$$\mathcal{E}_\varepsilon^{\mathcal{GL}}(z) := \int_A \int_B \frac{1}{2} |\nabla z(p)|^2 + \frac{1}{4\varepsilon^2} \left(|z(p)|^2 - 1 \right)^2 dA dB$$

“Dirichlet energy”

encourages z to change as little as possible

“Circular well potential”

encourages z to have unit norm everywhere



At a minimizer:

gets very large
near zeros

z has unit norm,
except near zeros

make zero set as small as possible

minimal surfaces as minimizers [Lin and Riviere 1999; Canevari et al. 2023]

Minimizing the Ginzburg-Landau Energy

discrete section: $Z \in \mathbb{C}^{V_A \times V_B}$

$$\mathcal{E}_\varepsilon^{\mathcal{GL}}(Z) = \frac{1}{2} \text{Tr} \left[\bar{Z}^\top L_A^\nabla Z (M_B^\nabla)^\top \right] + \frac{1}{2} \text{Tr} \left[\bar{Z}^\top M_A^\nabla Z (L_B^\nabla)^\top \right] + \frac{1}{4\varepsilon^2} \text{Tr} \left[U^\top M_A U M_B^\top \right]$$

some matrix expression of Z $\left(U_{i,j} = |Z_{i,j}|^2 - 1 \right)$

$L_A^\nabla, L_B^\nabla, M_A^\nabla, M_B^\nabla, M_A, M_B$ connection Laplacian, mass matrices — fixed matrices depending on A and B
[Knöppel & Pinkall 2016]

straightforward to differentiate

optimize with off-the-shelf LBFGS solver

Landmark Points

singularity pinning

CIRCULAR WELL POTENTIAL

$$\mathcal{W}(z) := \frac{1}{4} \int_A \int_B \left(|z(p)|^2 - 1 \right)^2 dA dB$$

“PINNED” CIRCULAR WELL POTENTIAL

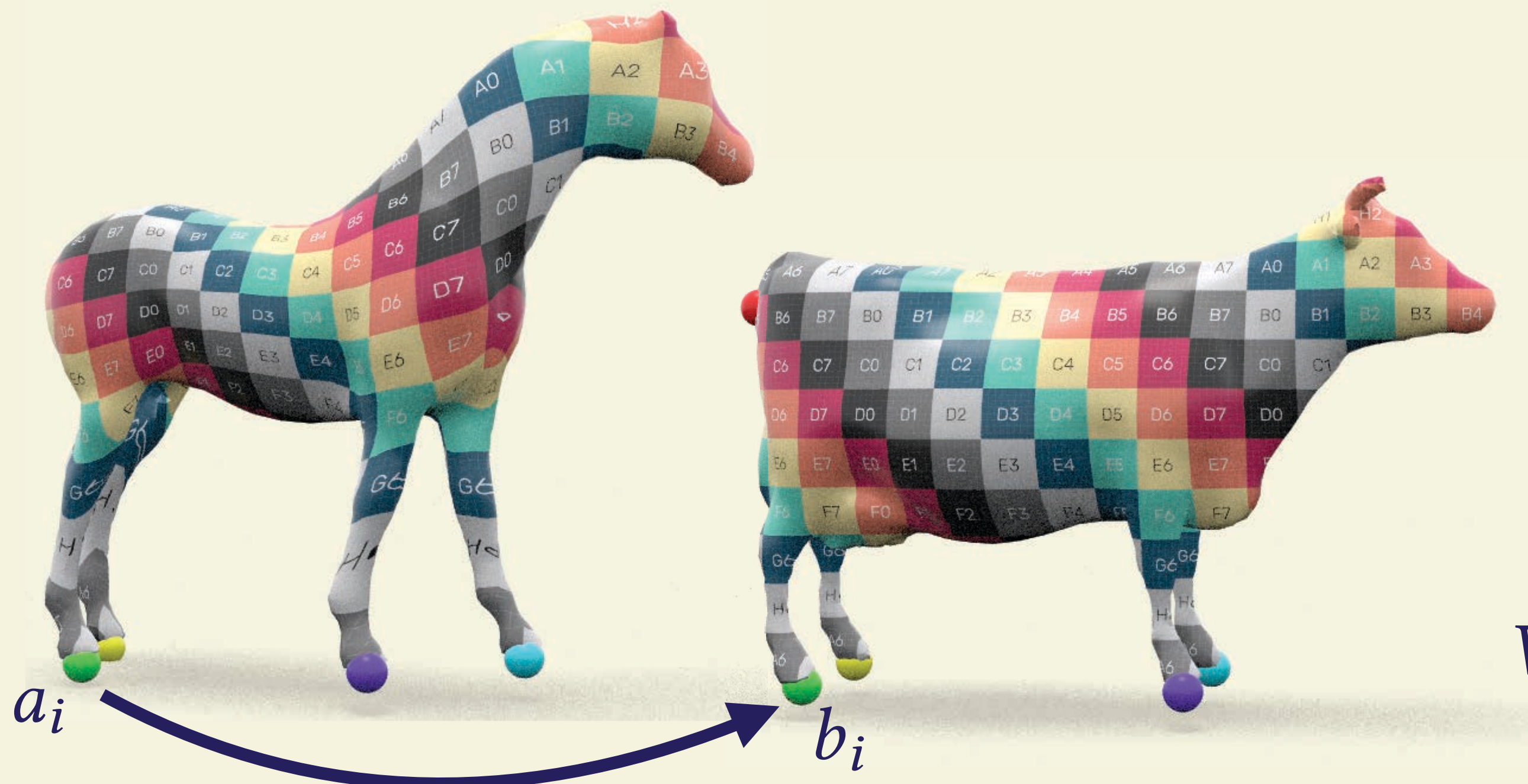
$$\mathcal{W}(z) := \frac{1}{4} \int_A \int_B \left(|z(p)|^2 - V(p) \right)^2 dA dB$$

$V(p)$ determines target magnitude

$$V(p) := 1 - \min_{\substack{\text{landmark} \\ \text{pairs } (a_i, b_i)}} \exp \left(- d_{A \times B}[p, (a_i, b_i)]^2 \right)$$

encourage zeros to form around $(a_i, b_i) \in A \times B$

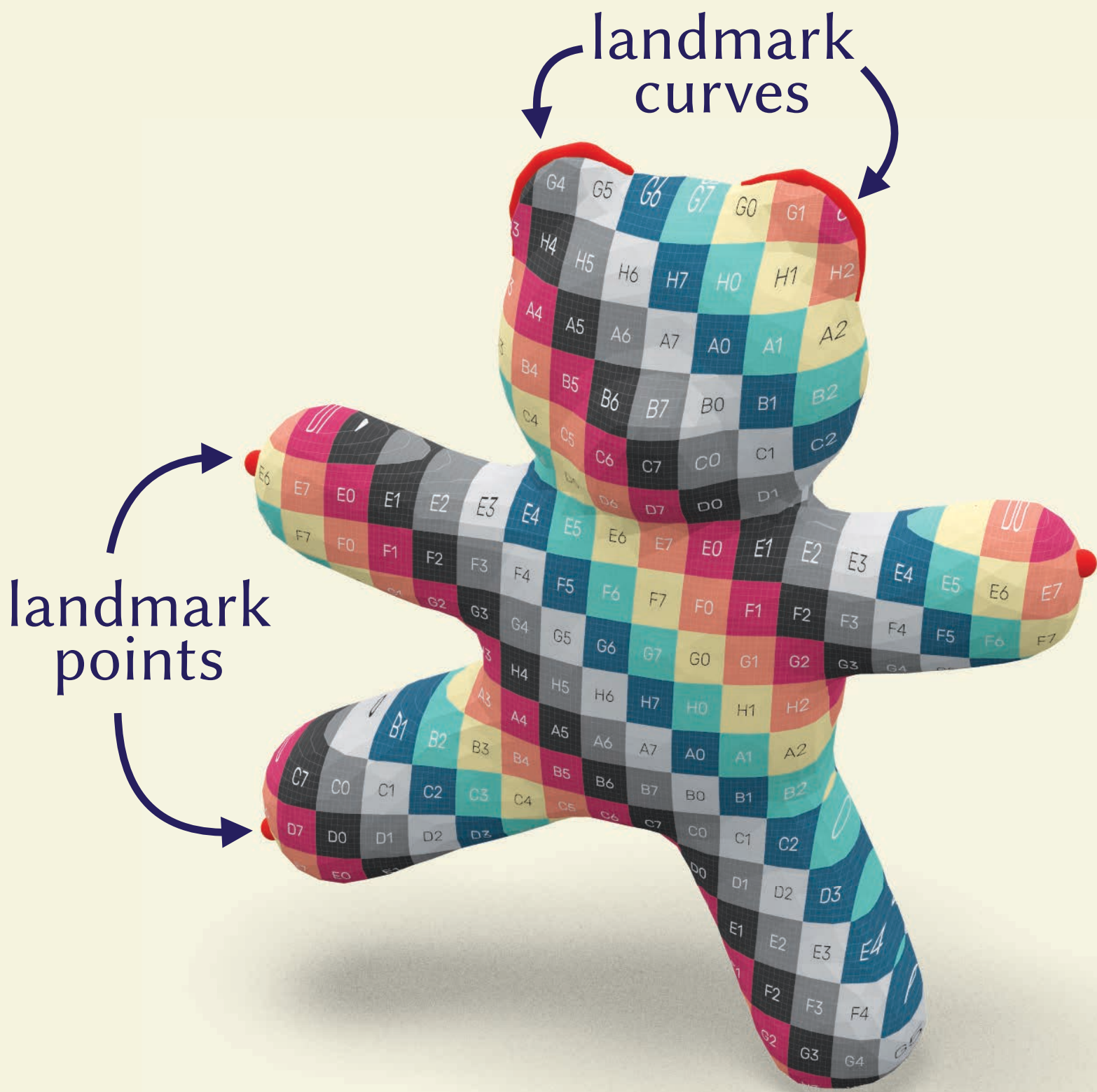
soft penalty encouraging
 a_i to map to b_i



Landmark Curves

landmark

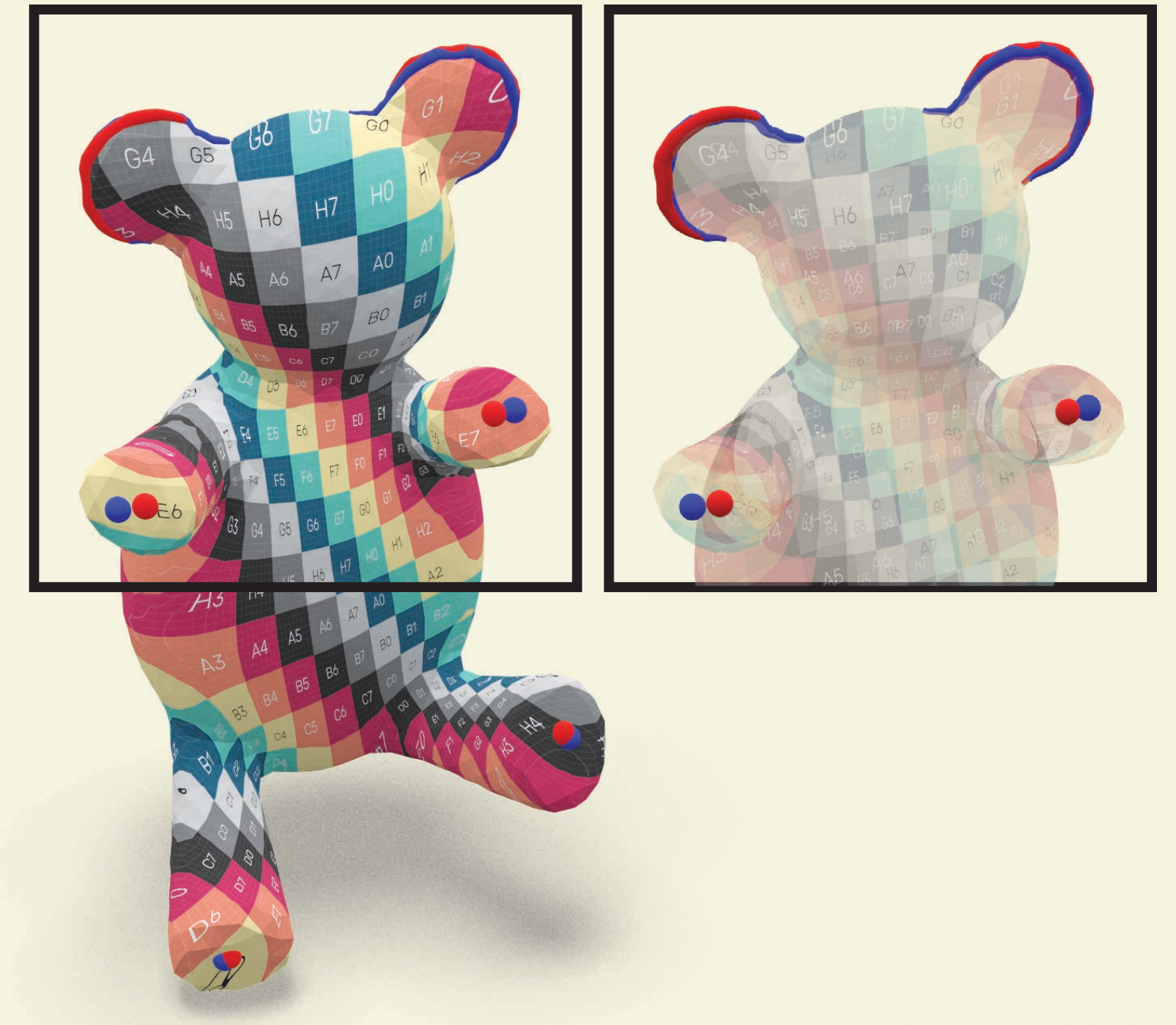
■ mapped landmark



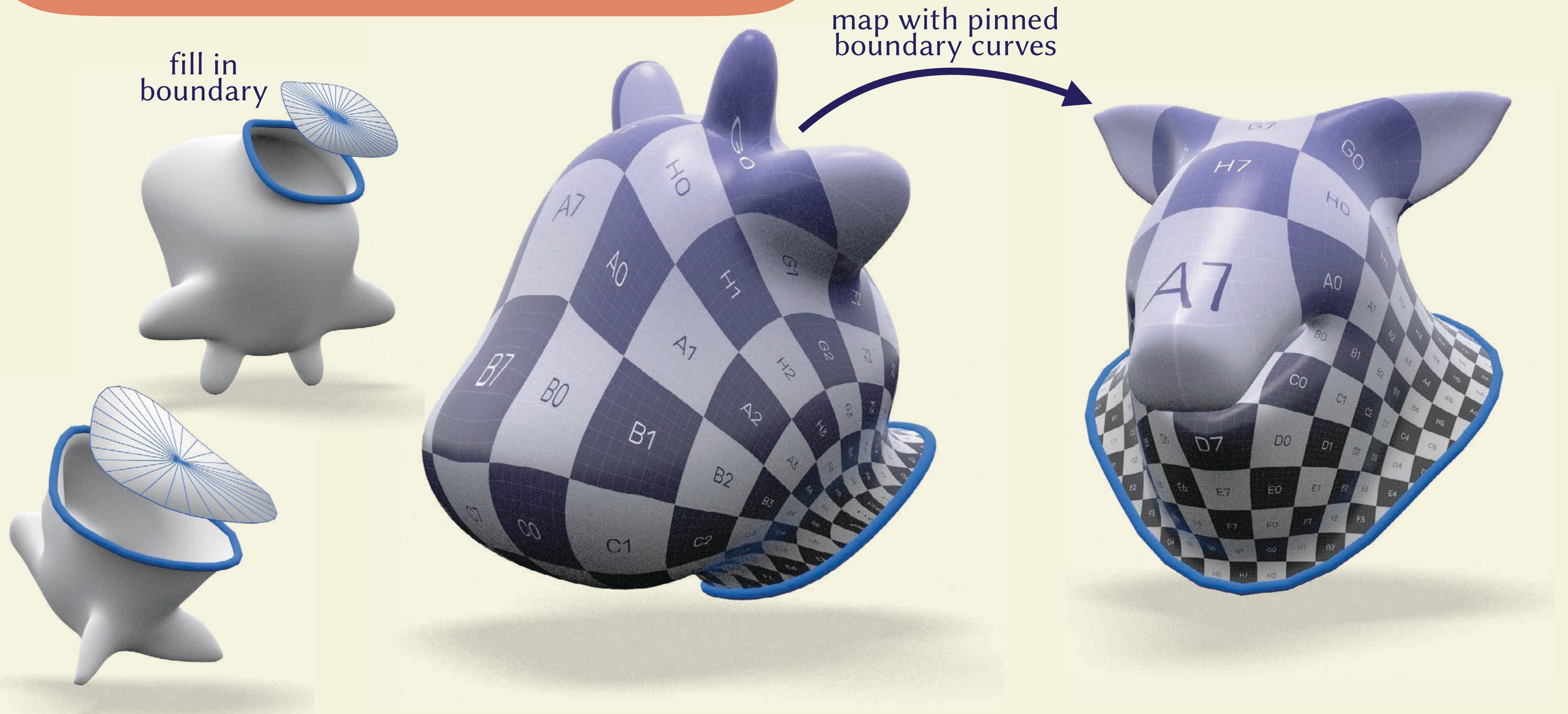
using landmark points only



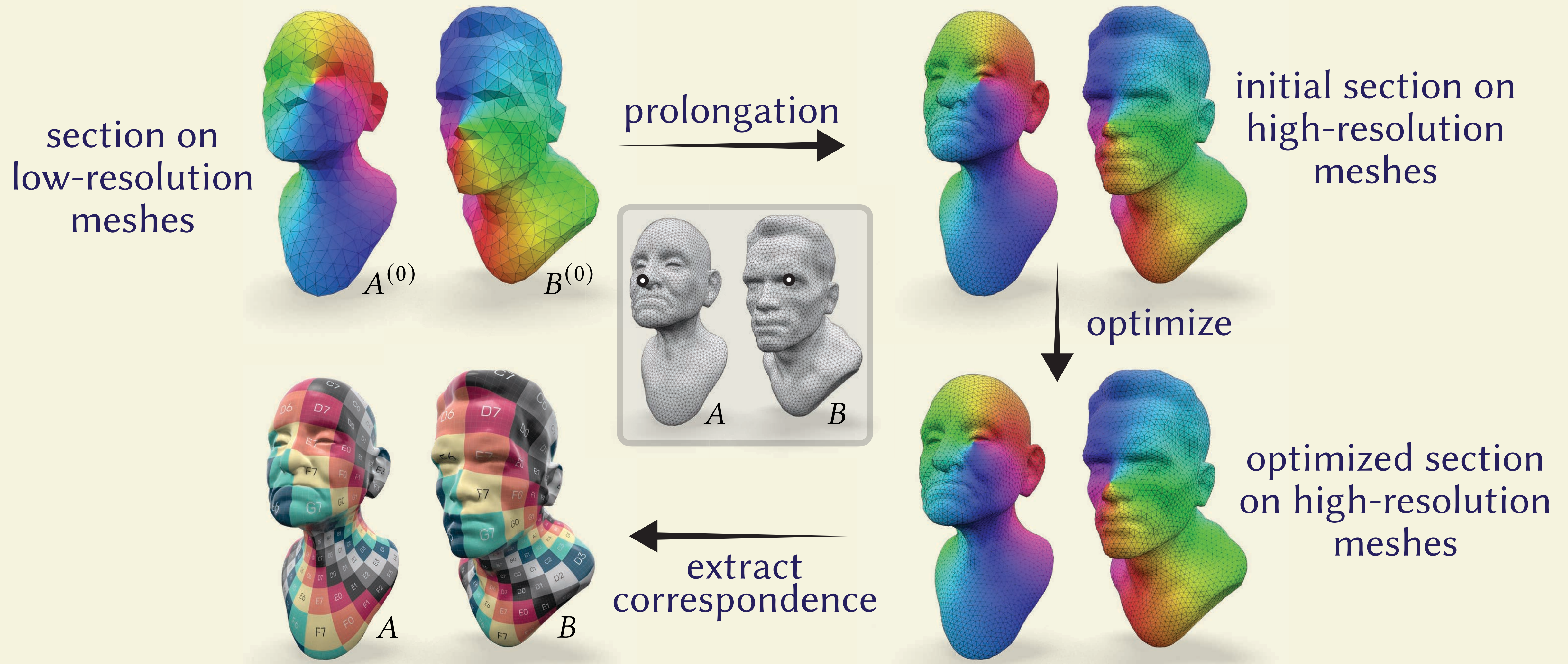
using landmark points & curves



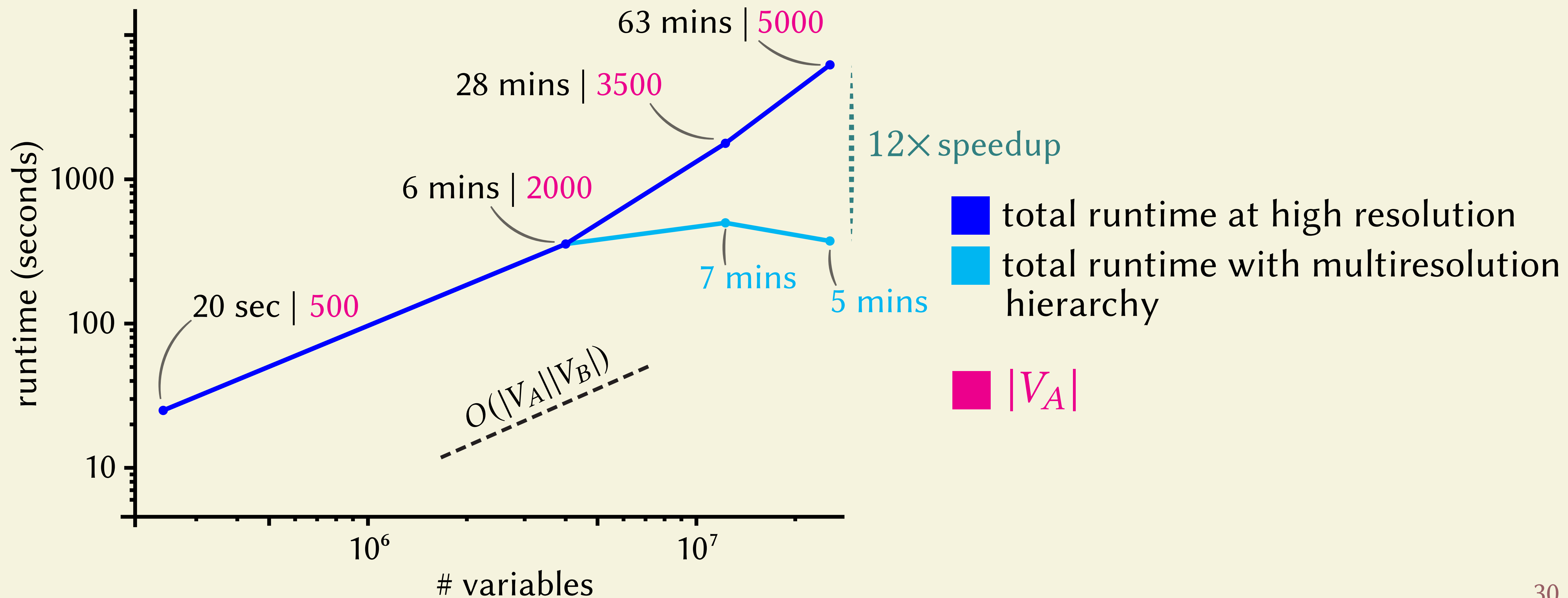
Handling Boundaries



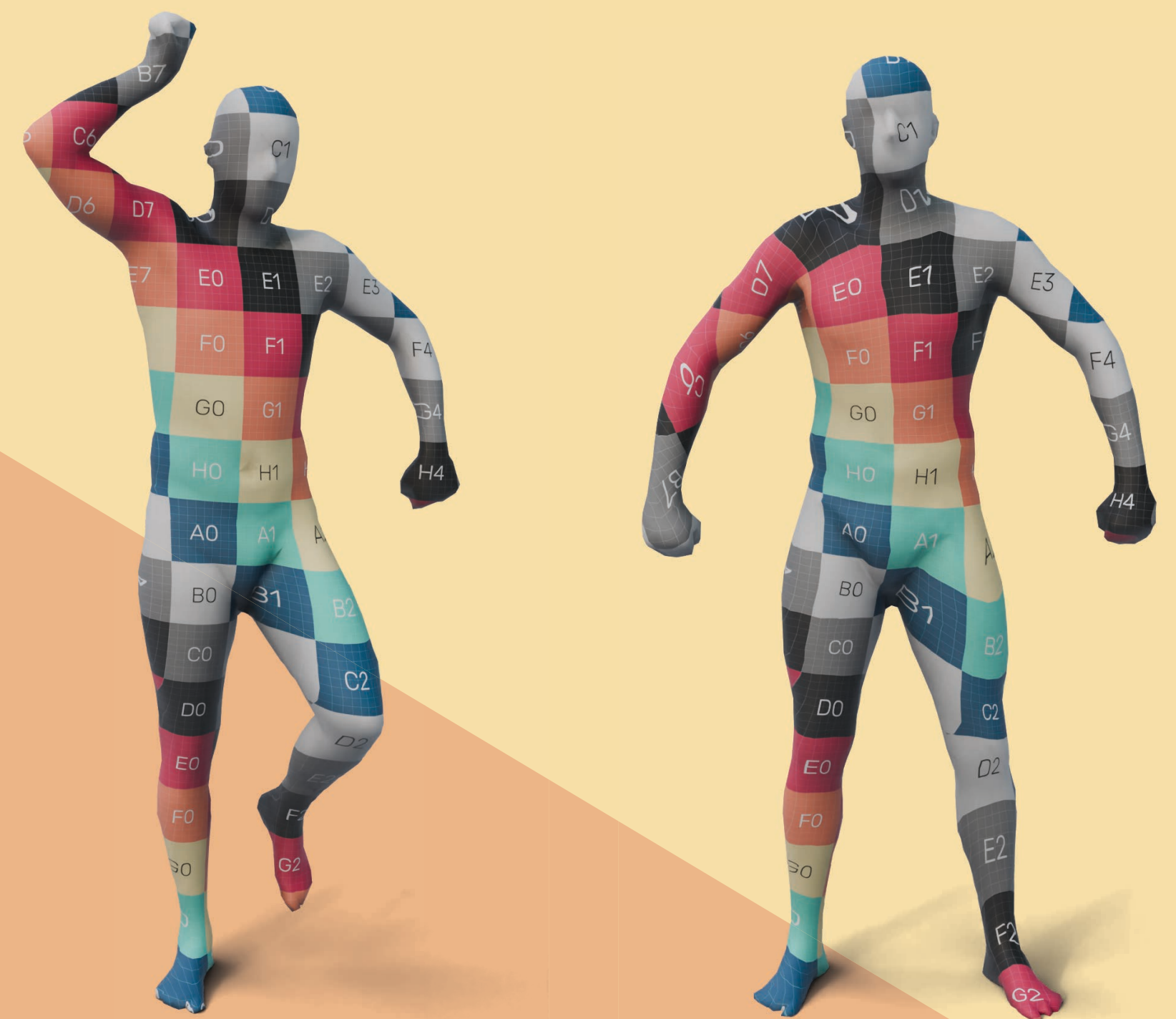
Acceleration via Multiresolution Hierarchy



Runtime



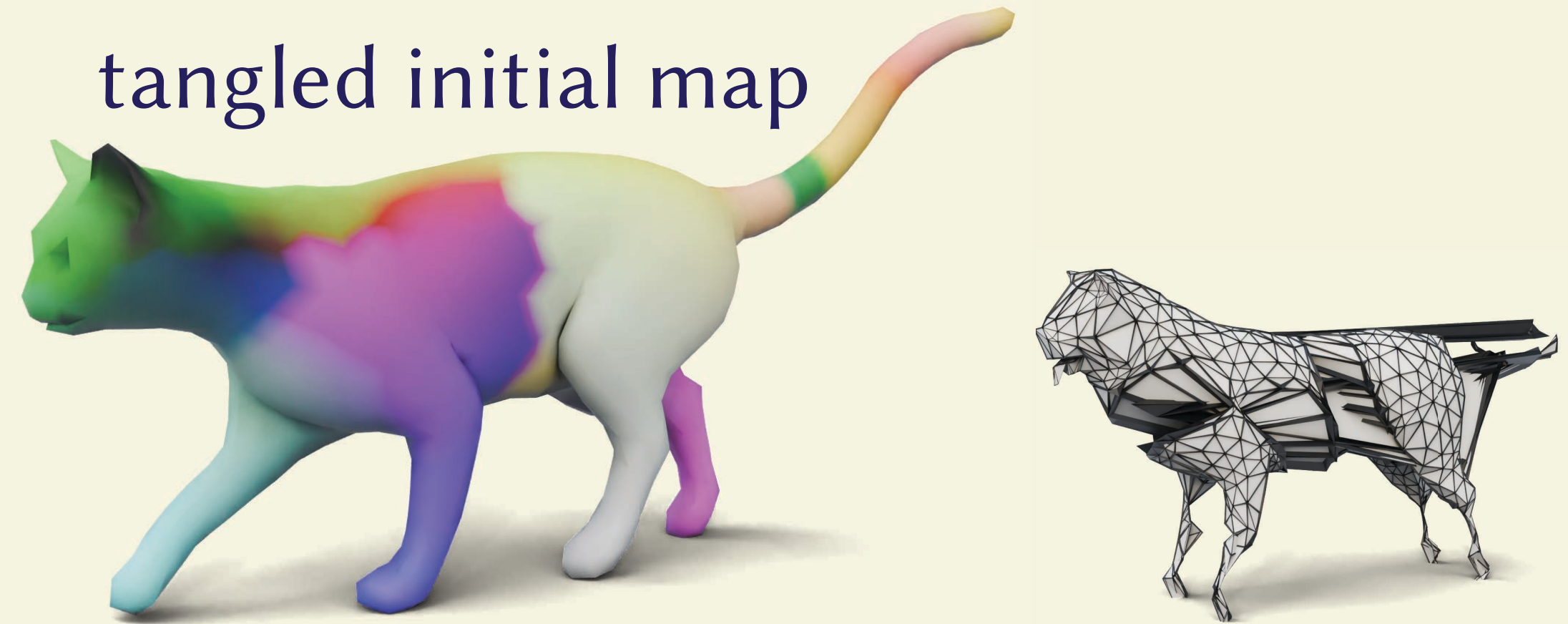
Examples and Comparisons



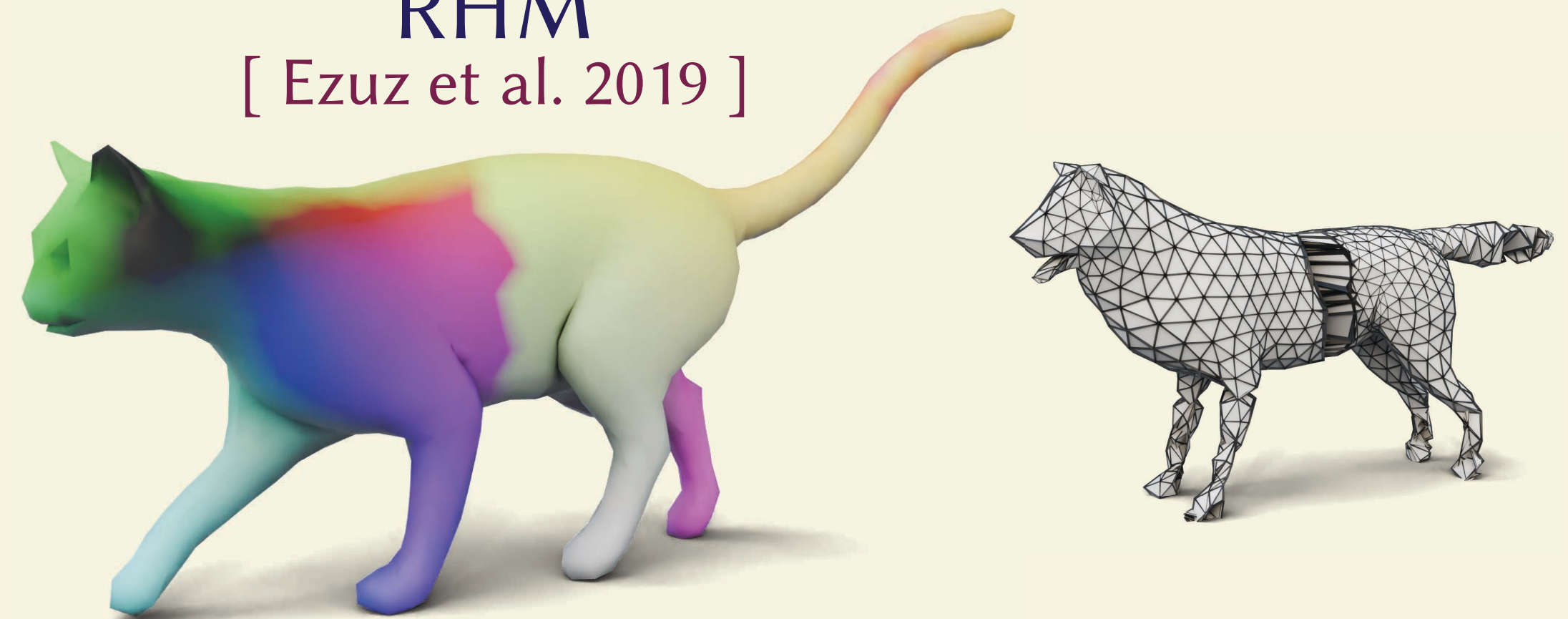
Map Untangling

Fixing Flips

tangled initial map



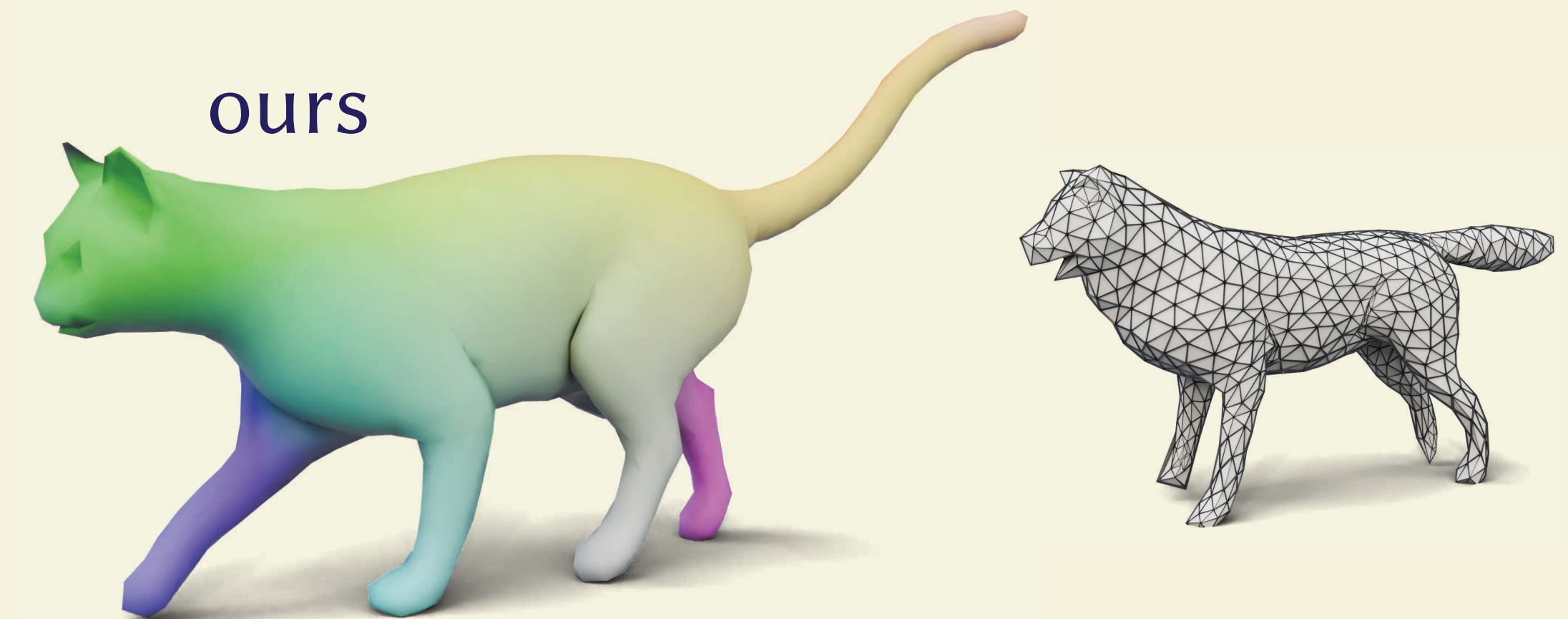
RHM
[Ezuz et al. 2019]



source



ours



Robustness to Initialization

Experiment from [Schmidt et al. 2020]

worse initialization

source mesh



RHM
[Ezuz et al.
2019]



ISM
[Schmidt
et al. 2020]



Ours



target mesh



worse initialization

Map Distortion

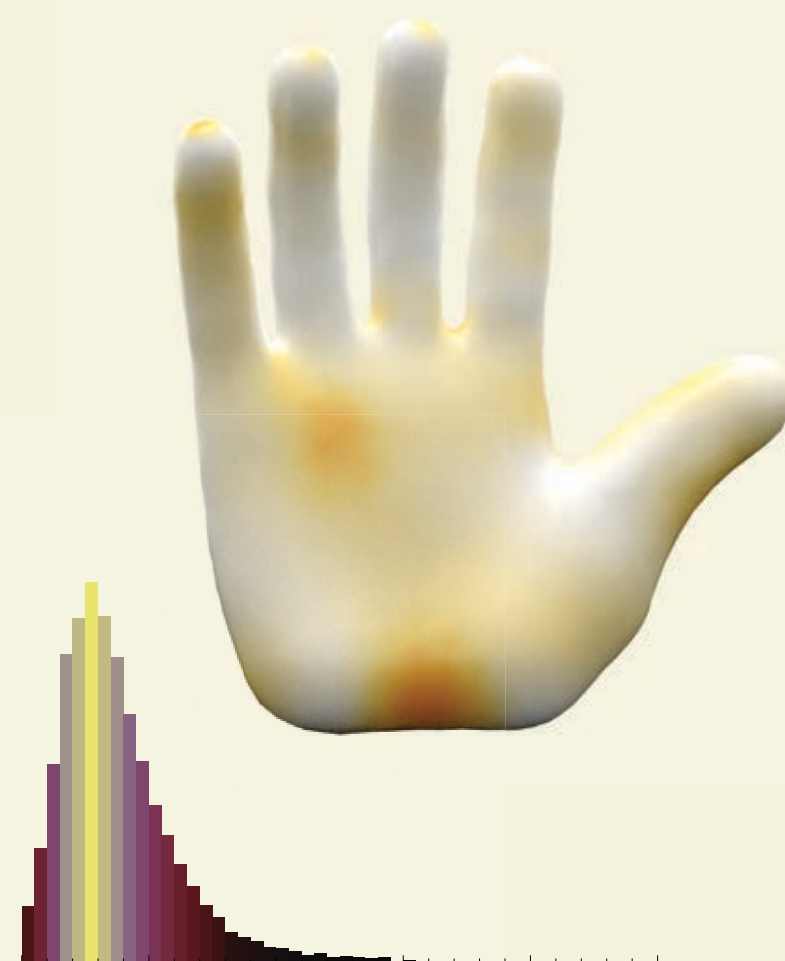


source mesh

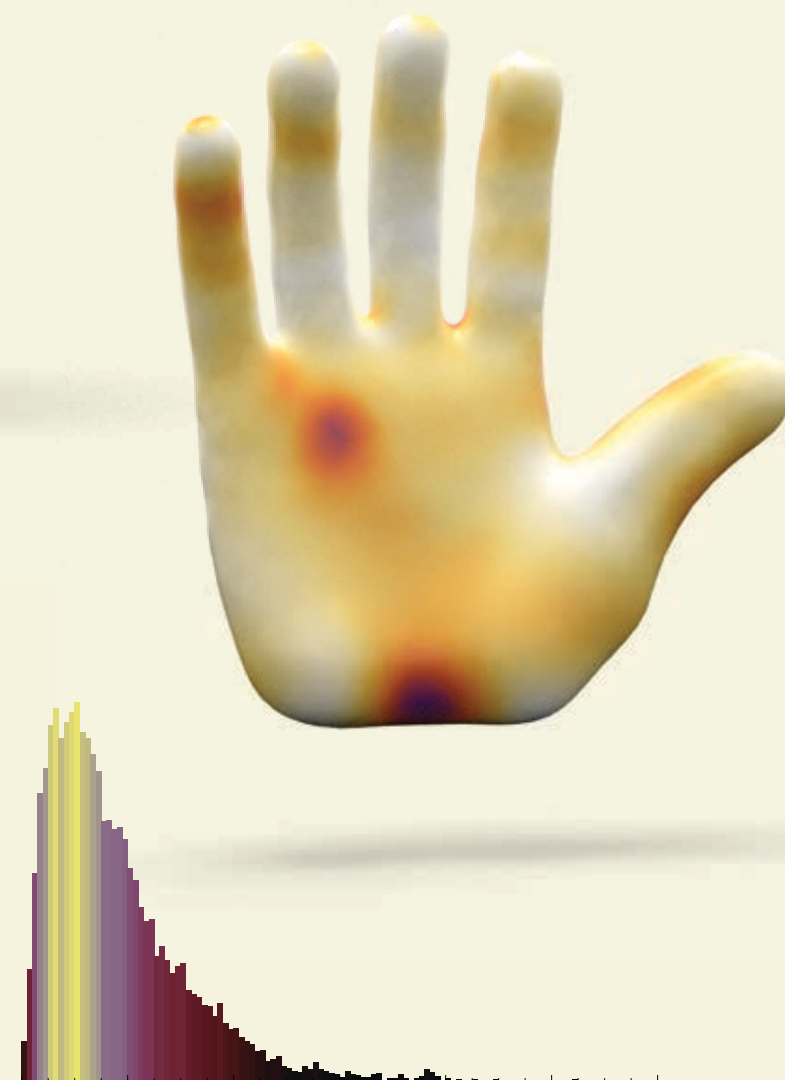


map
[ours]

distortion
(area)



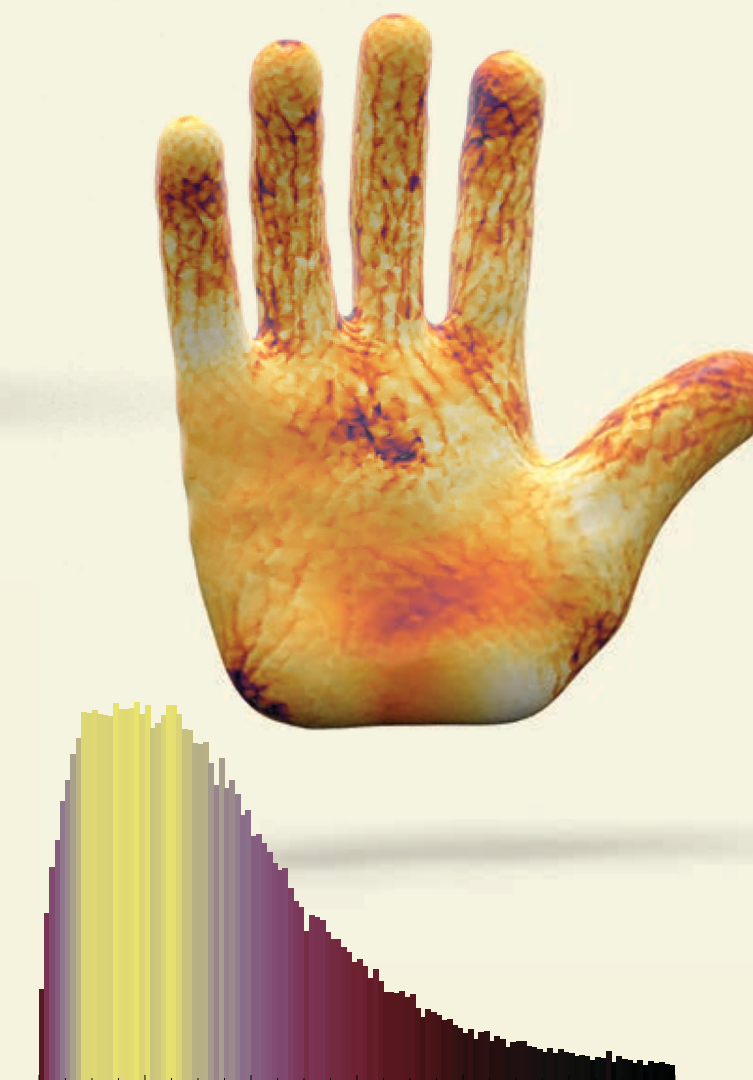
distortion
(symmetric Dirichlet)



distortion
(area)

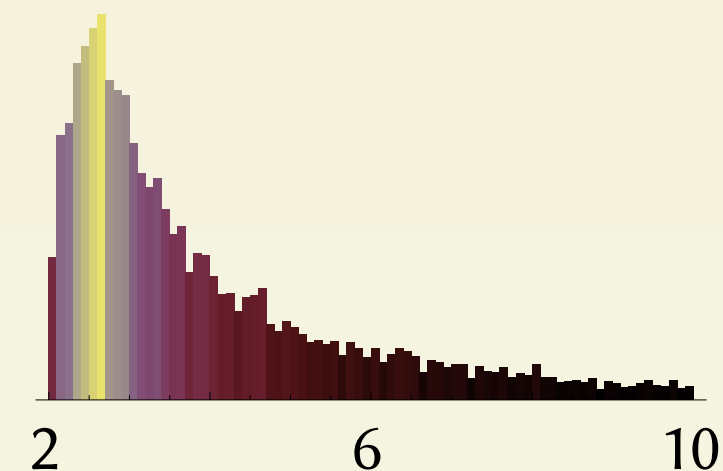
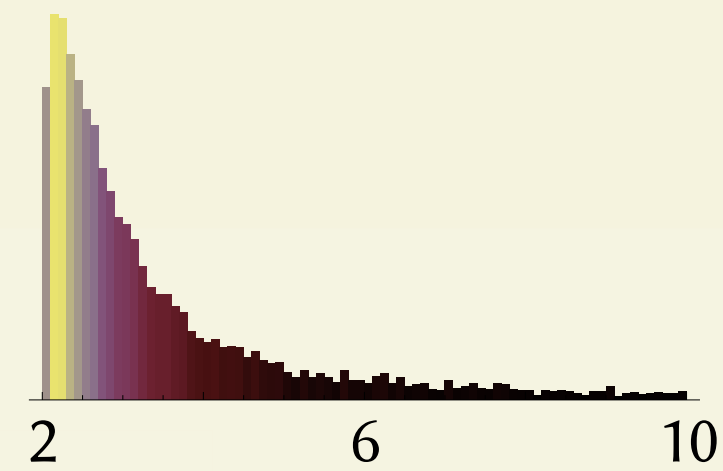


distortion
(symmetric Dirichlet)

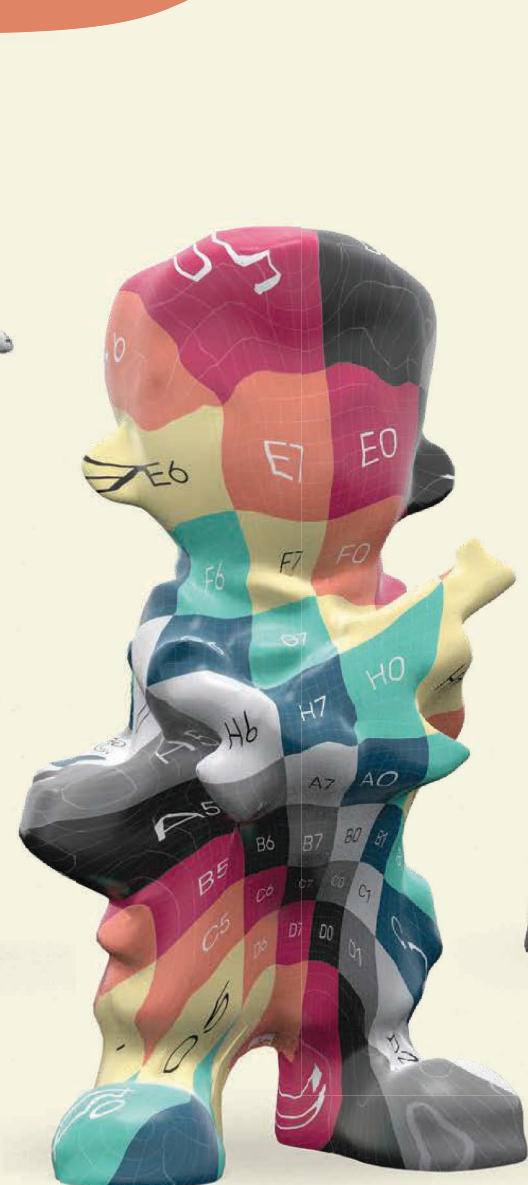


map
[Schmidt et al. 2020]

Map Distortion On Non-isometric Shapes



source mesh



map
[ours]



distortion (symmetric Dirichlet)

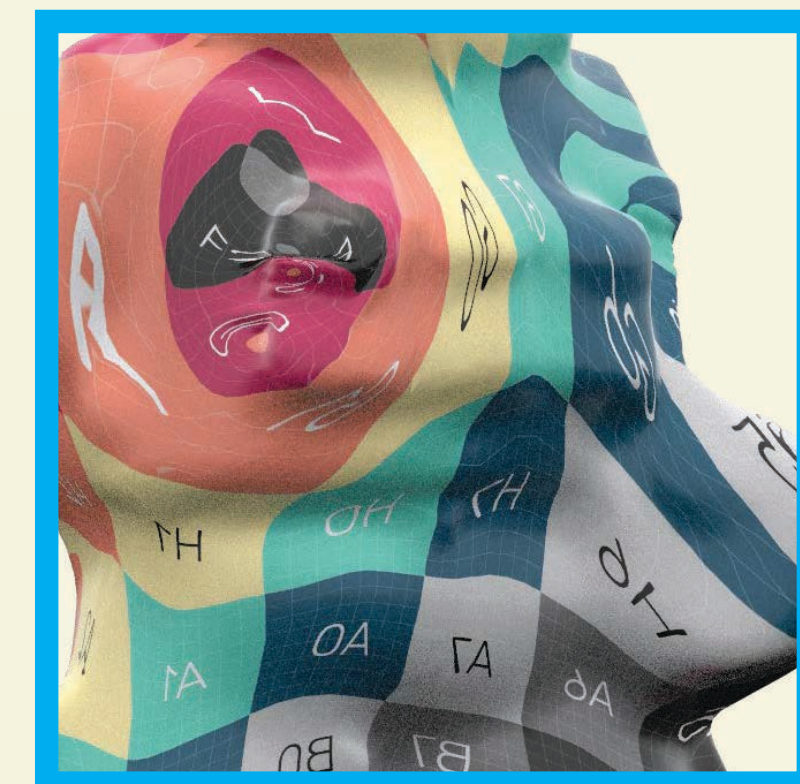
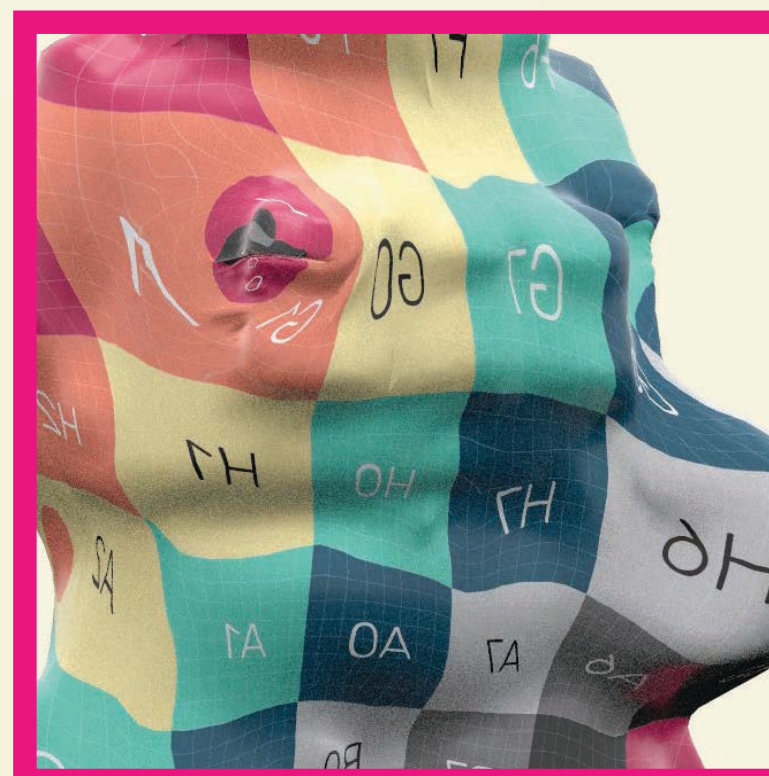


map
[Schmidt et al. 2023]



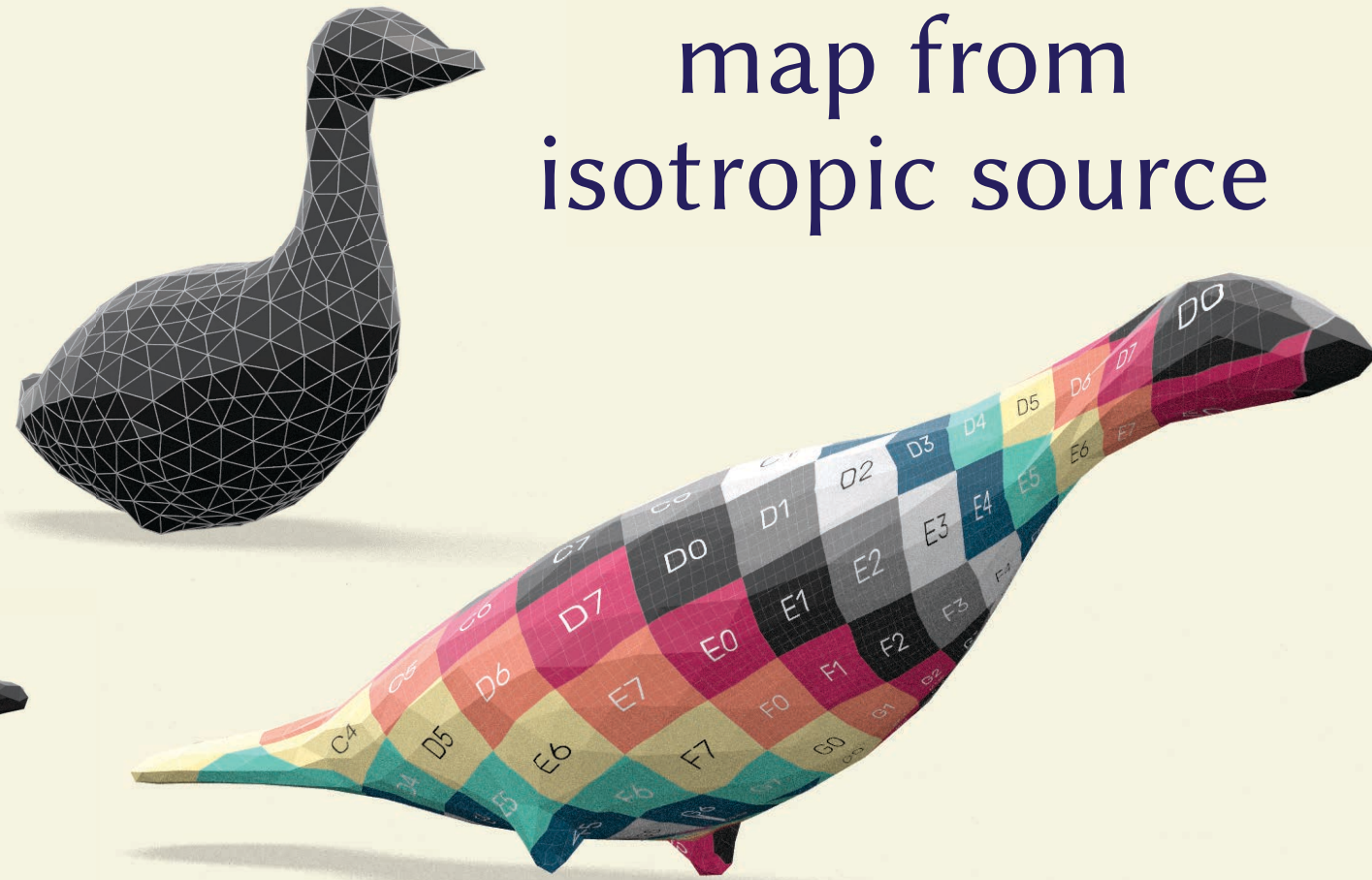
distortion

(symmetric Dirichlet)



Robustness to Triangle Quality

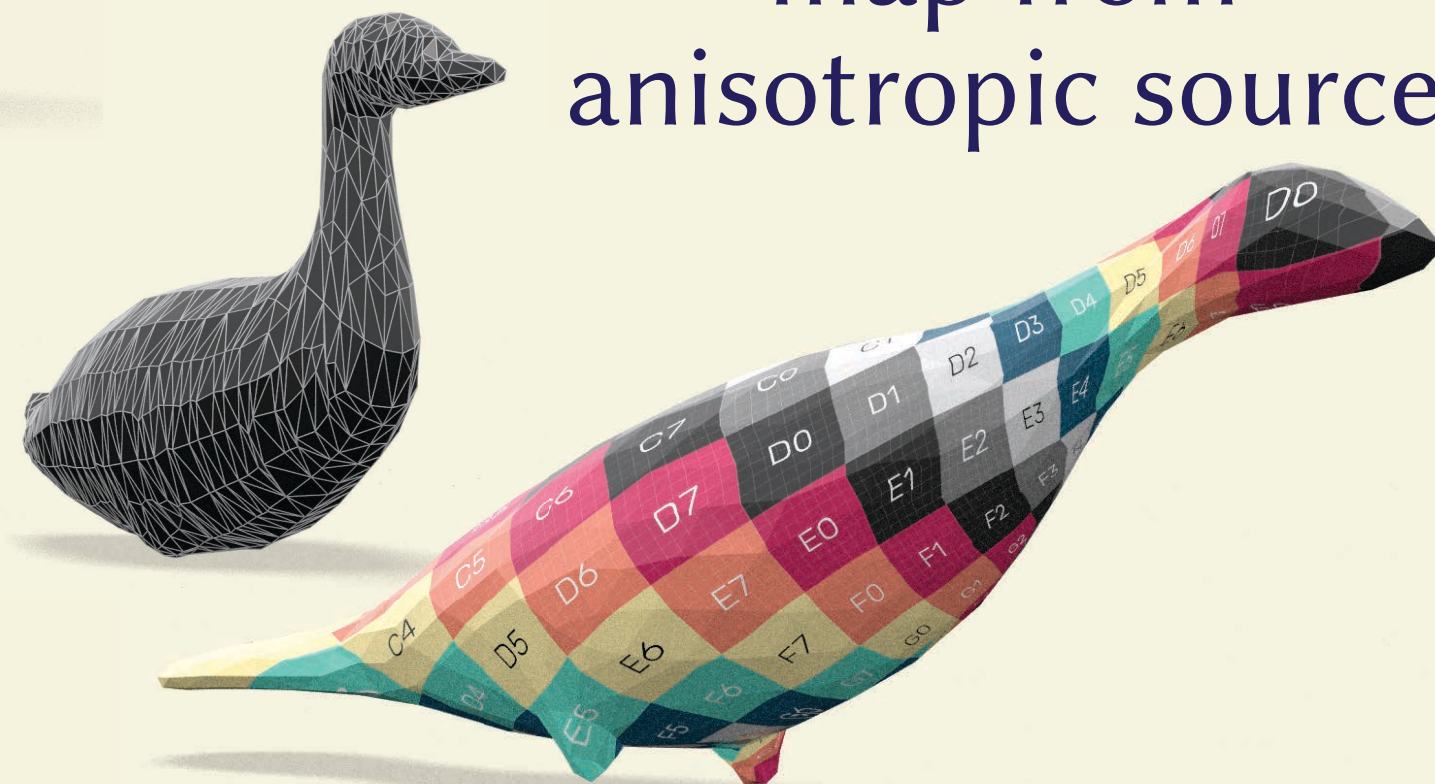
map from
isotropic source



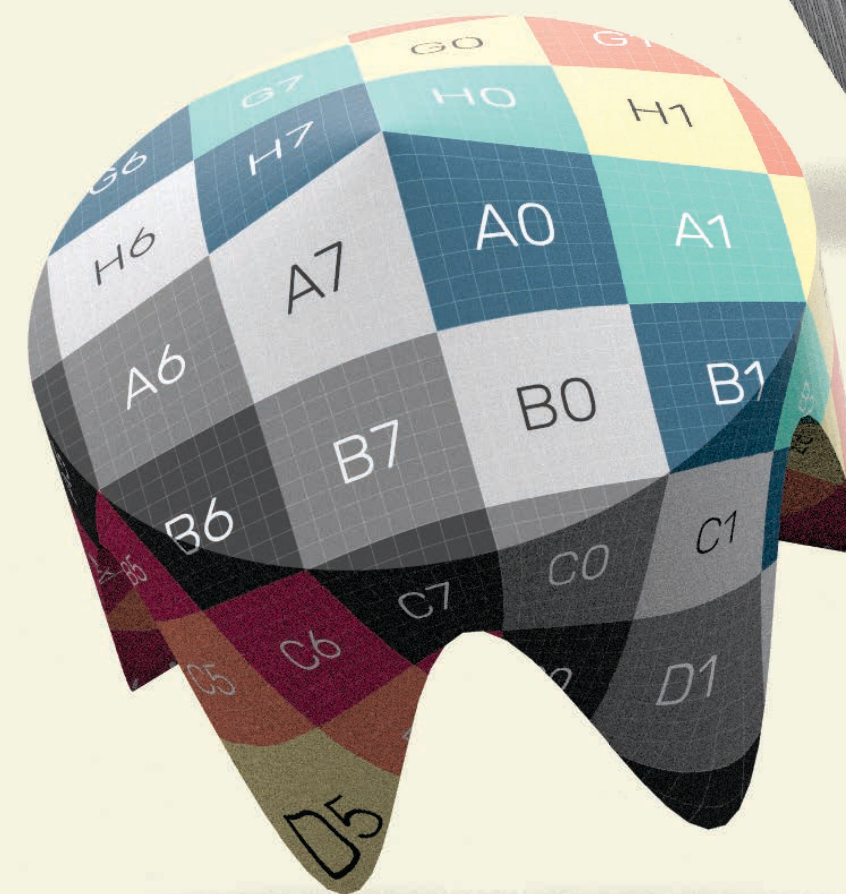
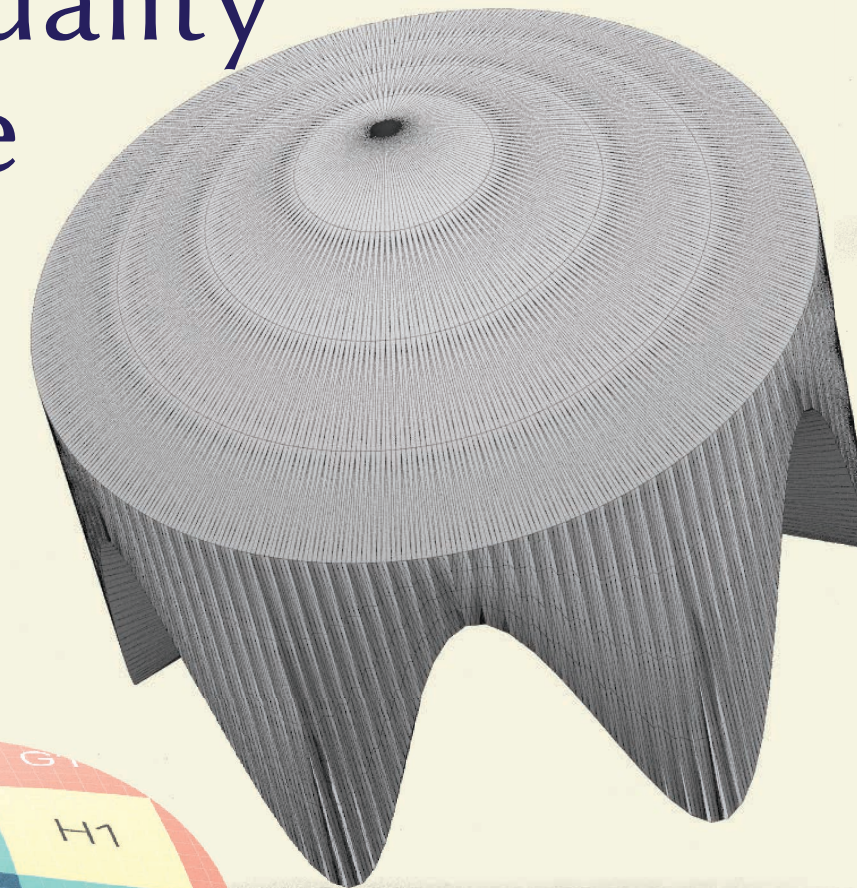
map from
anisotropic source



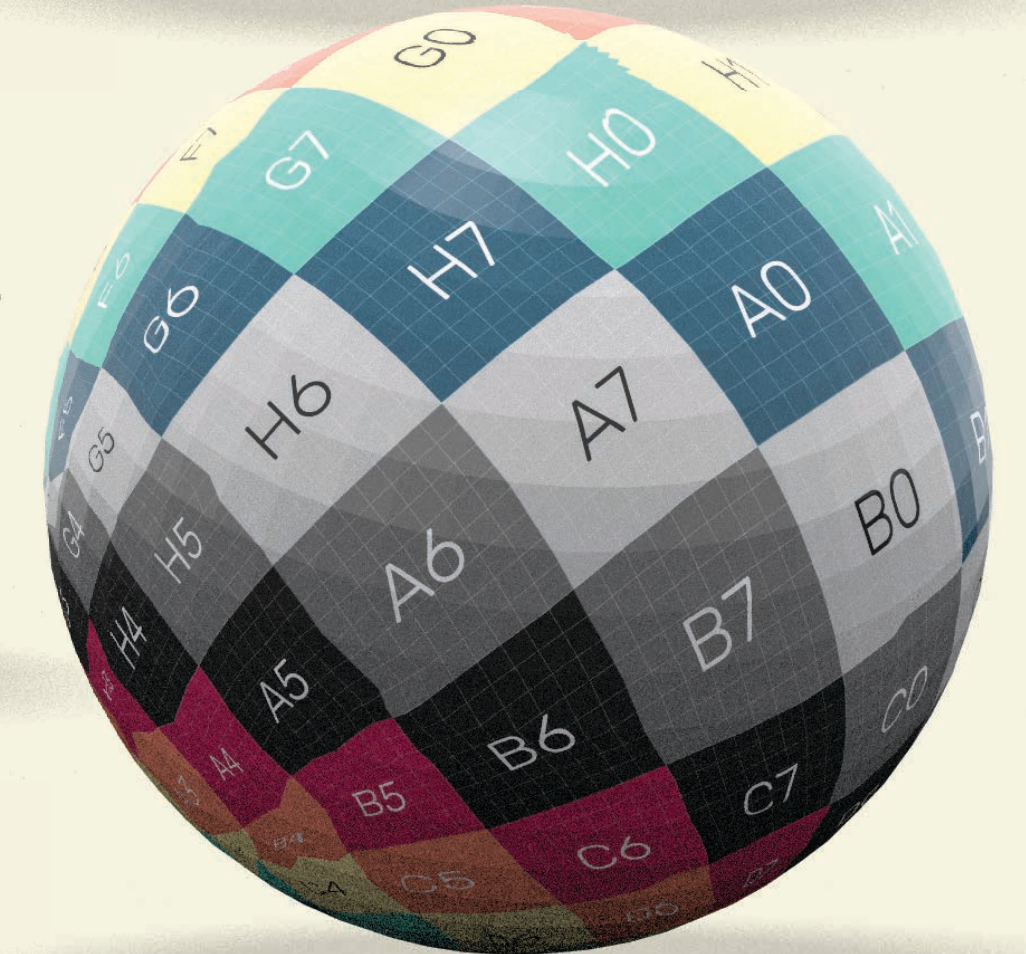
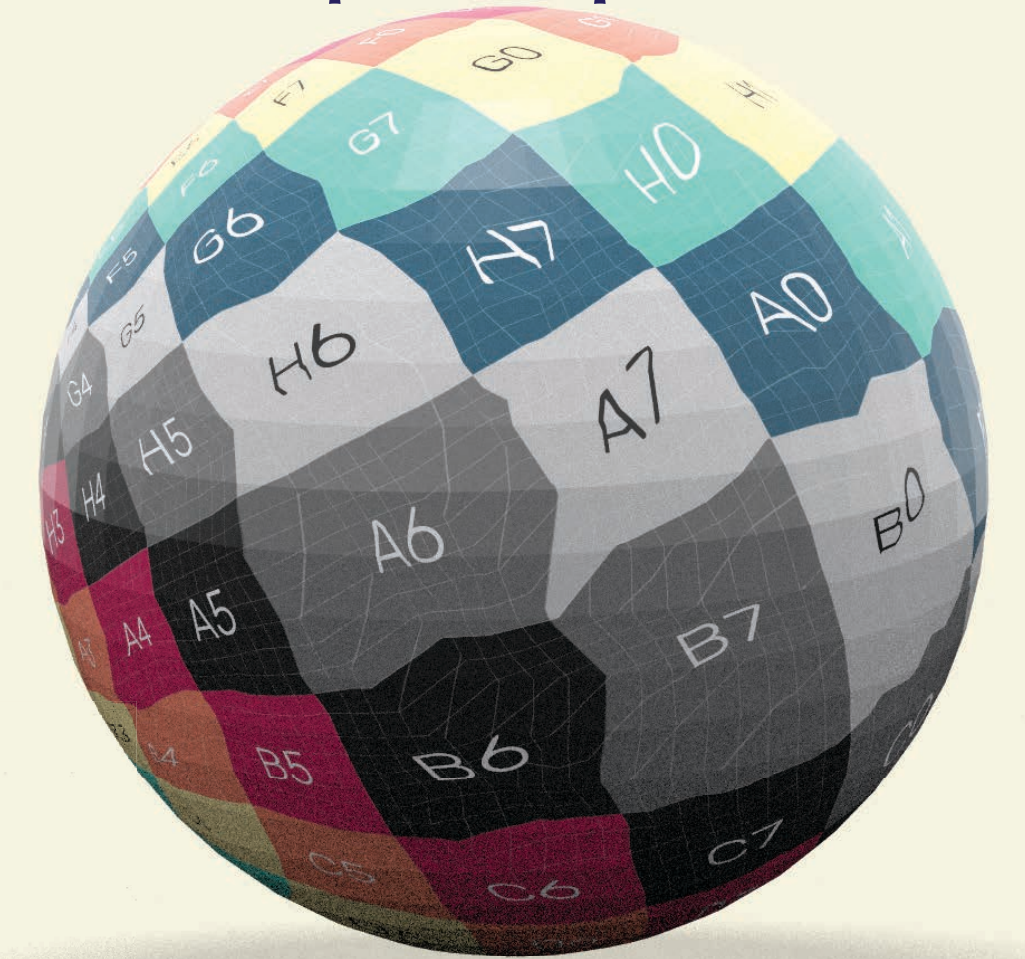
source



very low quality
source



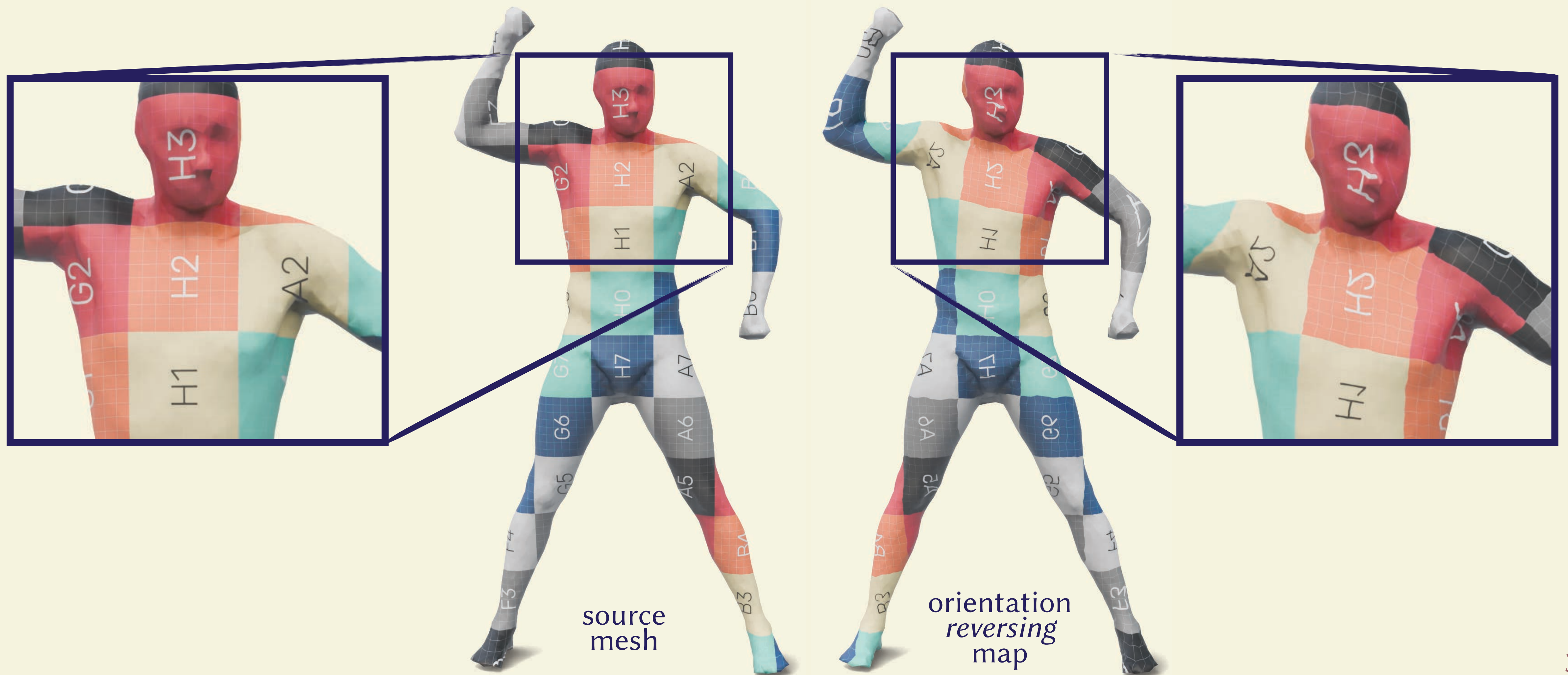
map to sphere



map to sphere using
intrinsic Delaunay
triangulation

[Bobenko and Springborn 2007;
Gillespie et al. 2021]

Detecting Symmetries



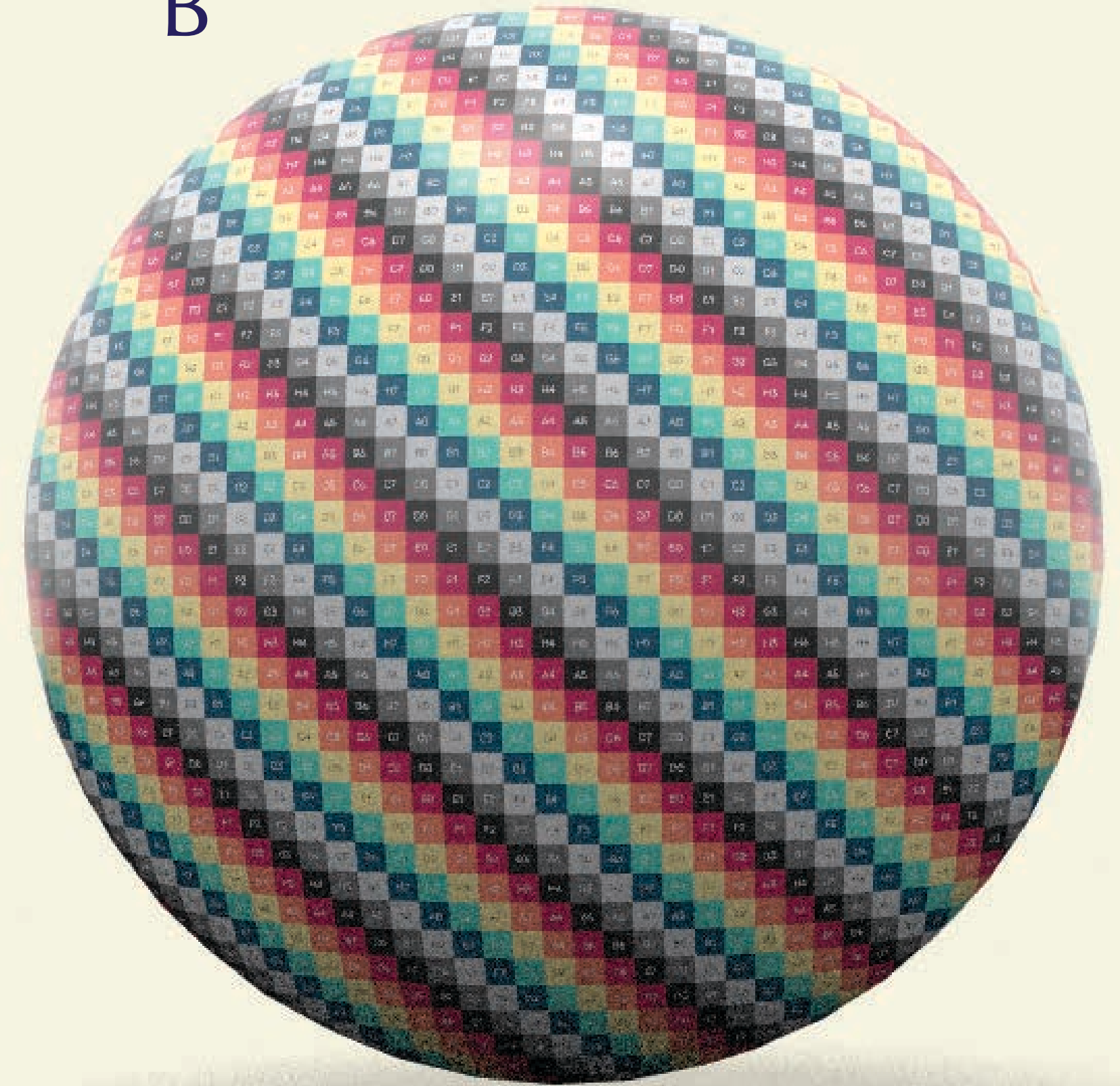
Beyond Bijections: Covering Maps

A

every point of A
appears twice on B

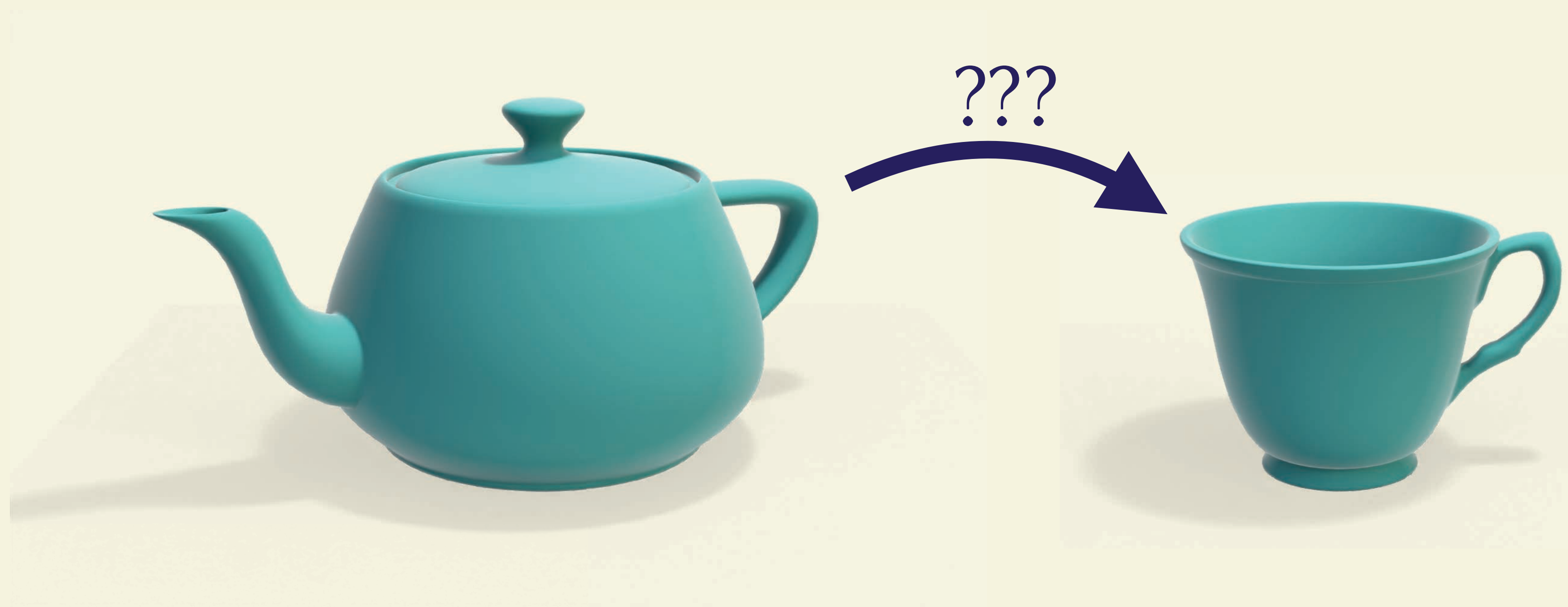


B



4 Future Work

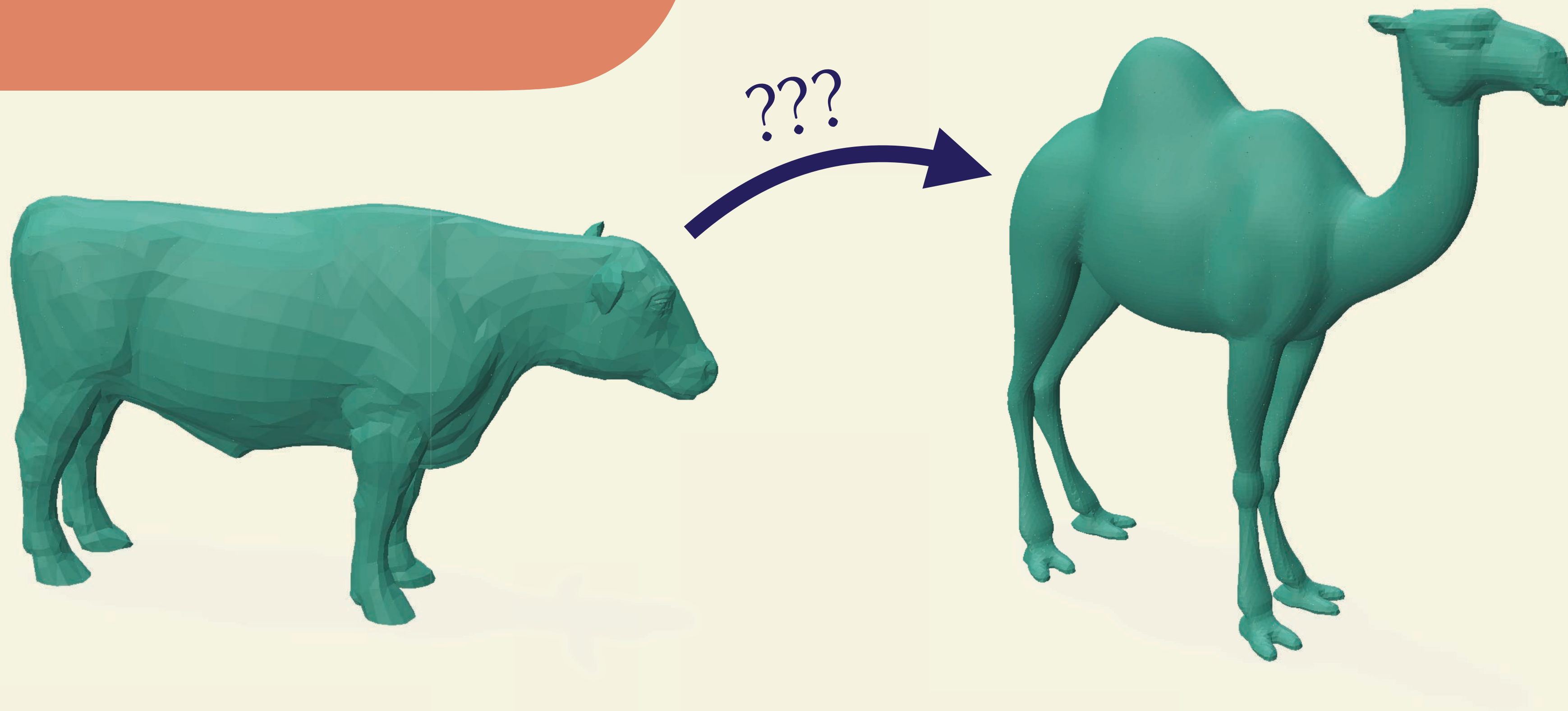
Surfaces with Handles



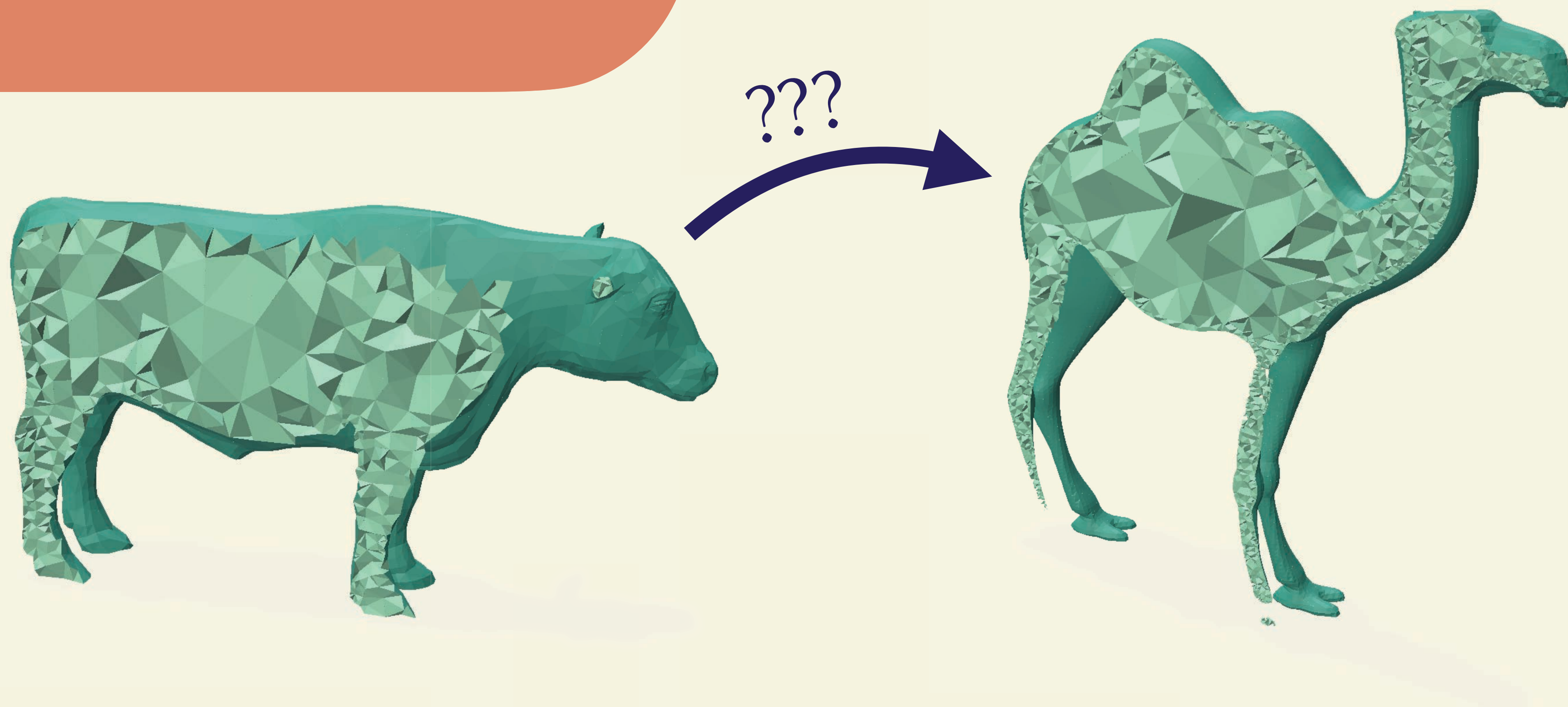
Optimize over both section *and* connection [Pigati & Stern 2021]

but then the optimization problem becomes much larger, and harder to solve

Volumetric Maps



Volumetric Maps



Replace complex numbers by 3×3 rotation matrices [Parise, Pigati & Stern 2025]

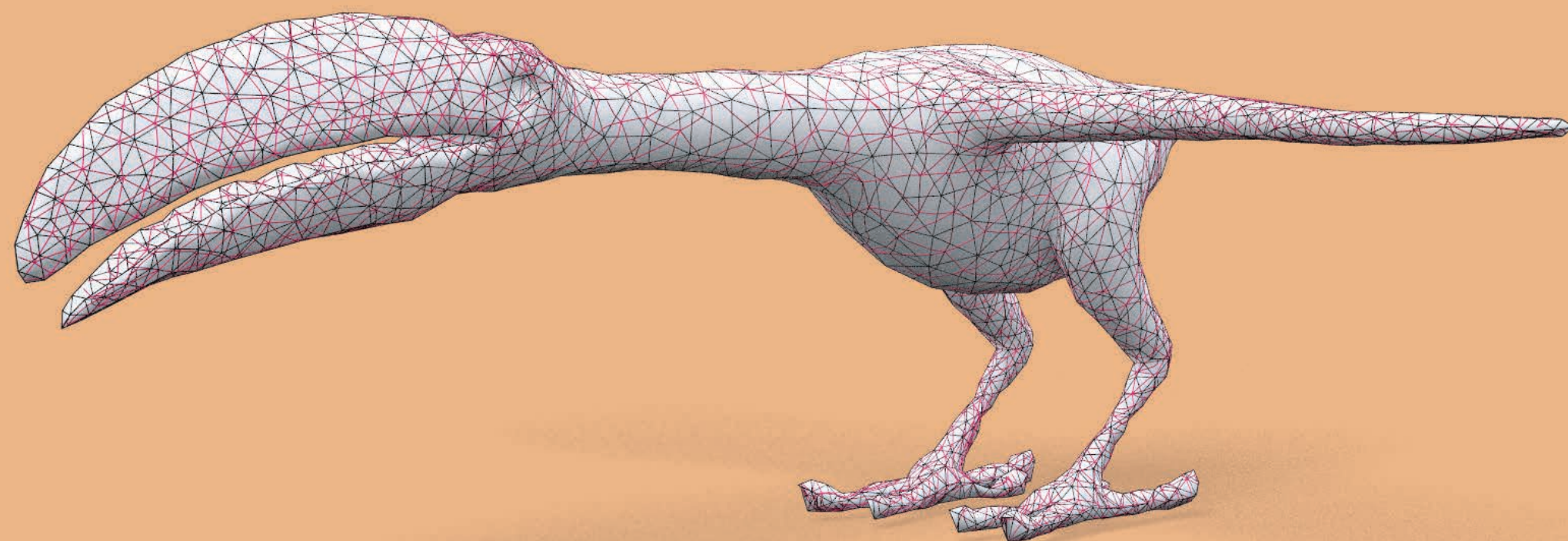
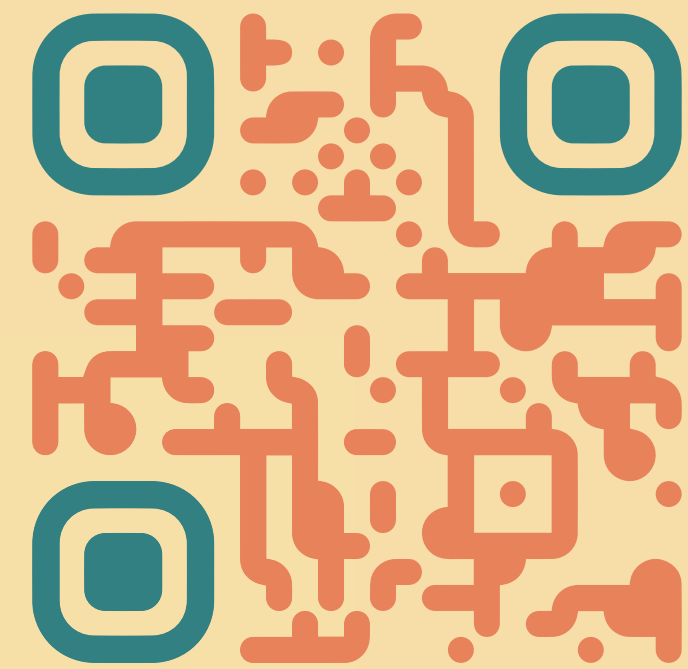
complex line bundles $\rightarrow$ quaternionic line bundles

everything gets much harder

Thanks for listening

✉ mark.gillespie81@gmail.com

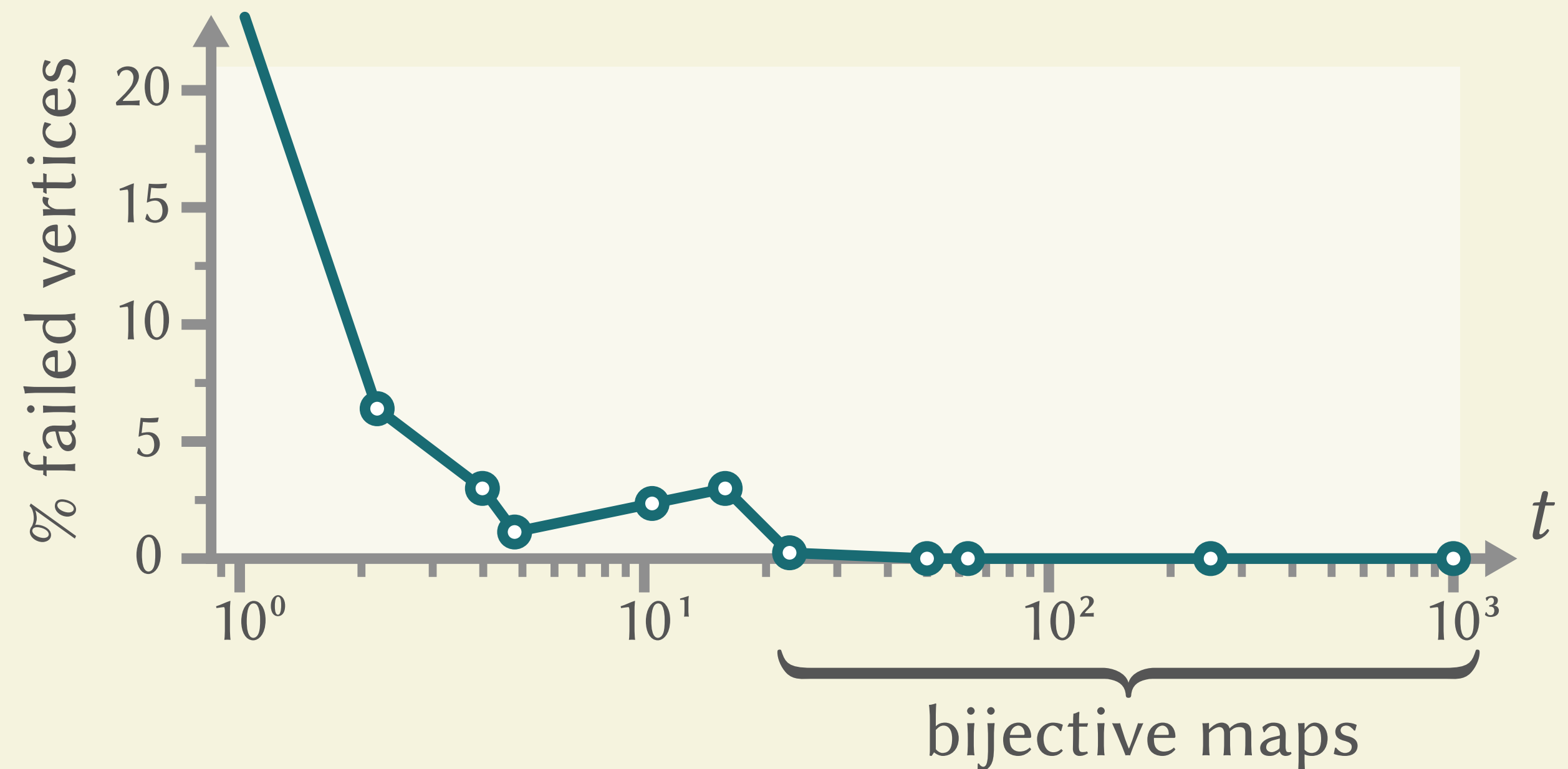
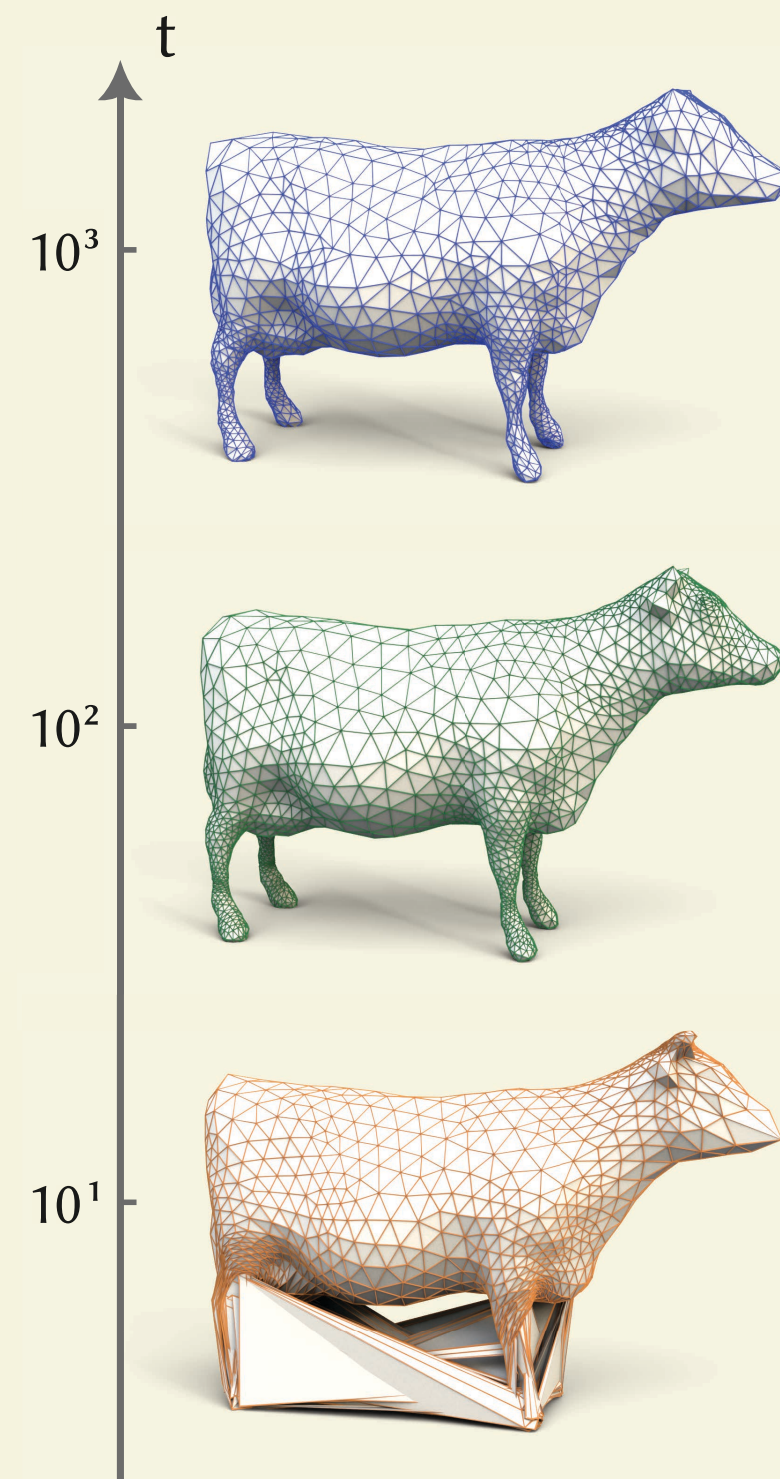
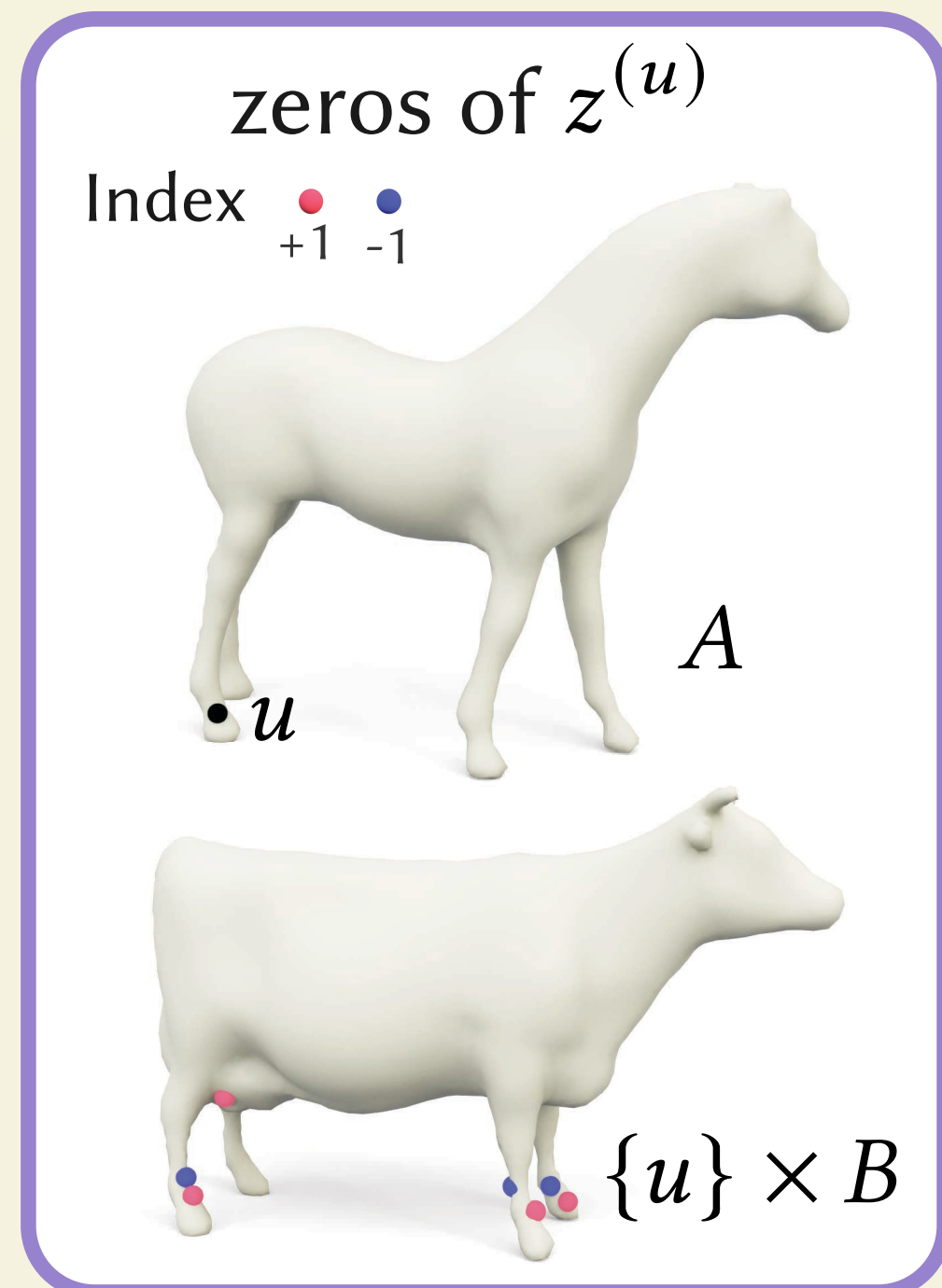
🌐 markjgillespie.com



Supplemental Slides

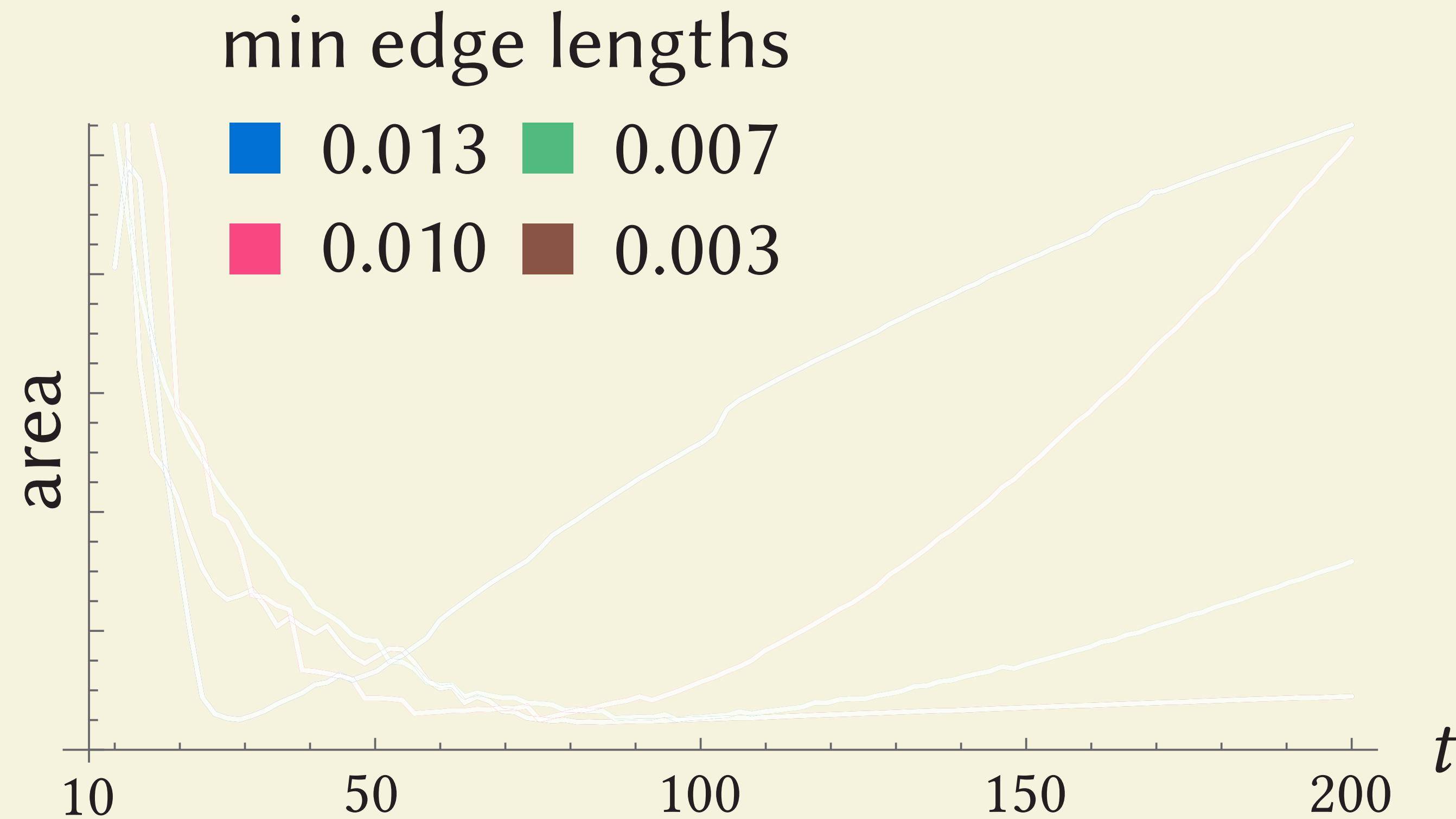
Picking the Ginzburg-Landau Parameter

$$\mathcal{E}_\epsilon^{\mathcal{GL}}(z) := \int_A \int_B \frac{1}{2} |\nabla z(p)|^2 + \frac{1}{4\epsilon^2} \left(|z(p)|^2 - 1 \right)^2 dA dB \quad \text{set } \frac{1}{\epsilon^2} \propto \lambda_0, \text{ the smallest eigenvalue of the connection Laplacian}$$

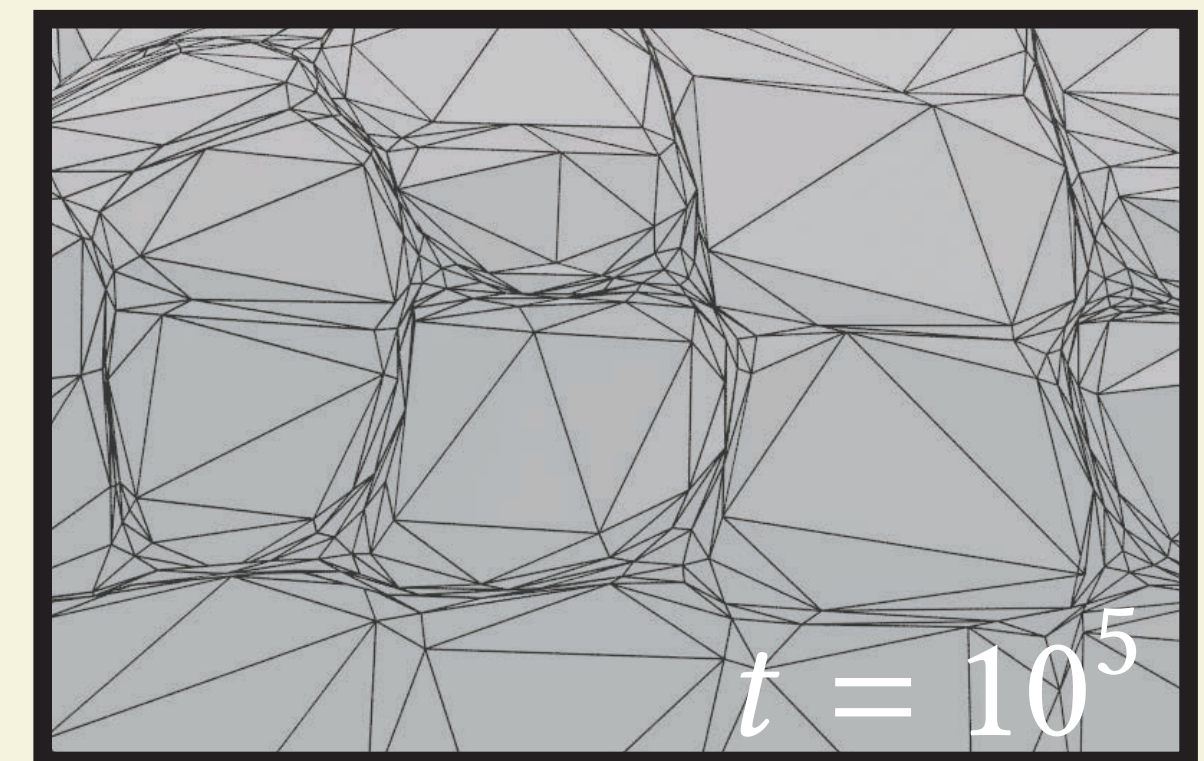
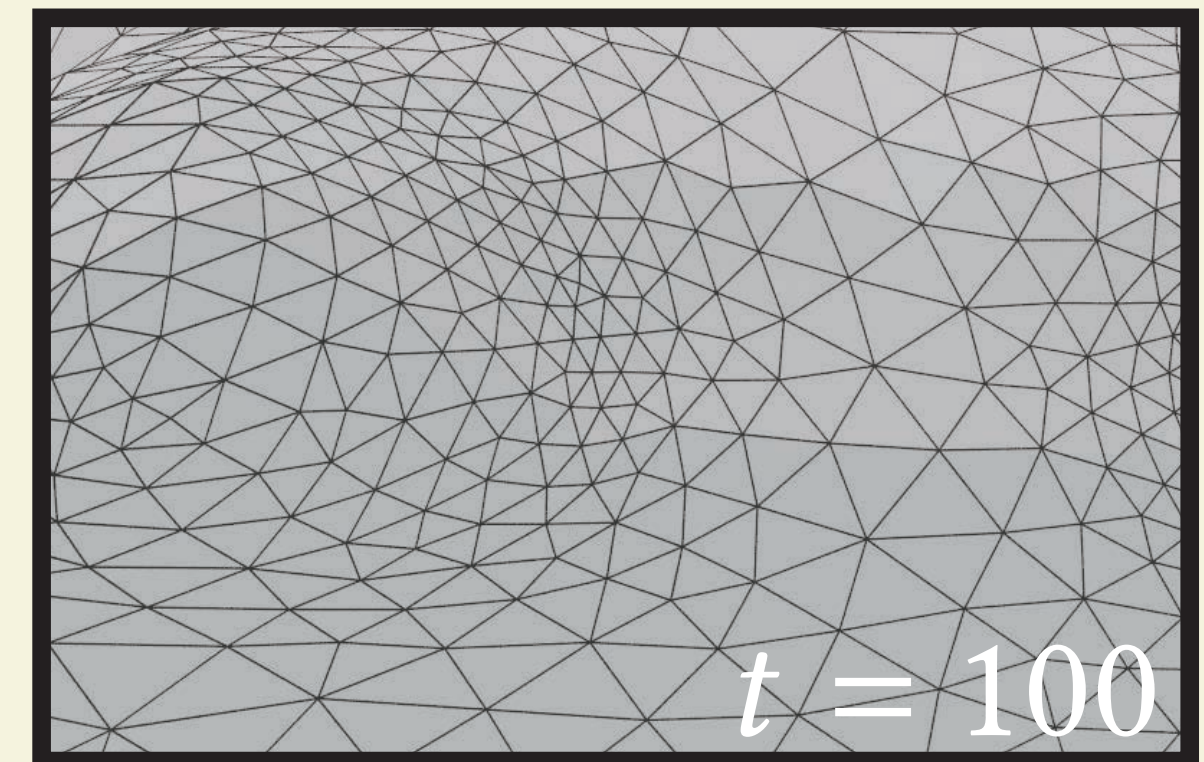


Picking the Ginzburg-Landau Parameter

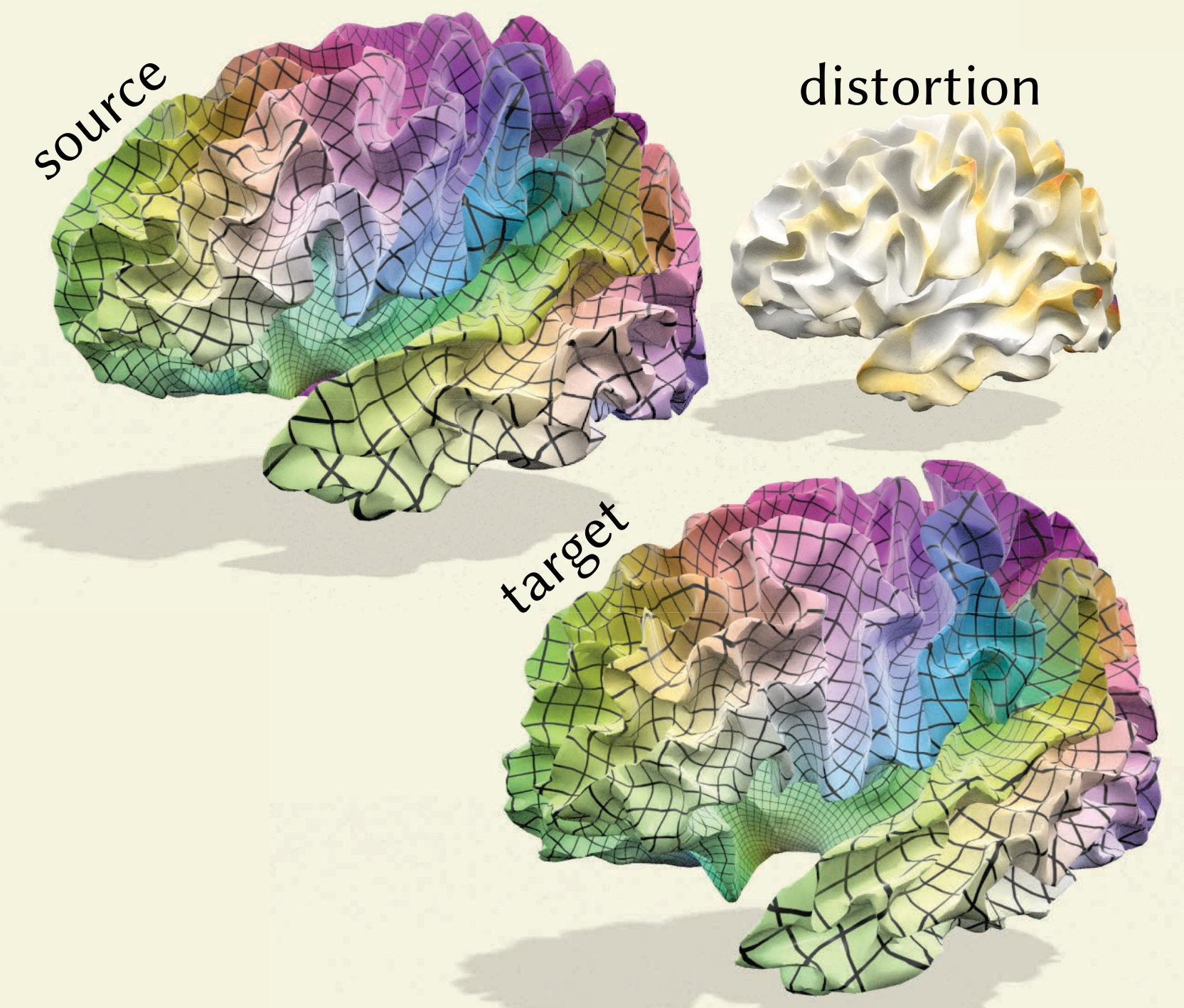
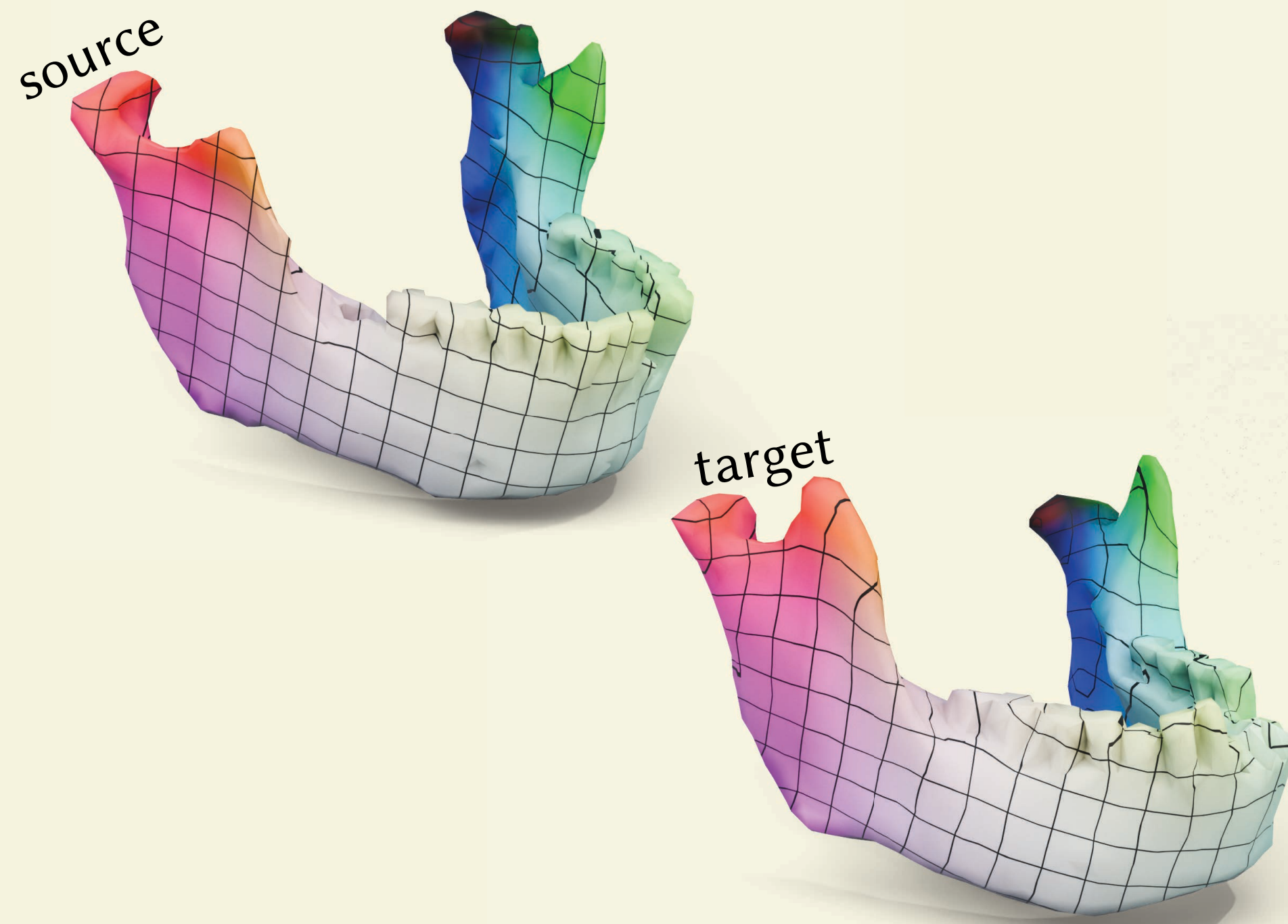
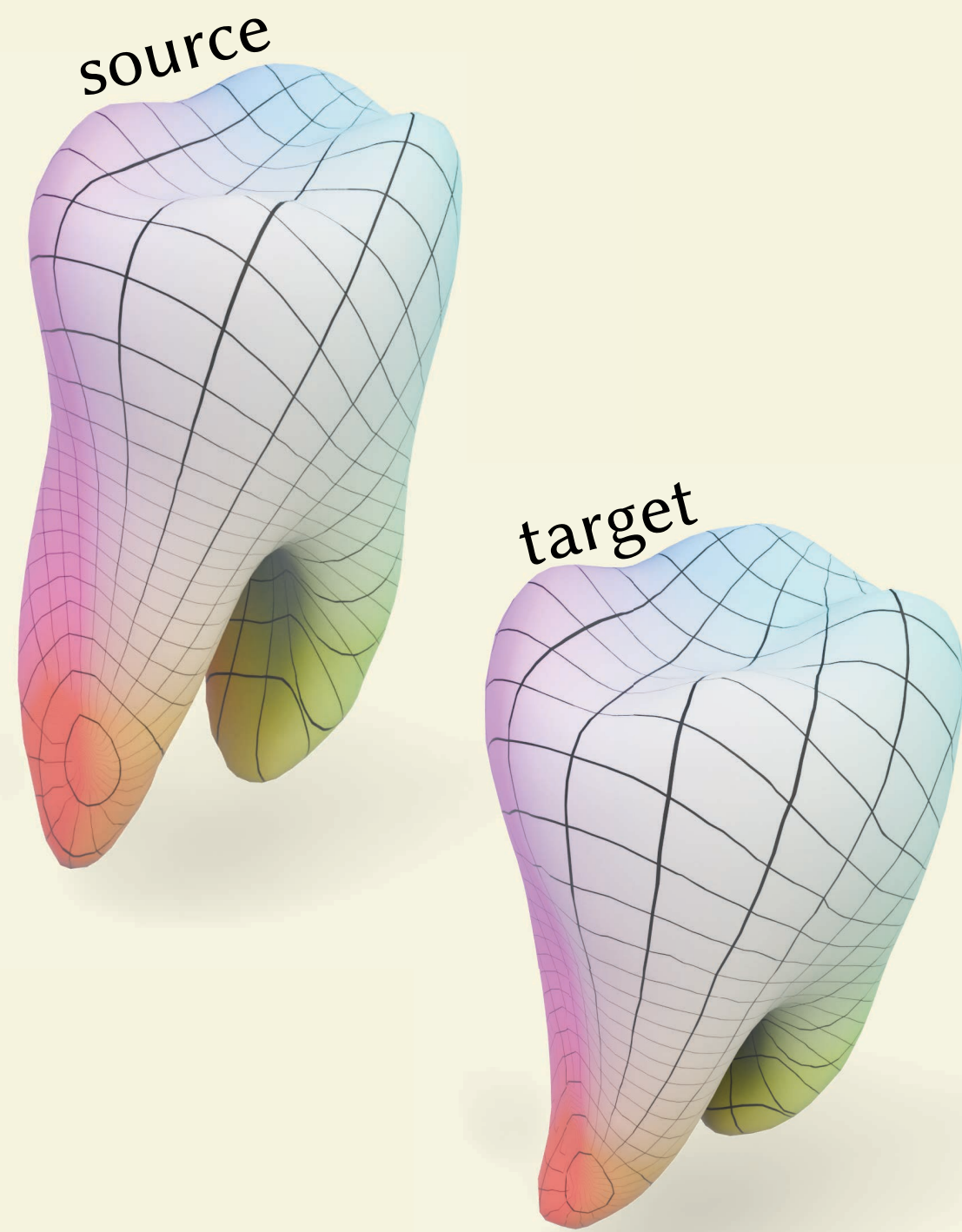
An experimental search for the optimal choice of Ginzburg-Landau parameter $\frac{1}{\varepsilon^2} = t\lambda_0$ where $\lambda_0 = \lambda_A + \lambda_B$.



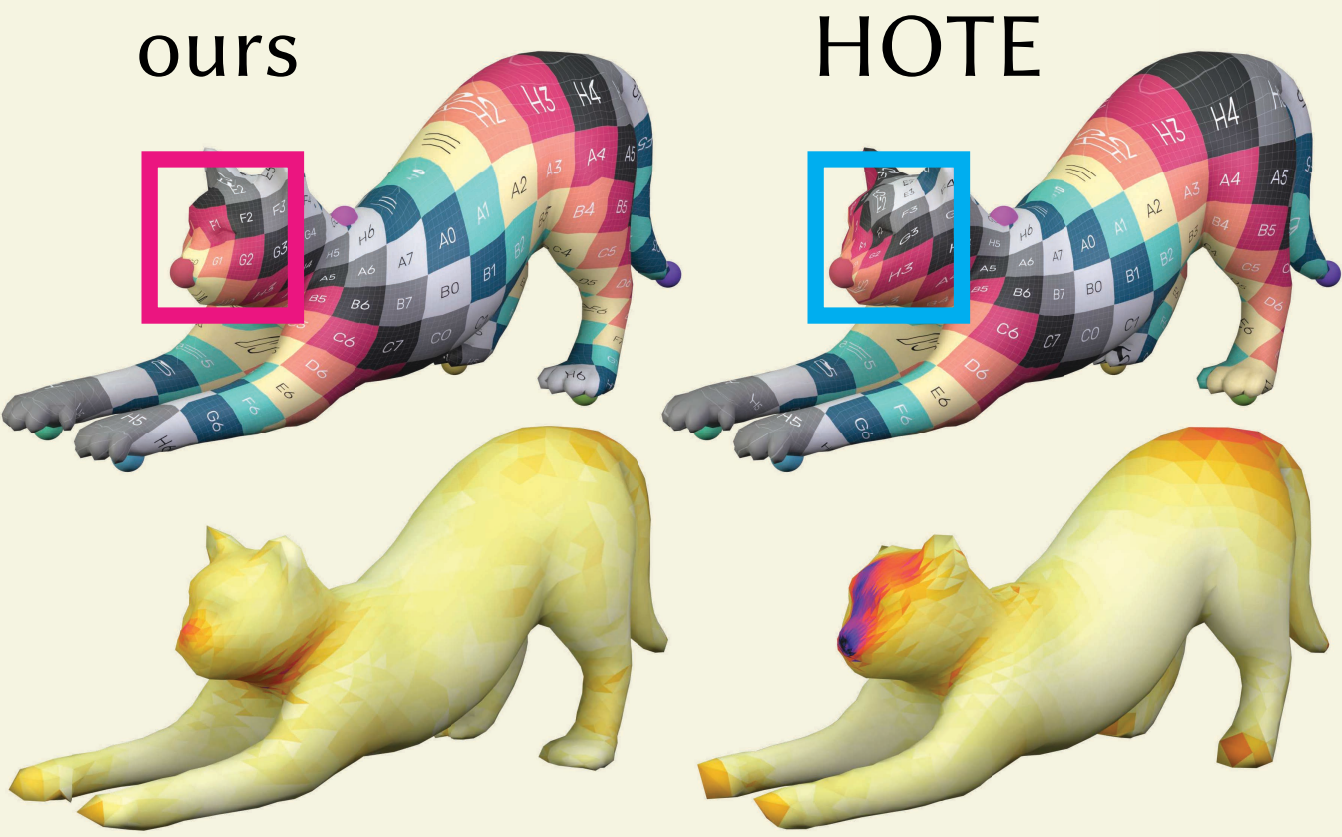
typical distortion



Gallery

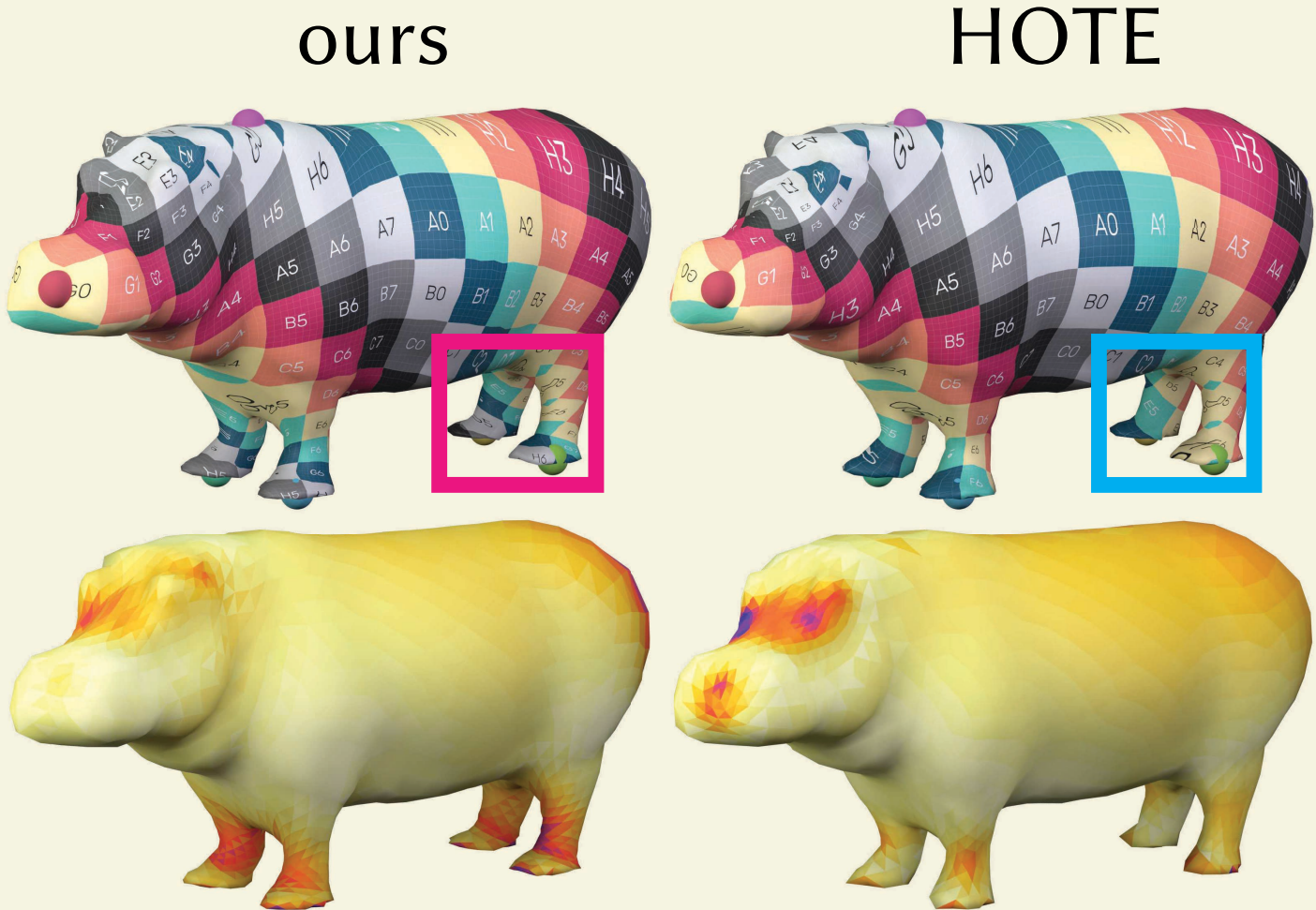
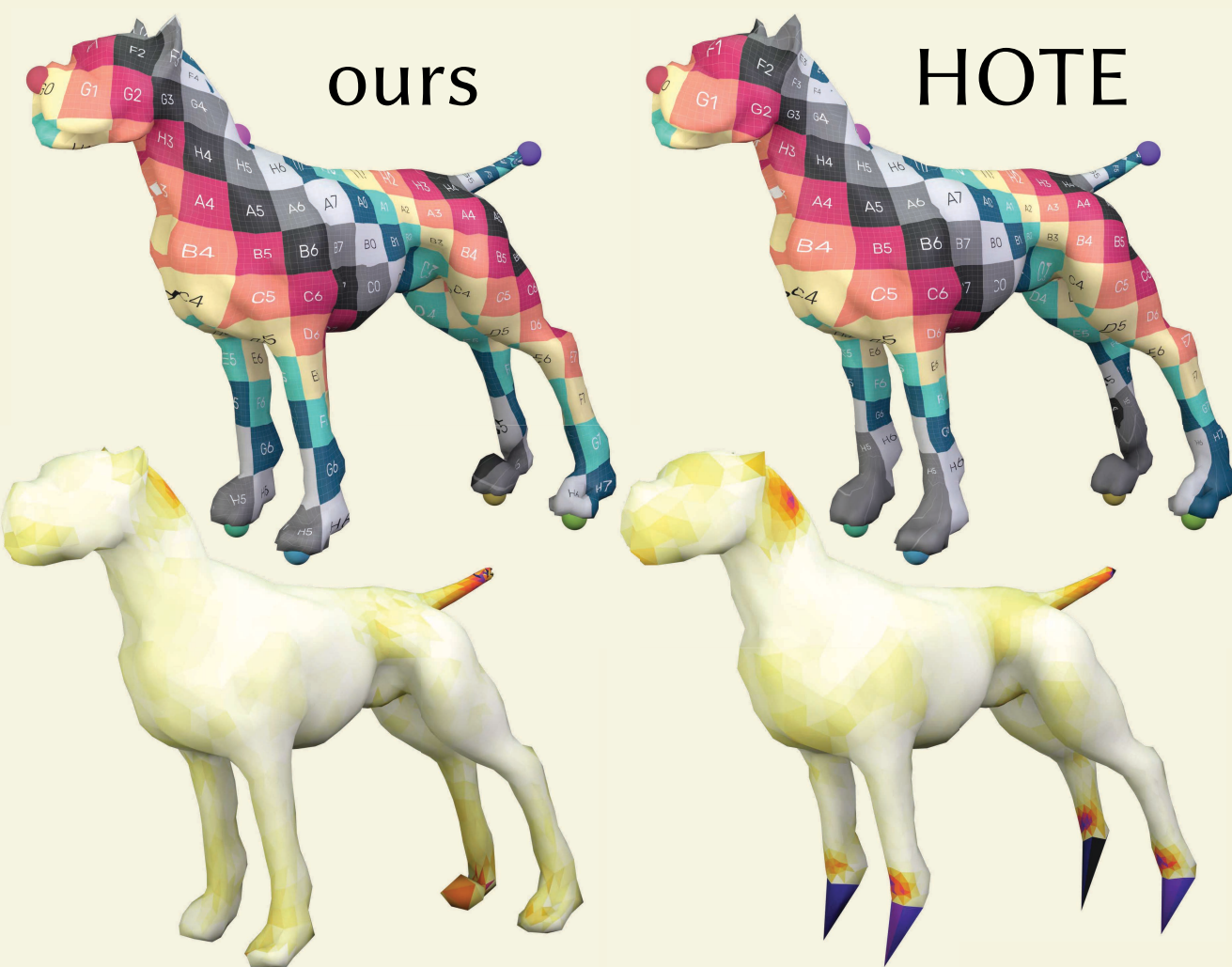


Orbifold Comparison



Symmetric Dirichlet Energy

	ours	HOTE
cat	2.7	13.5
hippo	8.9	25582.5
dog	3.4	14.8



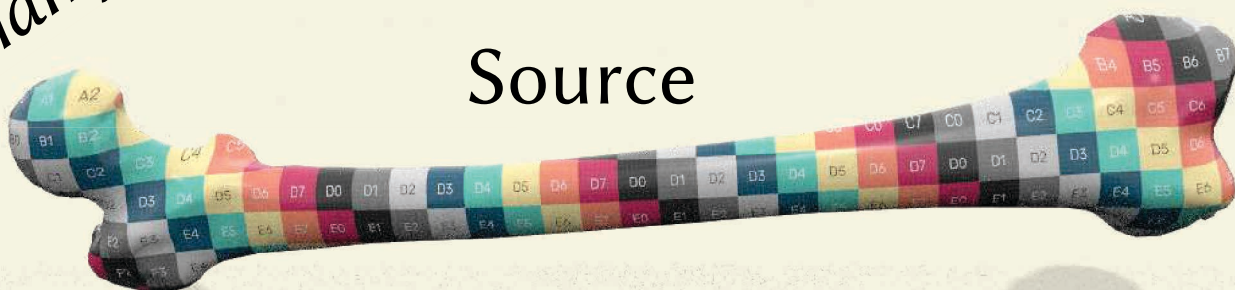
Femur Comparison

Symmetric Dirichlet Energy

	ours	AT
bear	1.94	2.16
deer	2.02	2.22
pig	2.41	2.55

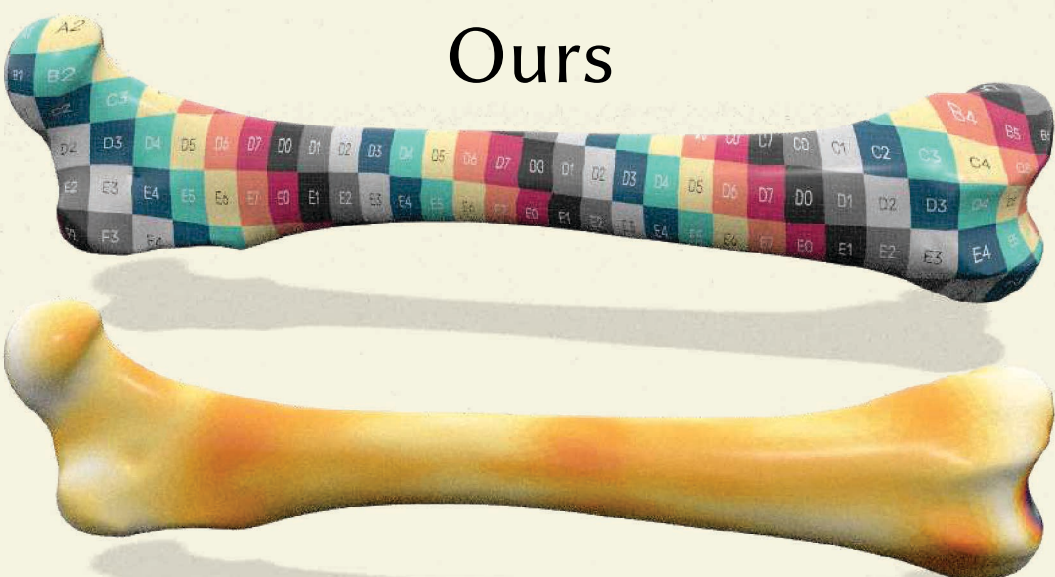
human femur

Source



bear

Ours

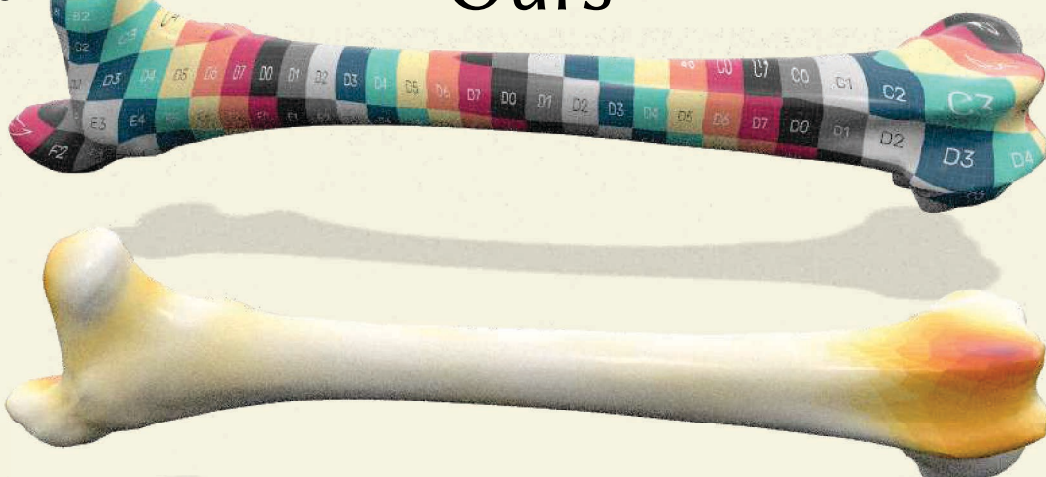


AT

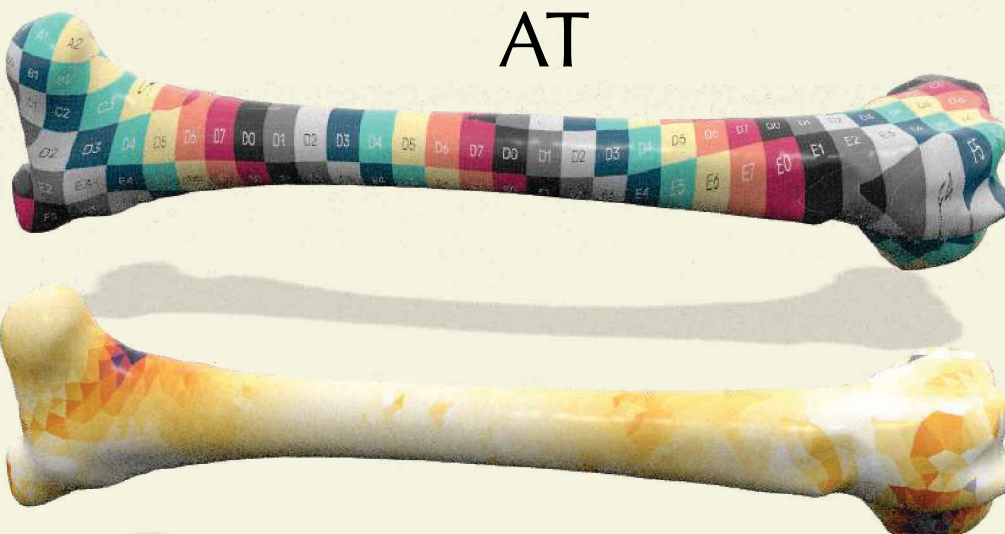


deer

Ours



AT



pig

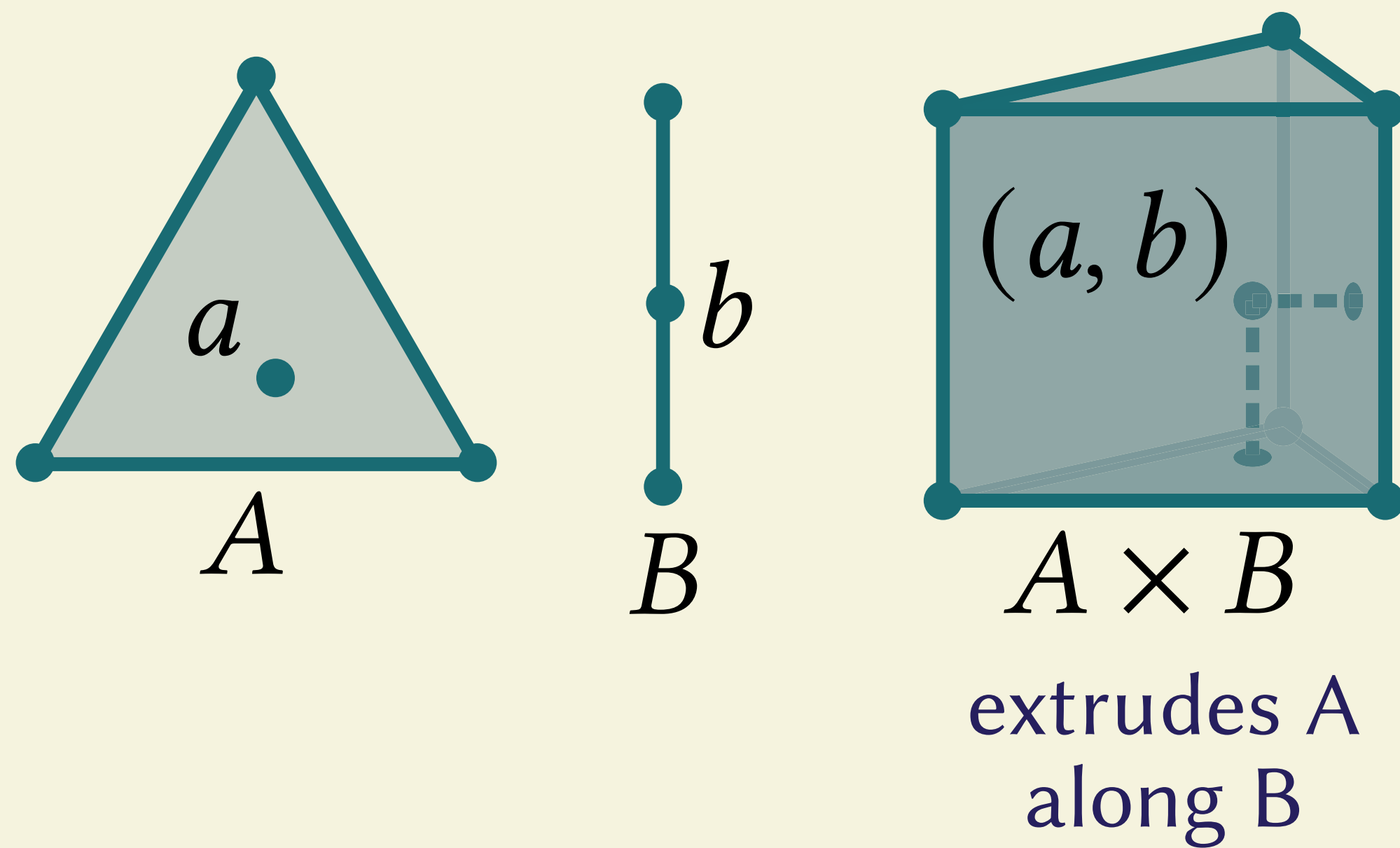
Ours



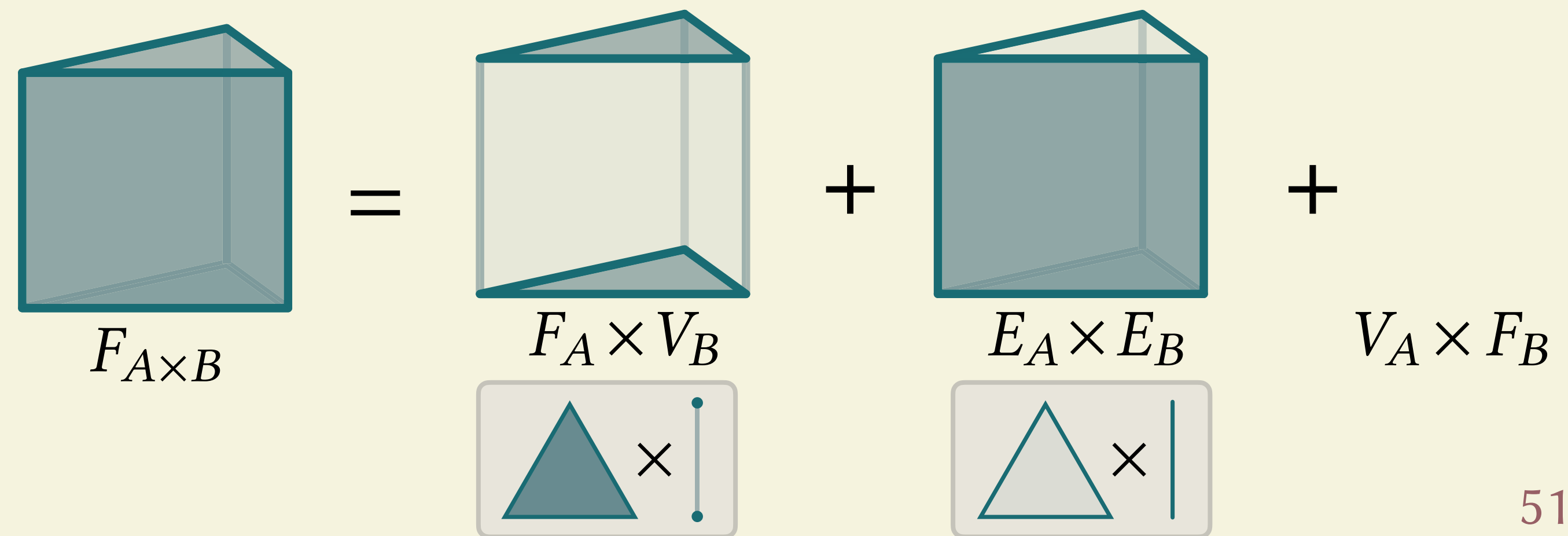
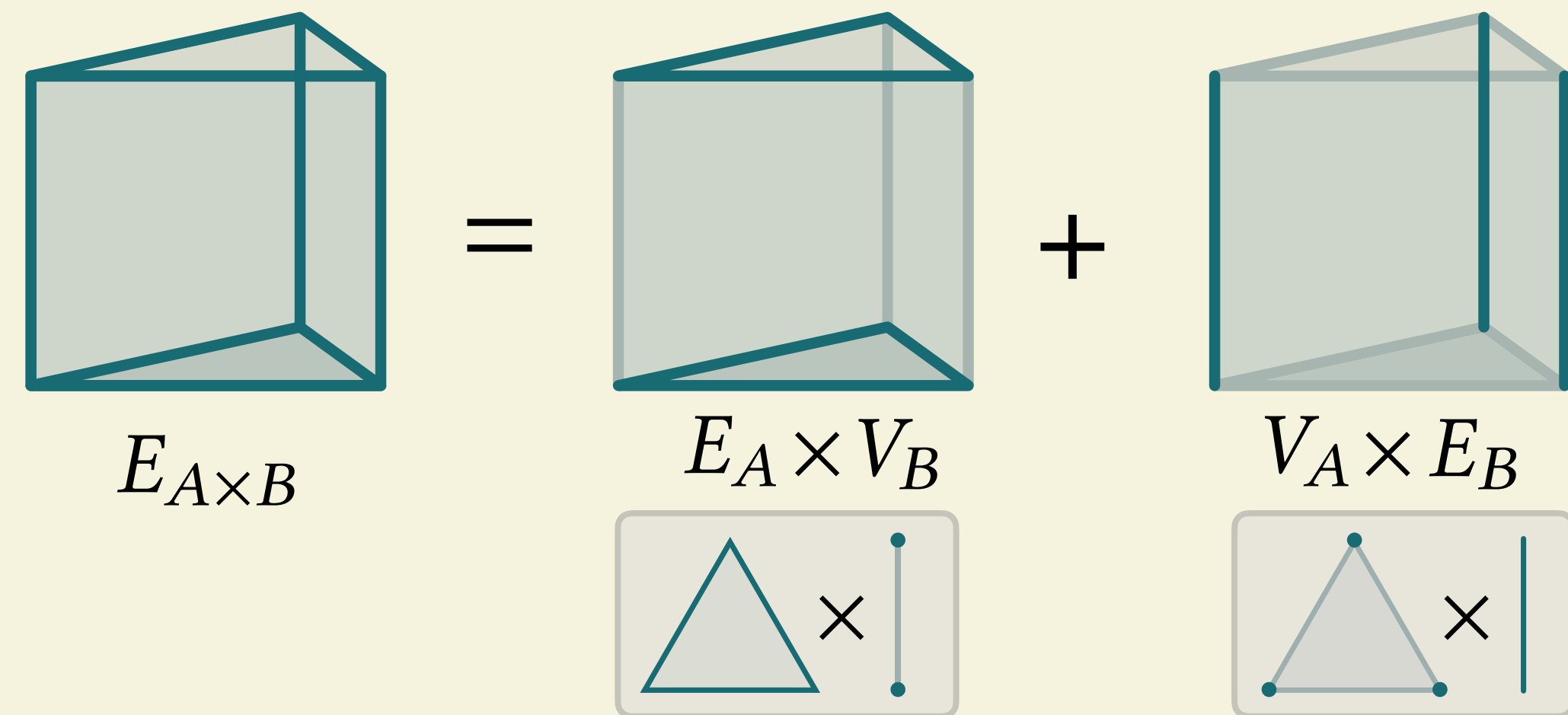
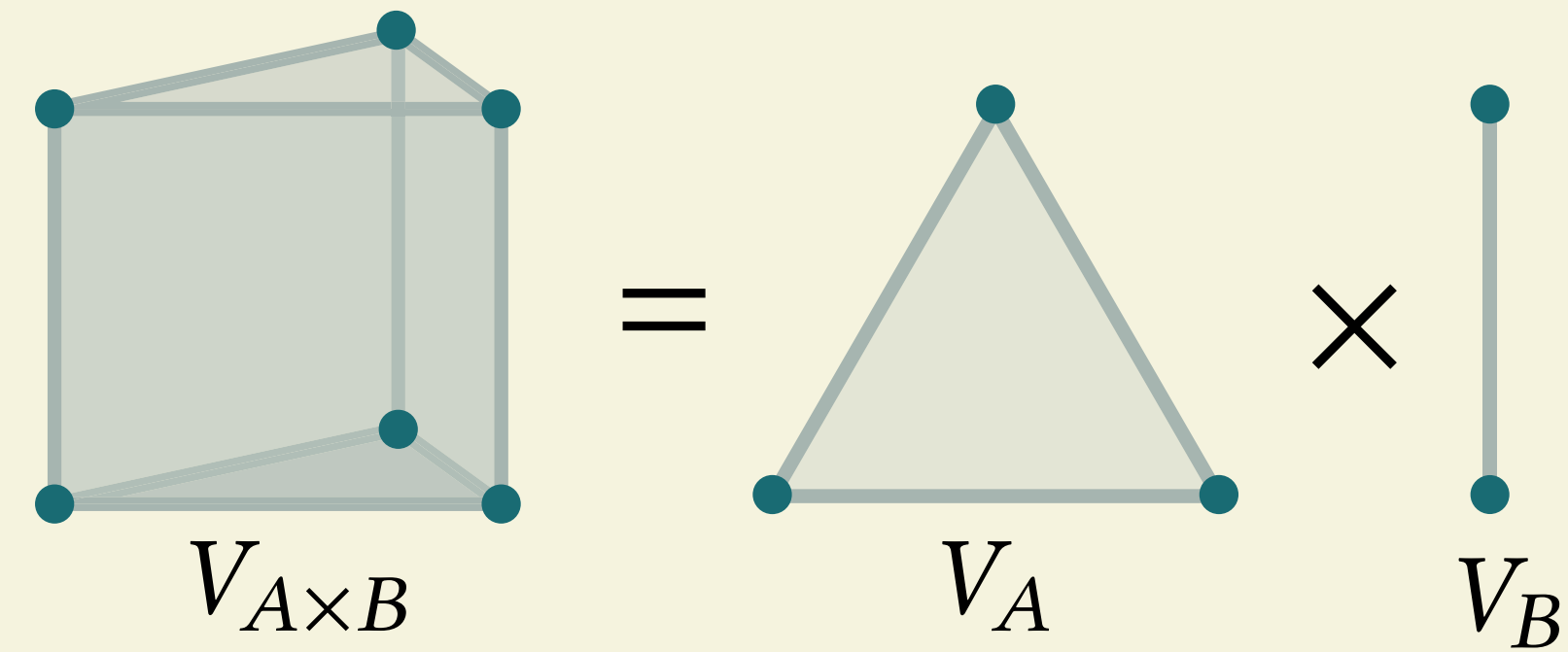
AT



Product Spaces

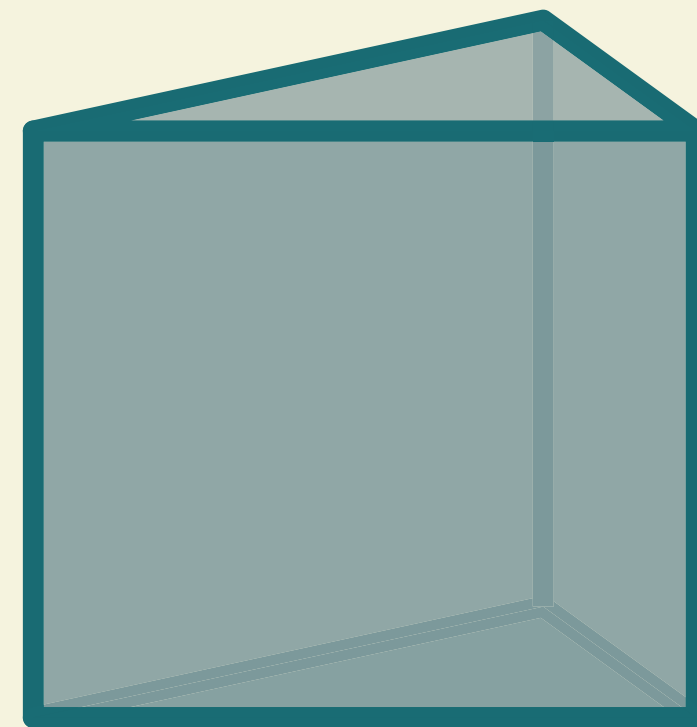


mesh structure



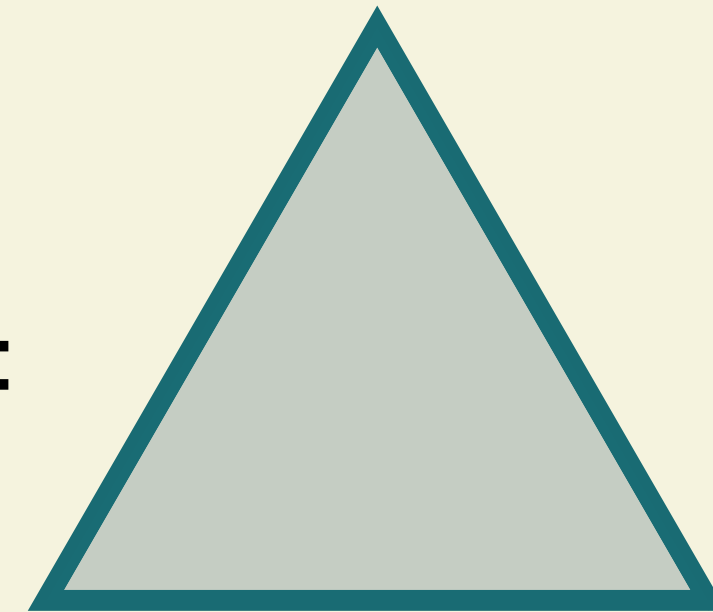
Product Spaces

product rule



$A \times B$

=

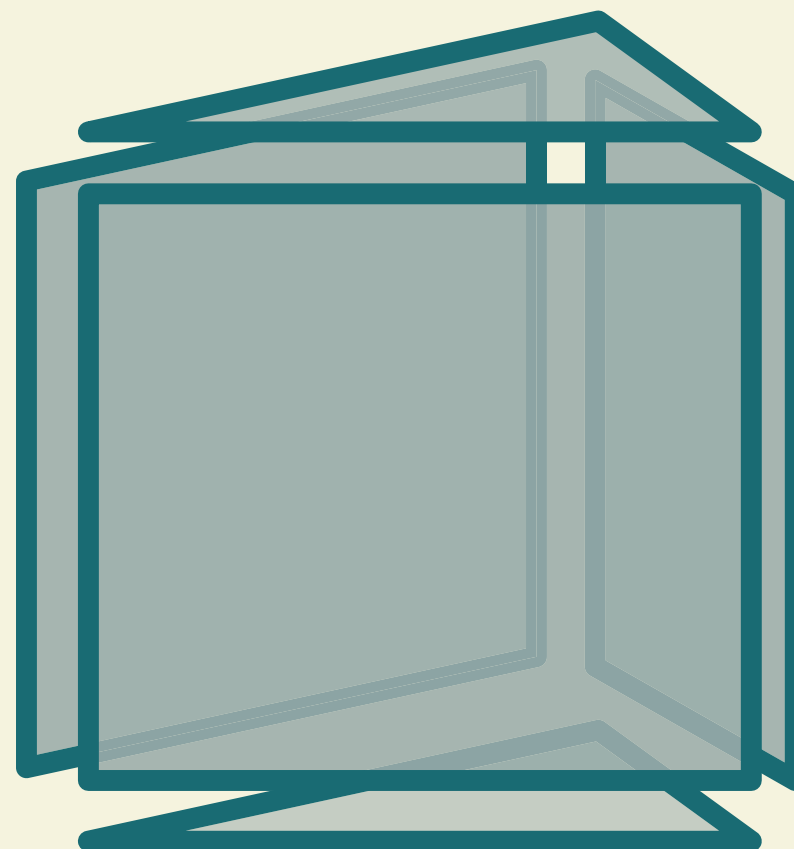


A

×

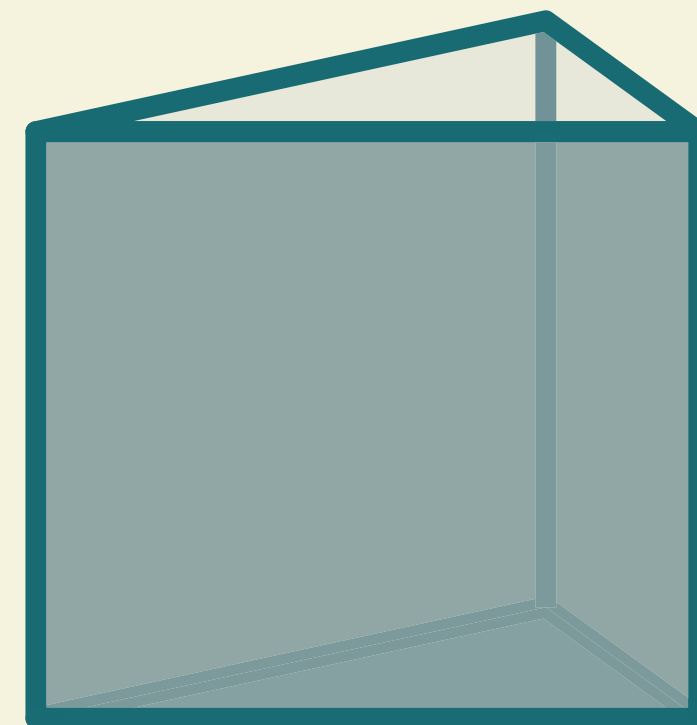


B



$\partial(A \times B)$

=



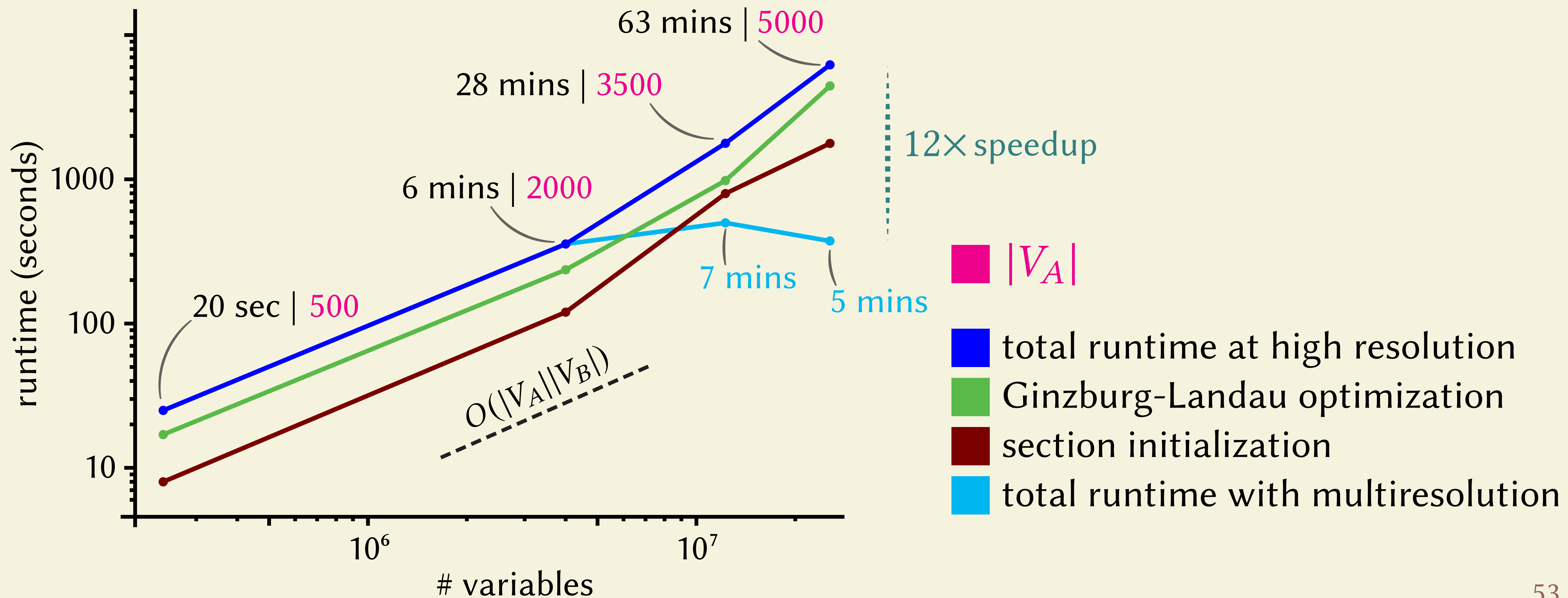
$(\partial A) \times B$

+

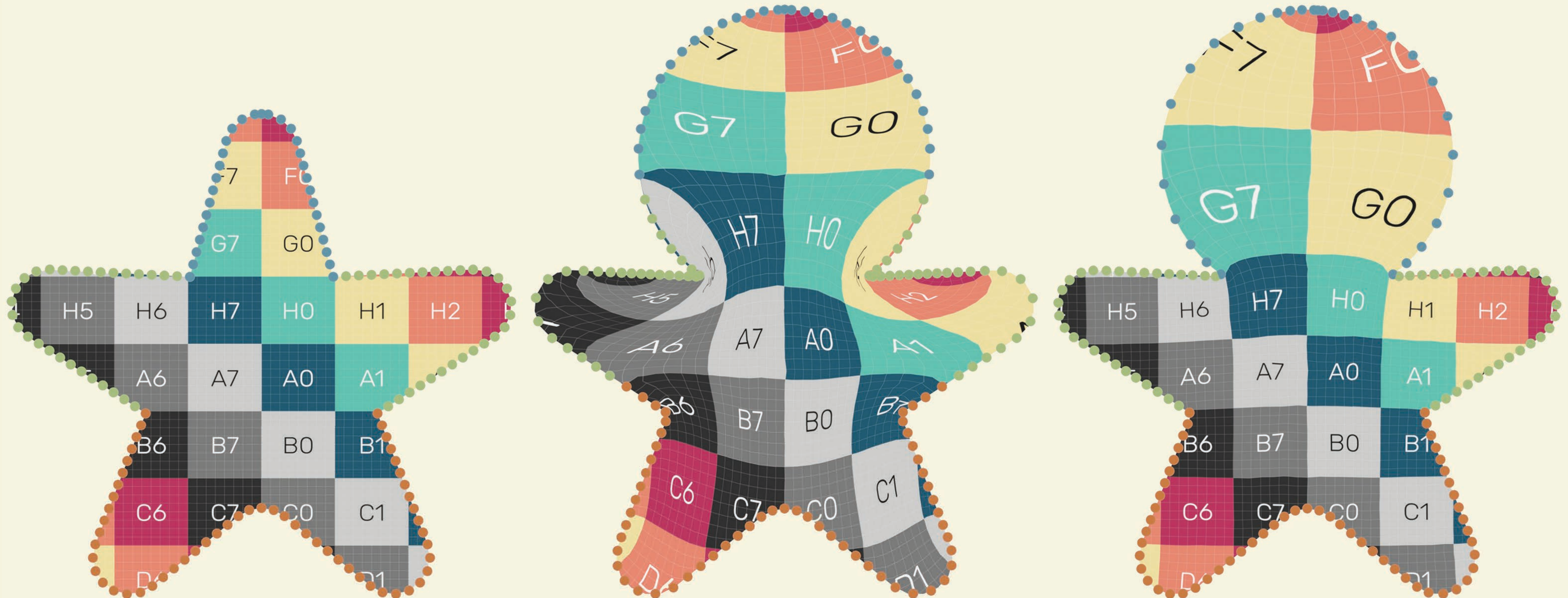


$A \times (\partial B)$

Runtime



The Importance of Curve Constraints



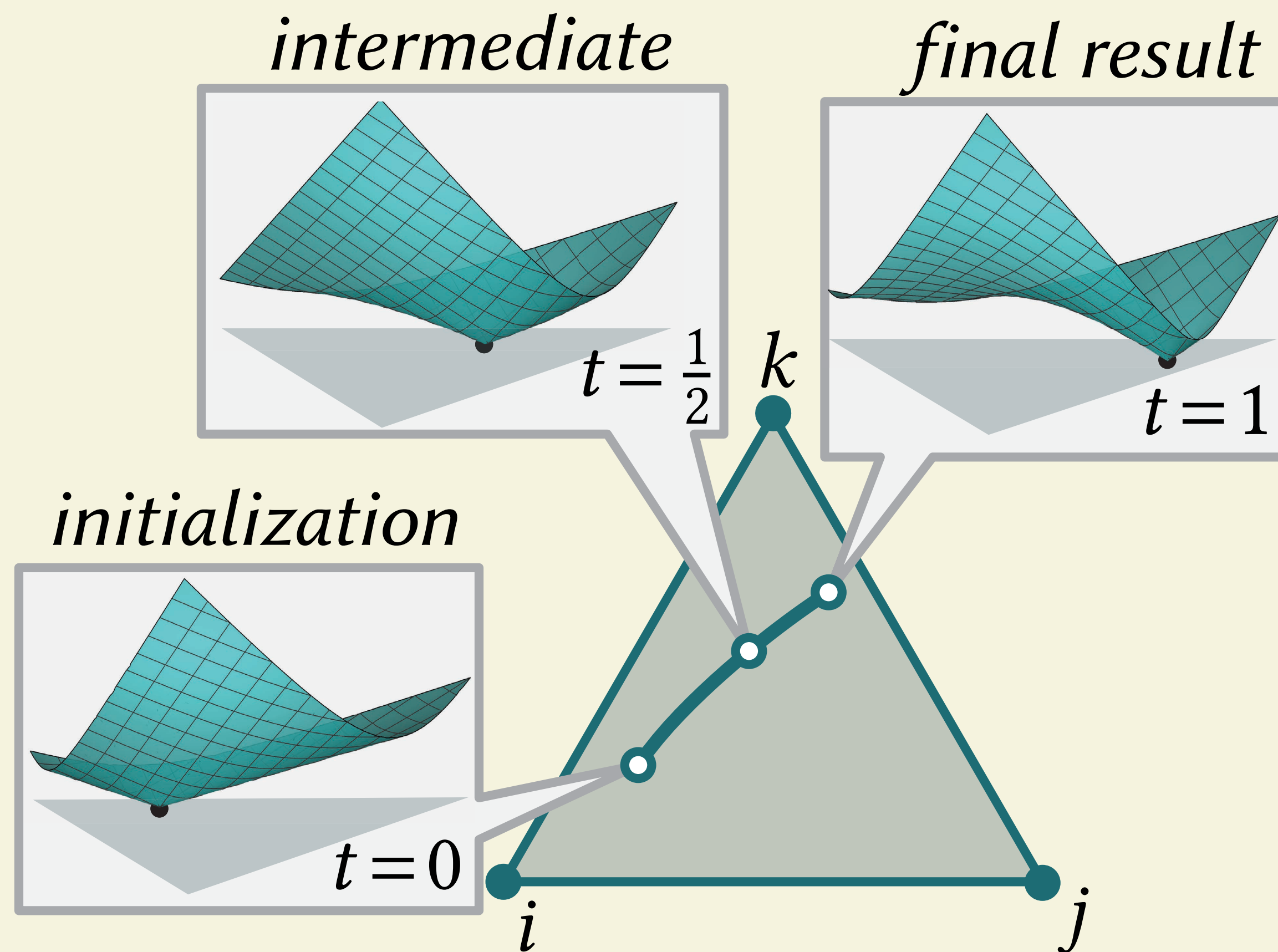
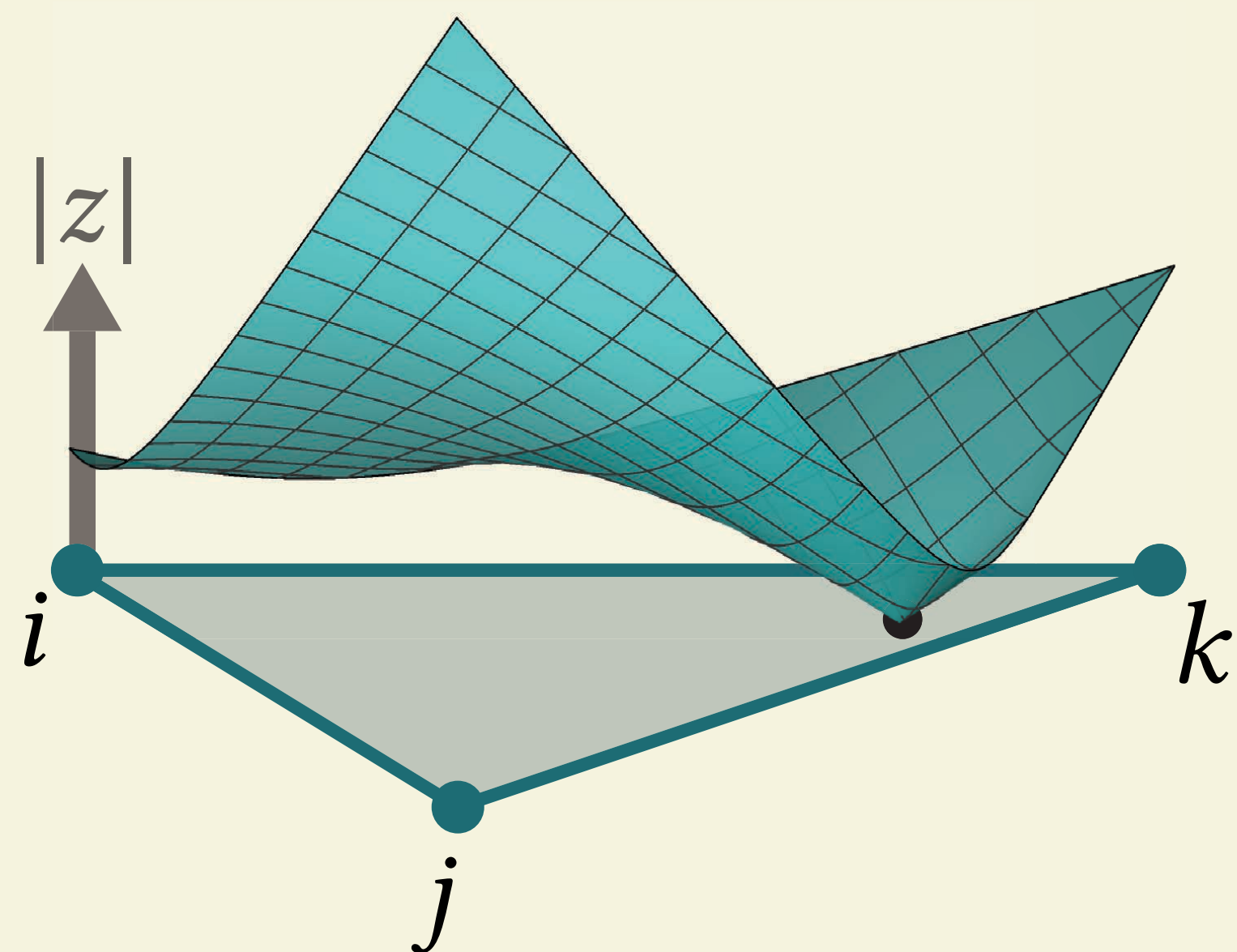
point constraints

curve constraints

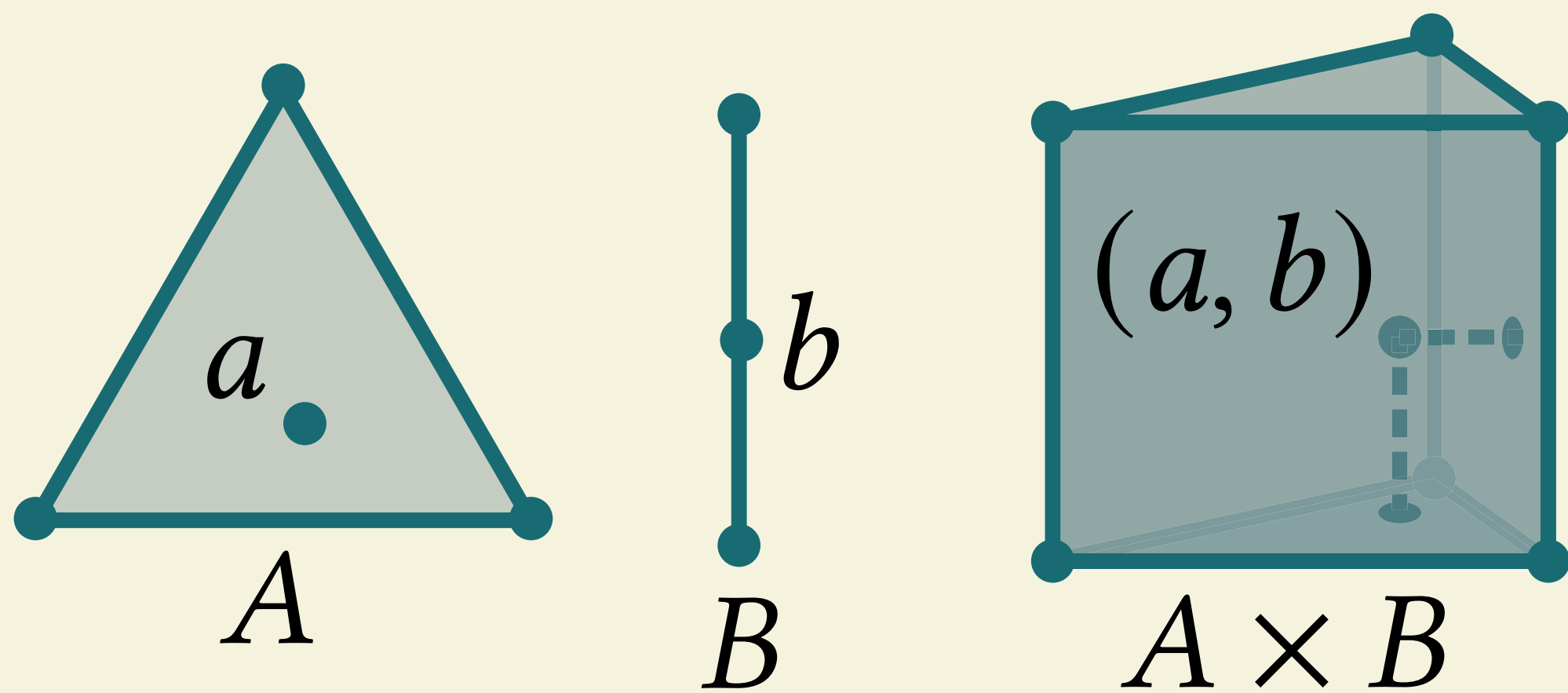
Finding Singularities

$$(1 - b_j - b_k)|z_i| + b_j|z_j|e^{l(b_k\Omega + \omega_{ij})} + b_k|z_k|e^{-l(b_j\Omega + \omega_{ki})} = 0.$$

$$\omega_{ij} := \arg\left(\frac{z_j}{r_{ij}z_i}\right) \in [-\pi, \pi).$$

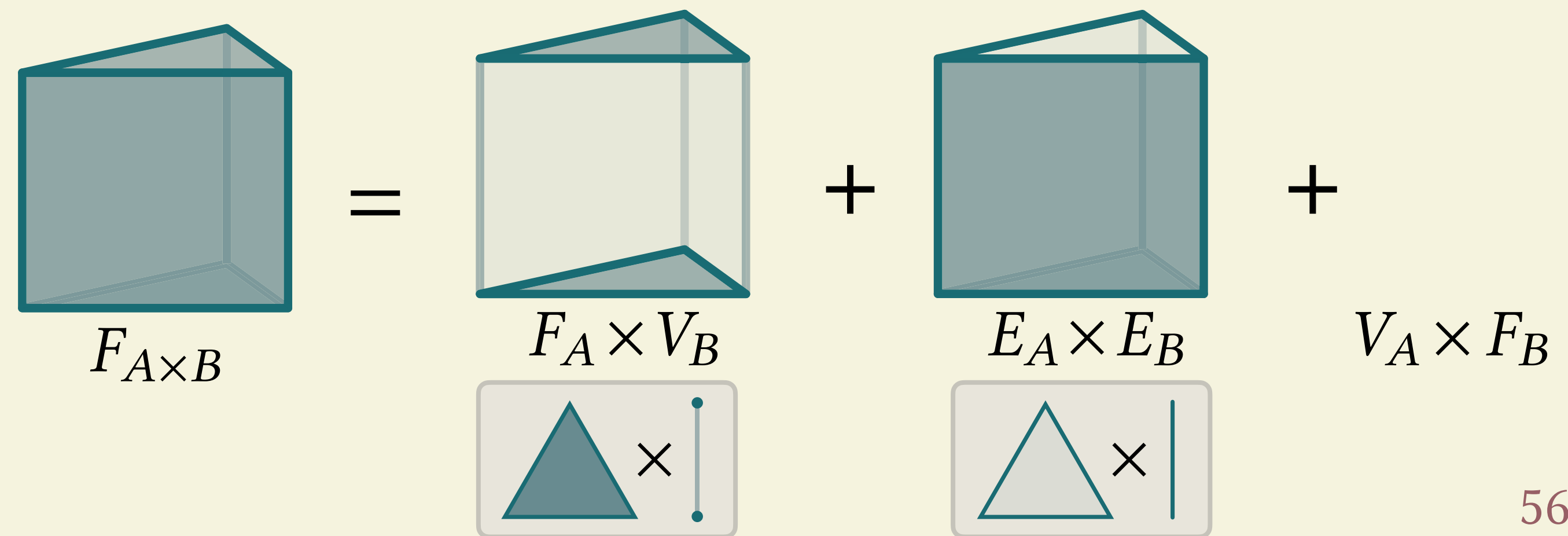
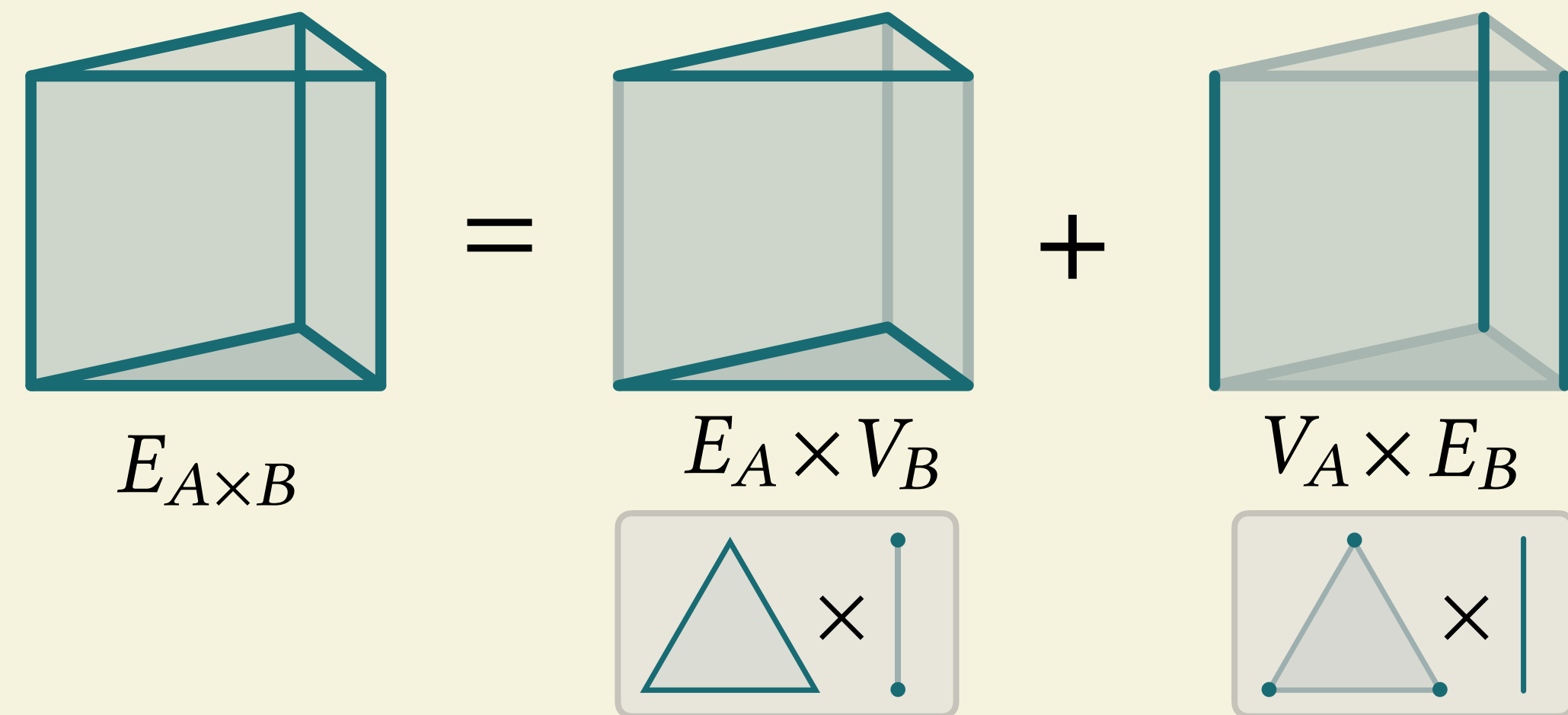
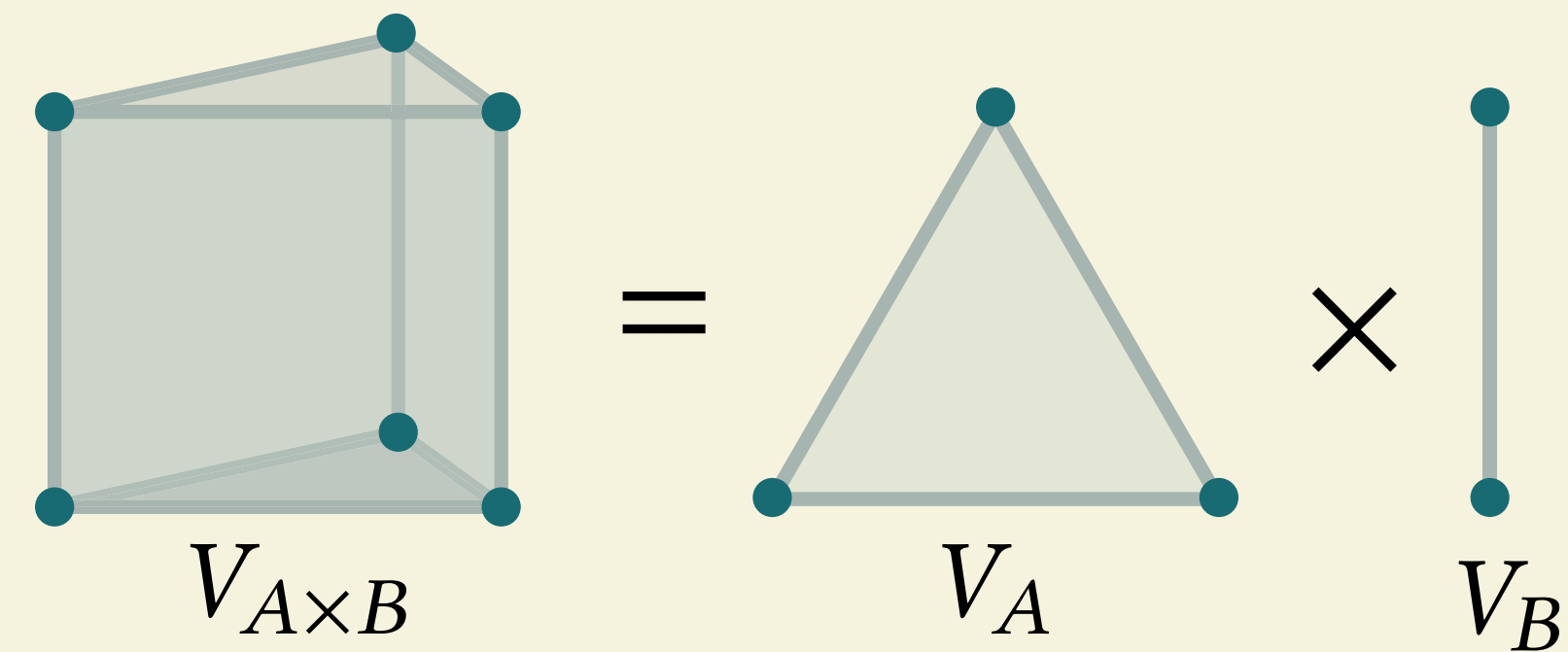


Product Spaces



extrudes A
along B

mesh structure



Initialization

4.4 Initialization

Our method can start from a (potentially noisy) input function given as vertex-face maps $\varphi : V_A \rightarrow F_B$ and $\psi : V_B \rightarrow F_A$. To use these in our algorithm, we must convert them into an initial complex section $z : A \times B \rightarrow \mathbb{C}$ whose zero set approximates the graphs of φ and ψ .

Strategy. We leverage the principle that low-energy states of a connection Laplacian concentrate their zeros in regions of high curvature (Section 3.1). To do so, we construct a specific connection $r^{\varphi, \psi}$ on $A \times B$ that concentrates all its curvature exactly along the graphs of φ and ψ . We then define our initial section z as the smallest eigenvector of the associated Laplacian. Because this bundle lies in the same homology class as the one constructed in Section 4.1, this section serves as a valid initialization for the following Ginzburg-Landau minimization.

Construction. We build $r^{\varphi, \psi}$ slice-by-slice using the trivial connections algorithm from Section 3.4.2. For each vertex $i_A \in V_A$, we compute a connection r^{i_A} on B that concentrates 2π curvature into the single face $\varphi(i_A)$. Then, for every edge of the form $i_A \times e_B$ in the product space, we assign $r_{i_A \times e_B}^{\varphi, \psi} = r_{e_B}^{i_A}$. Symmetrically, for each $i_B \in V_B$, we run trivial connection on A , concentrating curvature in face $\psi(i_B)$ and assign this connection to product space edges of the form $e_A \times i_B$. This results in a fully defined connection on the product space whose curvature is concentrated to the graph of the input maps. The zero set of the smallest eigenvector of the associated Laplacian acts as a good representation of the input correspondences (see Figure 17).

Laplacian. The connection constructed above does not factor as the product of independent connections on A and B , which would normally require assembling the full four-dimensional finite element Laplacian. Instead, we consider a simpler discretization of the Dirichlet energy using a lumped discretization for the outer integrals of Equation 6 while retaining a finite-element discretization for the inner integrals over the 2D slices. Explicitly, for every vertex $i_A \in V_A$, let L_B^{∇, i_A} denote the finite element Laplacian of r^{i_A} on B , and let $z^{(i_A)}$ be the restriction of section Z to the slice. Using similar notation for the slices fixed at i_B , we can express the Dirichlet energy of the connection $r^{\varphi, \psi}$ as:

$$\begin{aligned} \mathcal{E}_{\varphi, \psi}^D(Z) = & \sum_{v \in V_A} (M_A)_{v,v} \langle L_B^{\nabla, v} z^{(v)}, z^{(v)} \rangle_{\mathbb{C}} \\ & + \sum_{v \in V_B} (M_B)_{v,v} \langle L_A^{\nabla, v} z^{(v)}, z^{(v)} \rangle_{\mathbb{C}}, \end{aligned} \quad (34)$$

and so the blocks in the product-space connection Laplacian are given by the surface connection Laplacians on the individual slices weighted by the dual area of the vertex on the complementary mesh.

The smallest eigenvalue can then be computed via an iterative matrix-free solver such as LOBPCG [Knyazev et al. 2007].