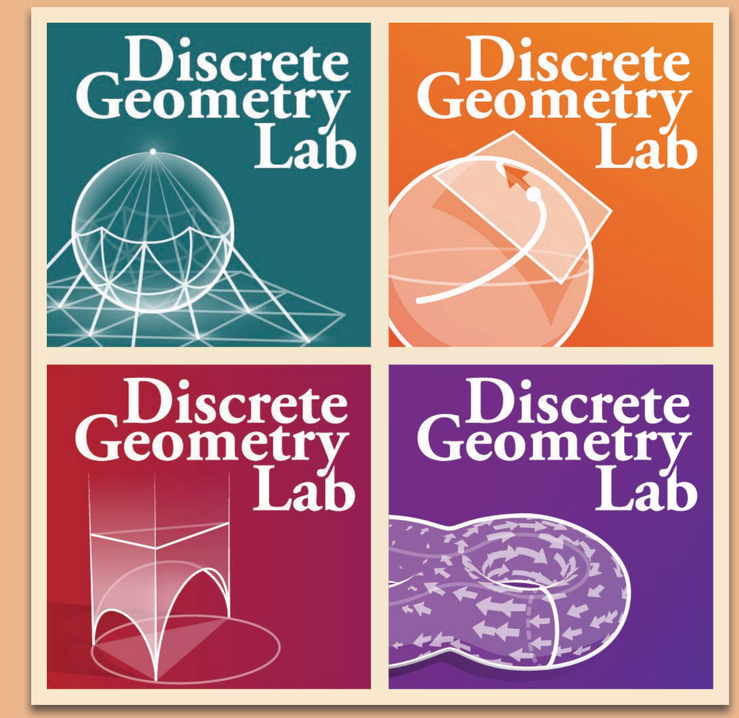
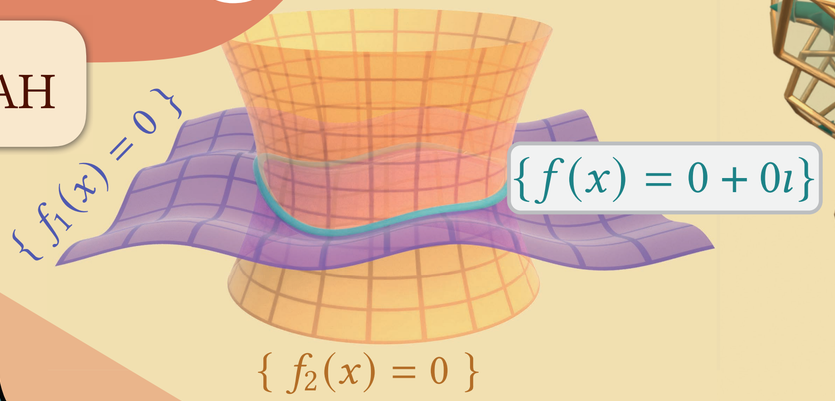
Denison University Ronneberg Lecture

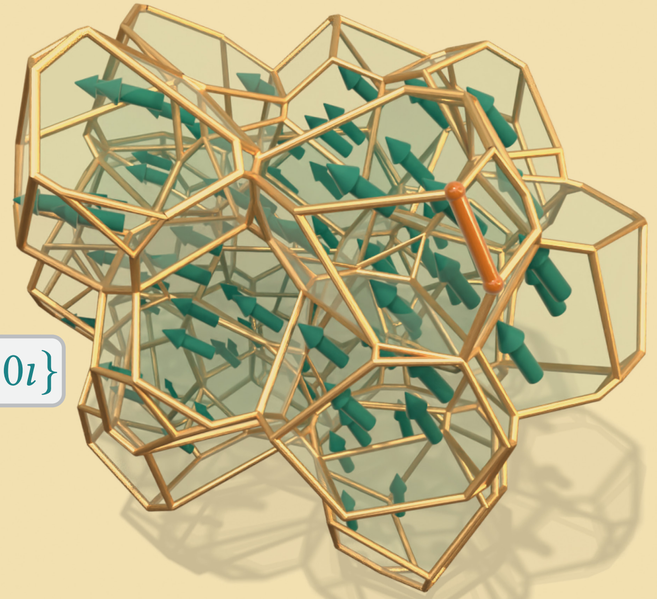
Thursday November 5, 2026

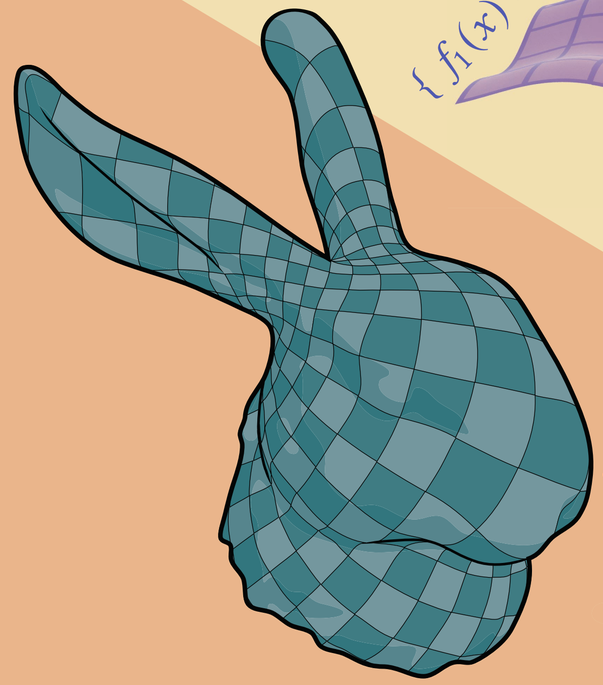
# An Introduction to Geometry Processing

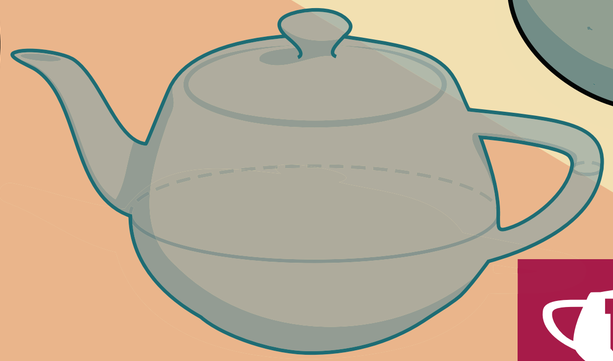
Mark Gillespie, University of Utah

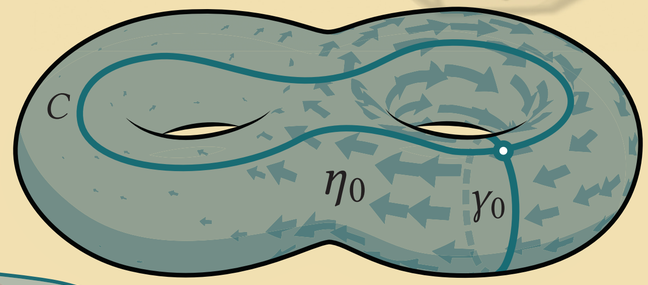




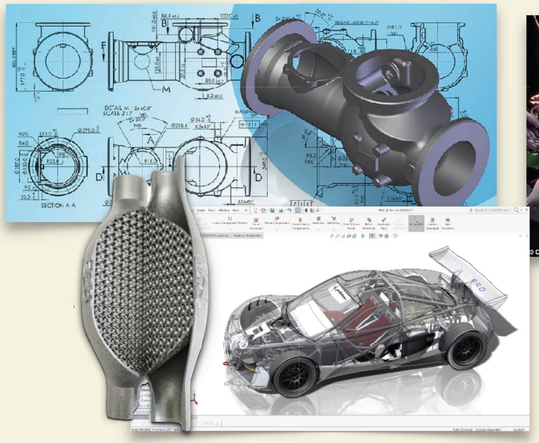




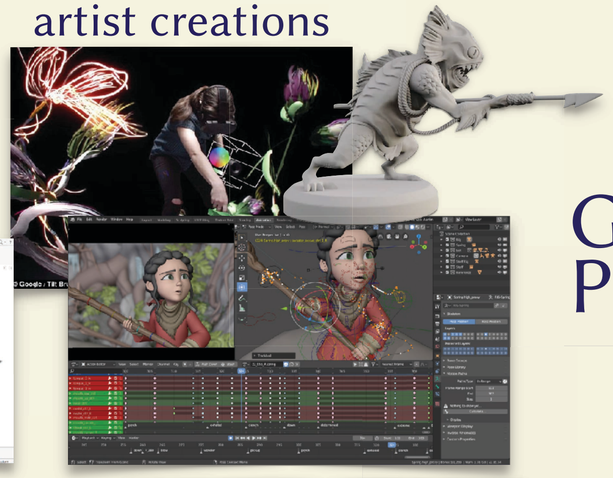

Computer Graphics
THE UNIVERSITY OF UTAH

# Geometry is all around us

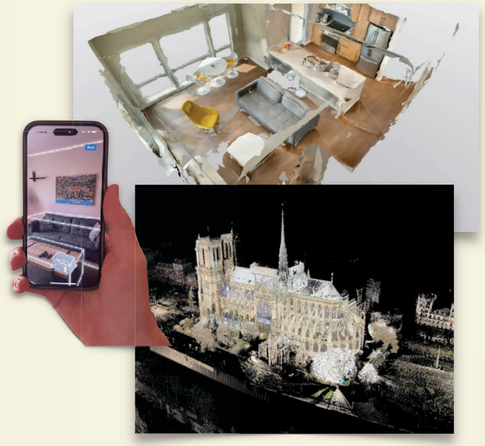
technical assets



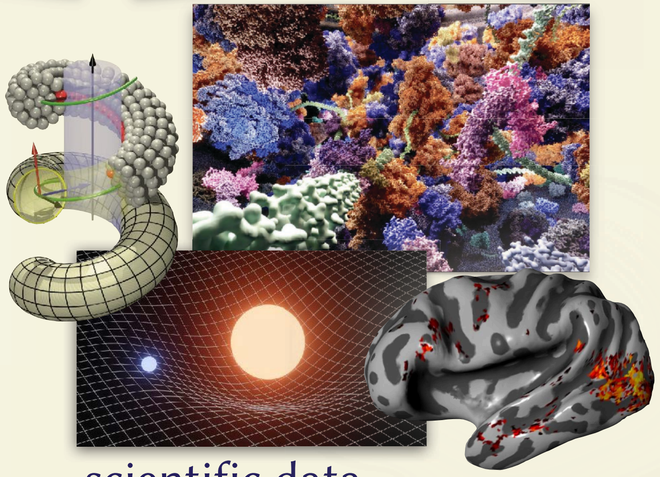
artist creations



3D scans



scientific data

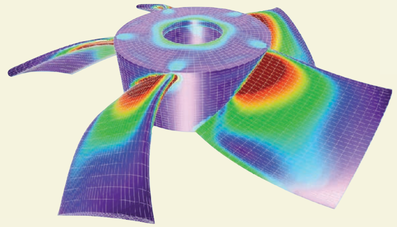


Geometry Processing

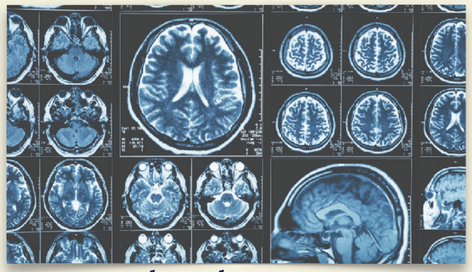
animation



simulation



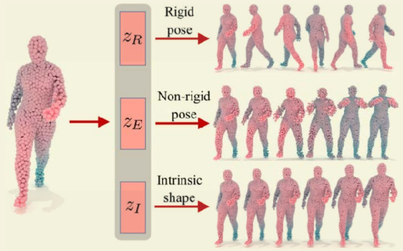
medical imaging



robotics



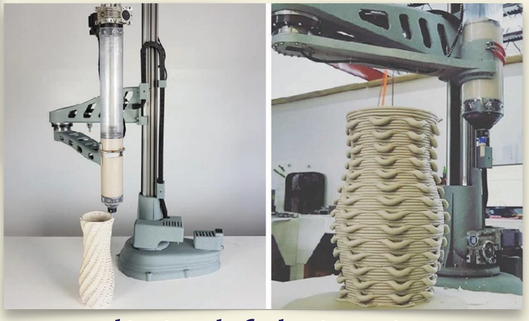
geometric ML

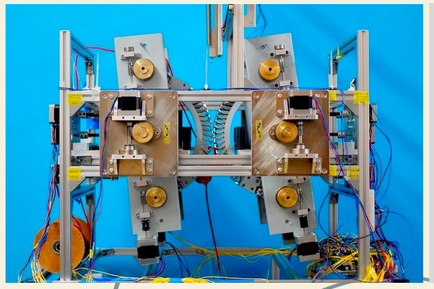



creative tools

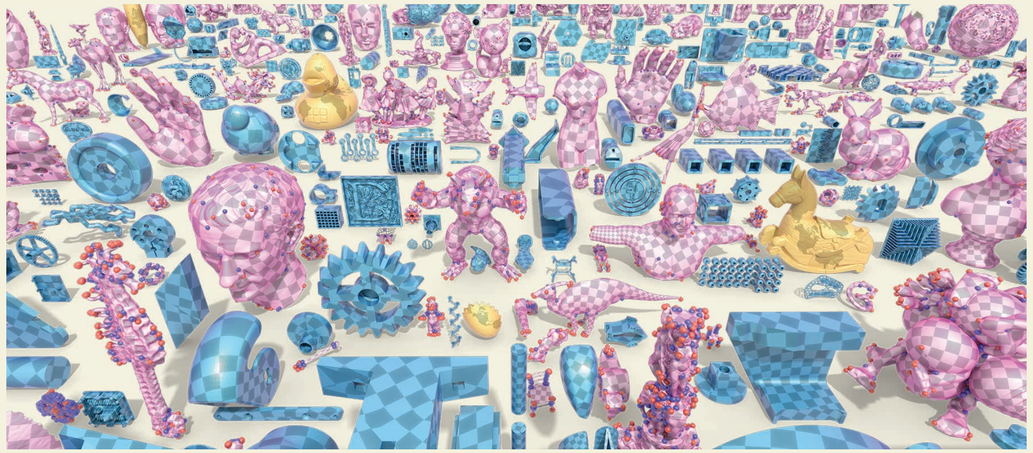


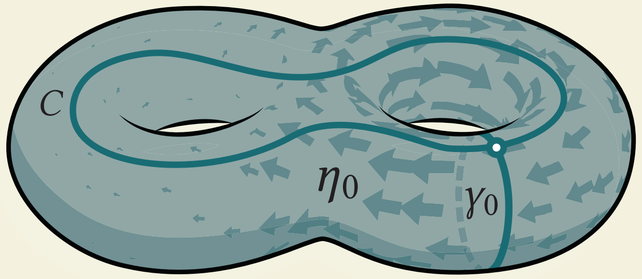
digital fabrication

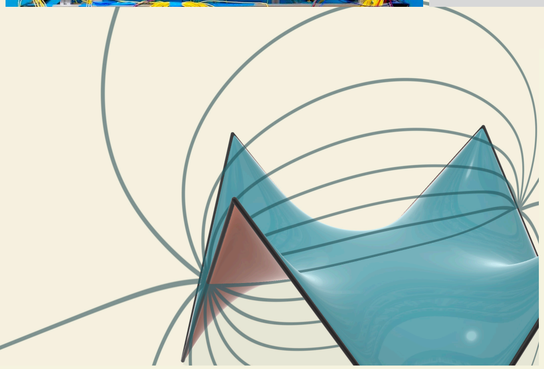




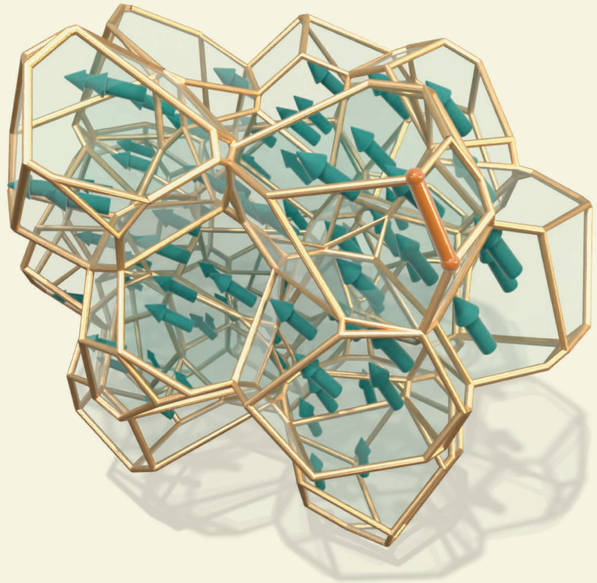


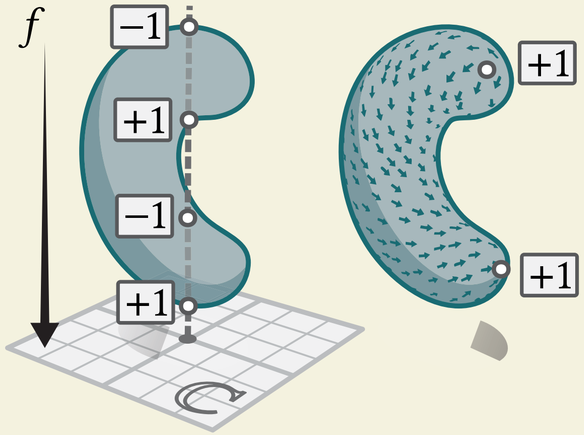




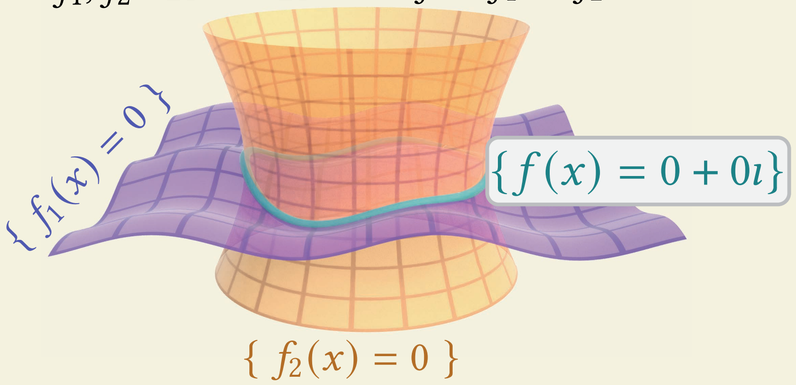



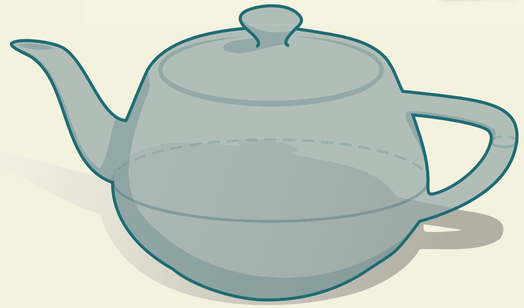
Thinking deeply about the **geometry** yields
**more efficient** algorithms
and **higher-quality** outputs





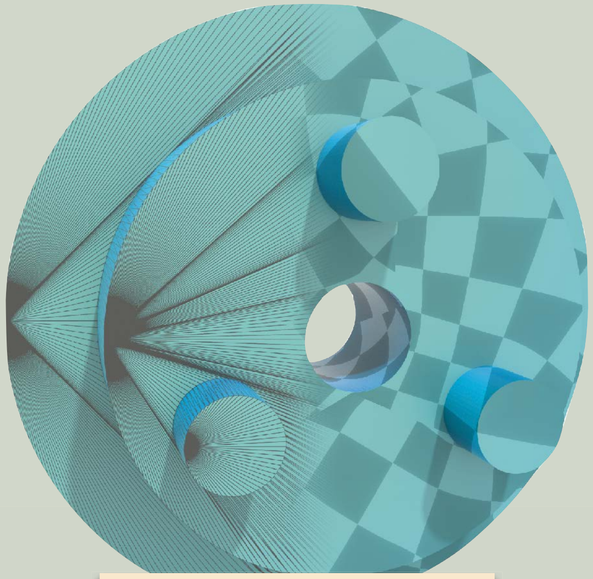

$f_1, f_2 : \mathbb{R}^3 \to \mathbb{R} \iff f = f_1 + \imath f_2 : \mathbb{R}^3 \to \mathbb{C}$



# Talk Outline

## I. Surface Parameterization



unwrap a surface onto the plane

Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. 2002. Least squares conformal maps for automatic texture atlas generation. ACM Trans. Graph. 21, 3. https://doi.org/10.1145/566654.566590

Patrick Mullen, Yiying Tong, Pierre Alliez, and Mathieu Desbrun. 2008. Spectral conformal parameterization. Symposium on Geometry Processing (SGP '08), 1487–1494.

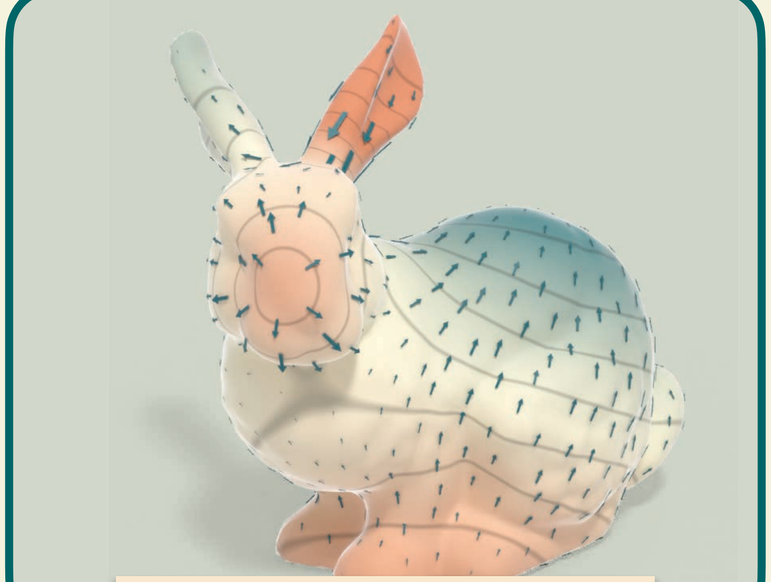
## II. Shape Correspondence



map from one shape to another

Etienne Corman, Yousuf Soliman, Robin Magnet, and Mark Gillespie. 2026. Implicit Minimal Surfaces for Bijective Correspondences. ACM Trans. Graph. 45, 4, Article 162. https://doi.org/10.1145/3811368

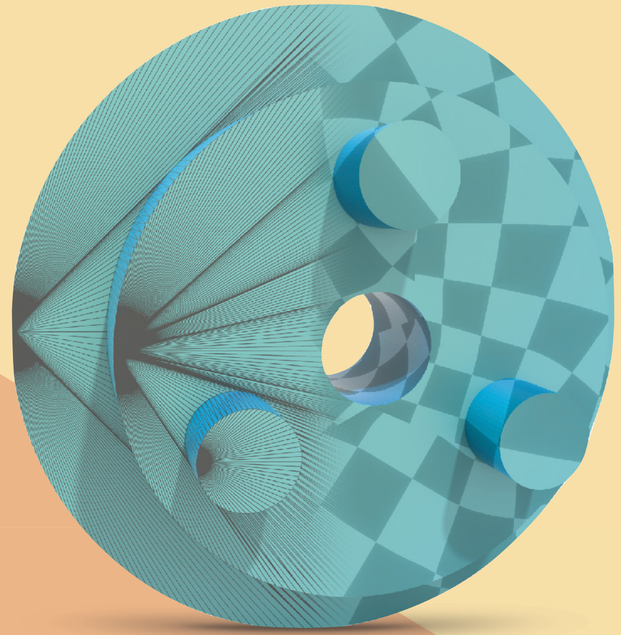
## III. Vector Field Design



find a smooth collection of directions on a surface

Keenan Crane, Fernando de Goes, Mathieu Desbrun, and Peter Schröder. 2013. Digital geometry processing with discrete exterior calculus. ACM SIGGRAPH 2013 Courses. Article 7, 1–126. https://doi.org/10.1145/2504435.2504442

# Surface Parameterization



# Parameterization

a map from a surface to the plane

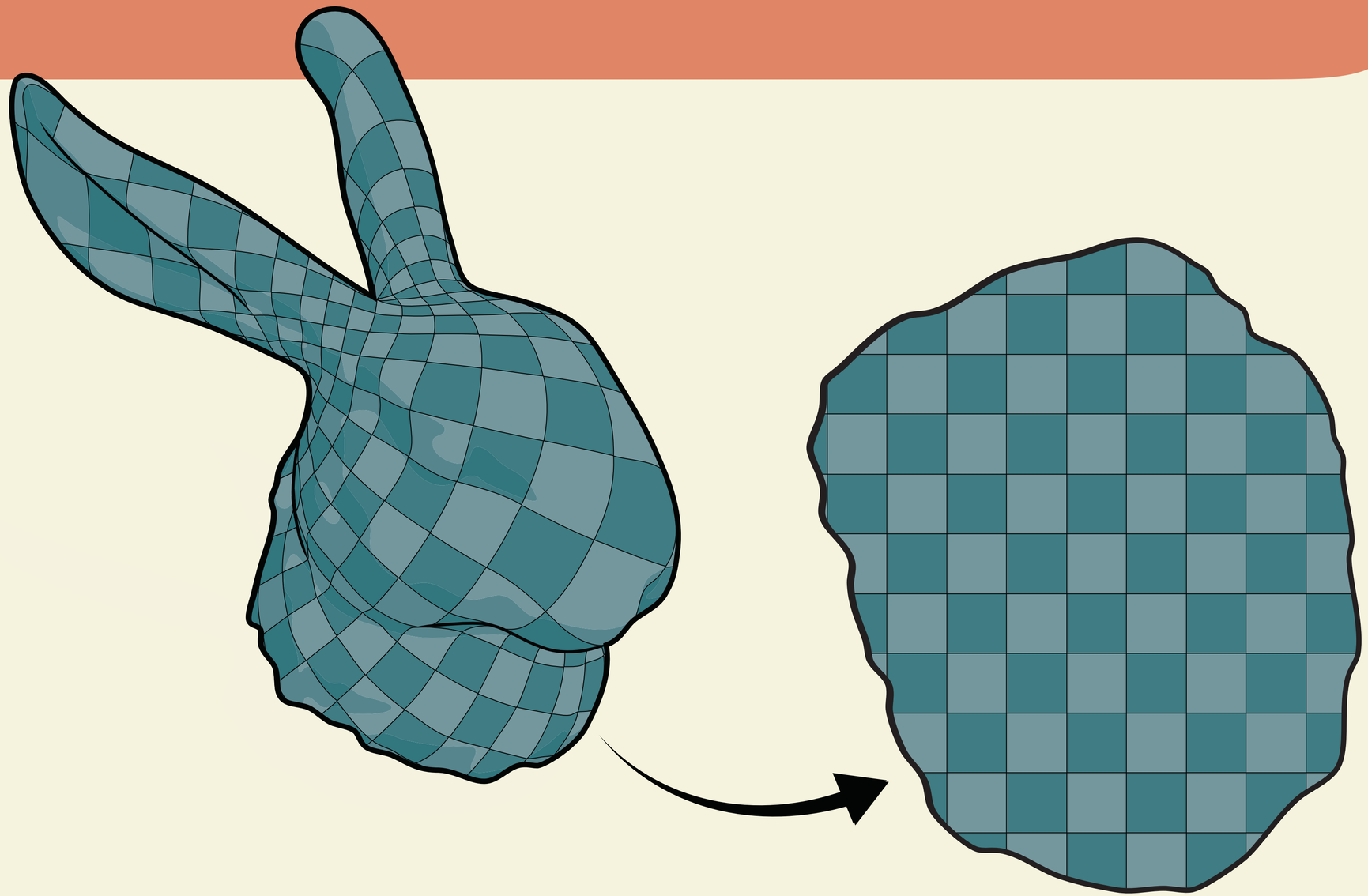


[ Crane *et al.* 2013 ]



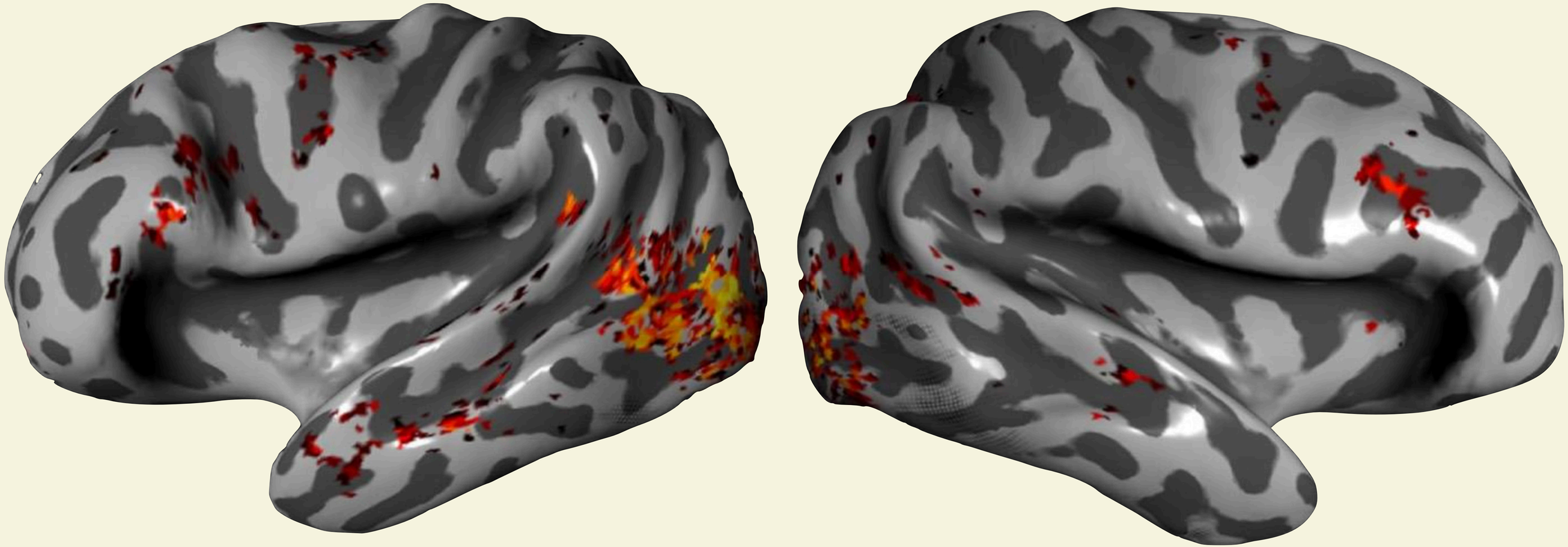
[ Kunimune 2018 ]

# Parameterization of general surfaces



# Applications of parameterization

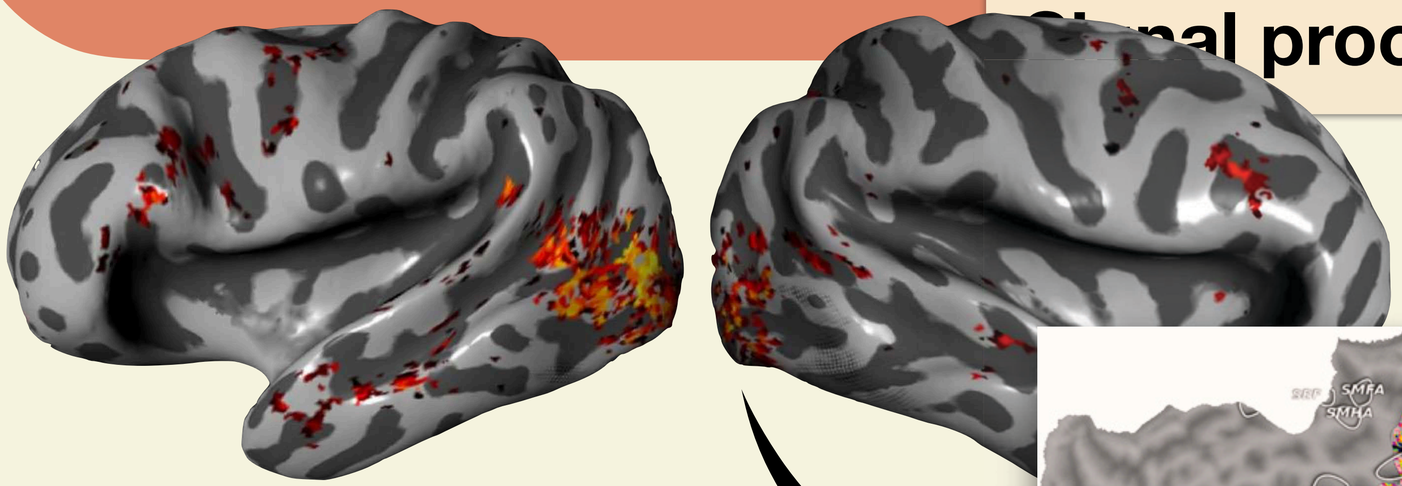
**Signal processing**

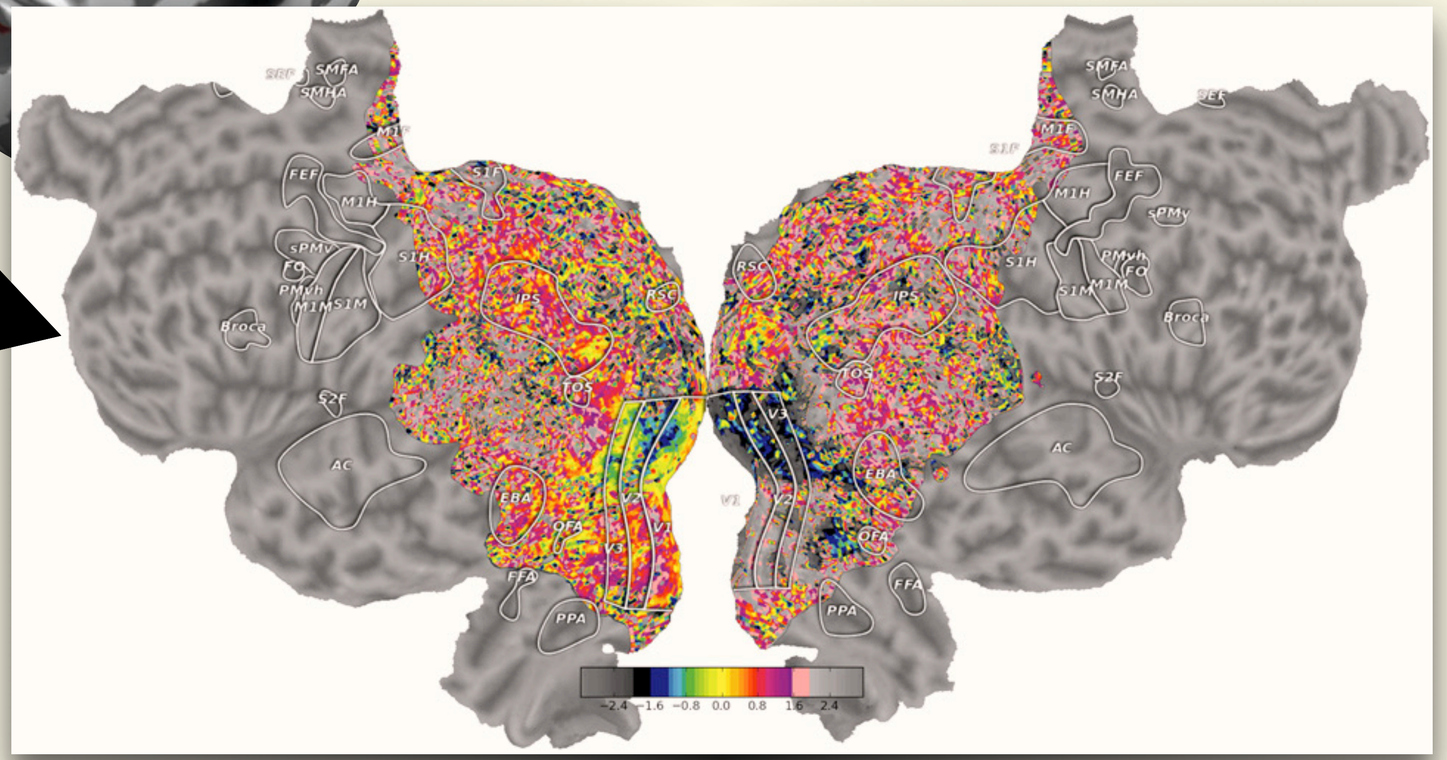


[ Gao *et al.* 2015 ]

# Applications of parameterization

## Signal processing

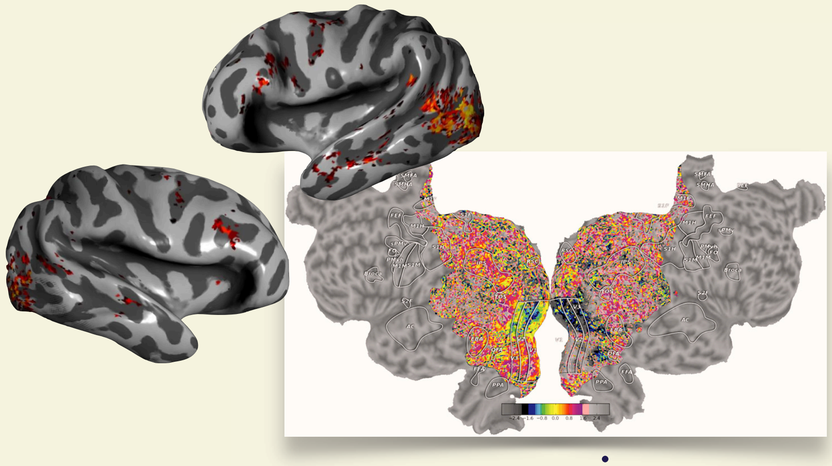





[ Gao *et al.* 2015 ]

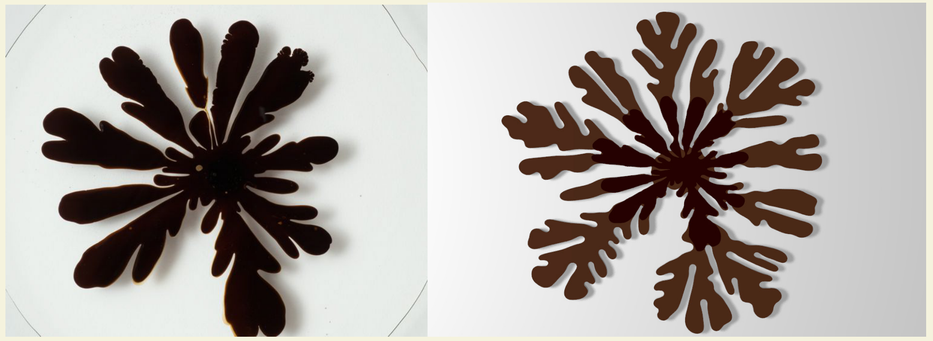
# Applications of parameterization

## Signal processing



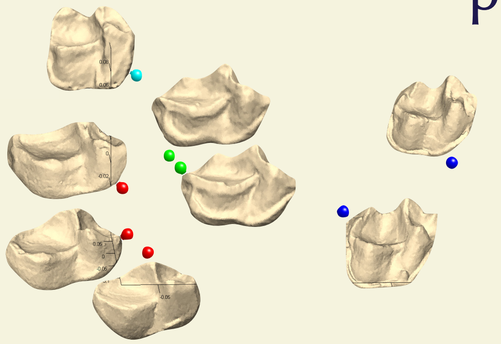
neuroscience

[ Gao *et al.* 2015 ]



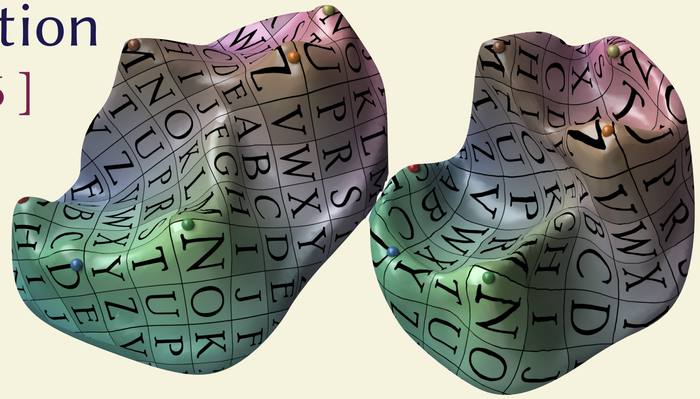
physical simulation

[ Segall *et al.* 2016 ]



shape analysis

[ Lipman & Daubechies 2011 ]



learning on surfaces

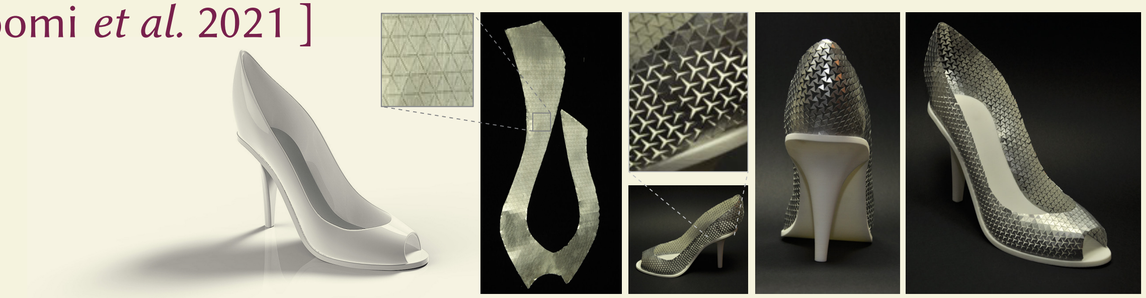
[ Maron *et al.* 2017 ]

# Applications of parameterization

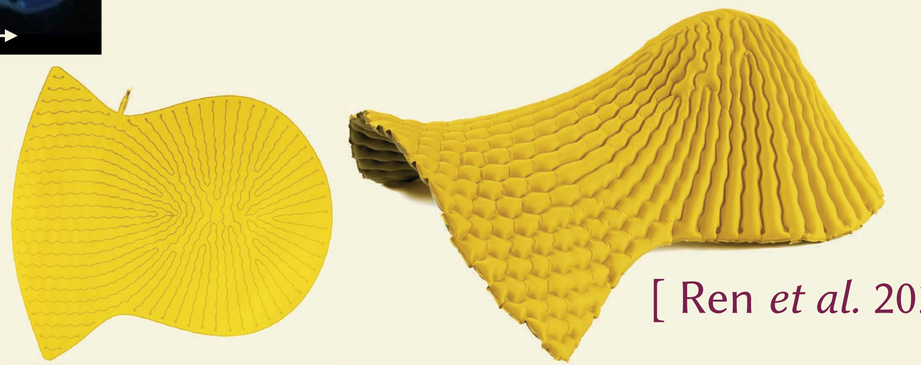
## Fabrication



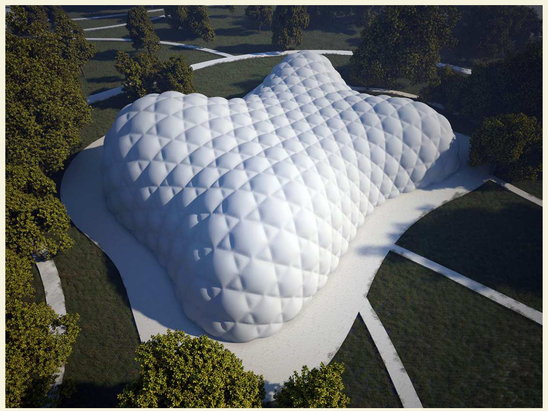

[ Nojoomi *et al.* 2021 ]



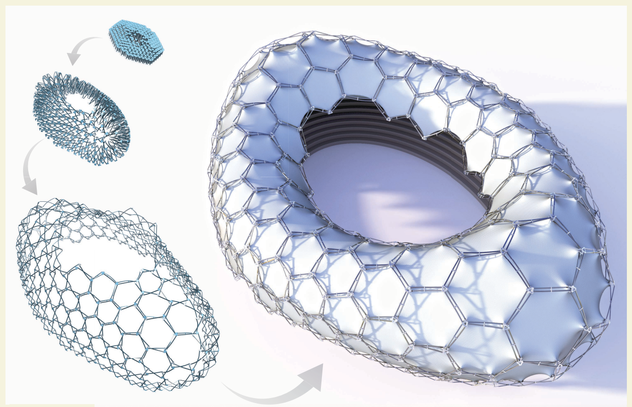
[ Konaković *et al.* 2016 ]



[ Ren *et al.* 2024 ]



[ Konaković-Luković *et al.* 2018 ]



[ Ren *et al.* 2022 ]

# Applications of parameterization

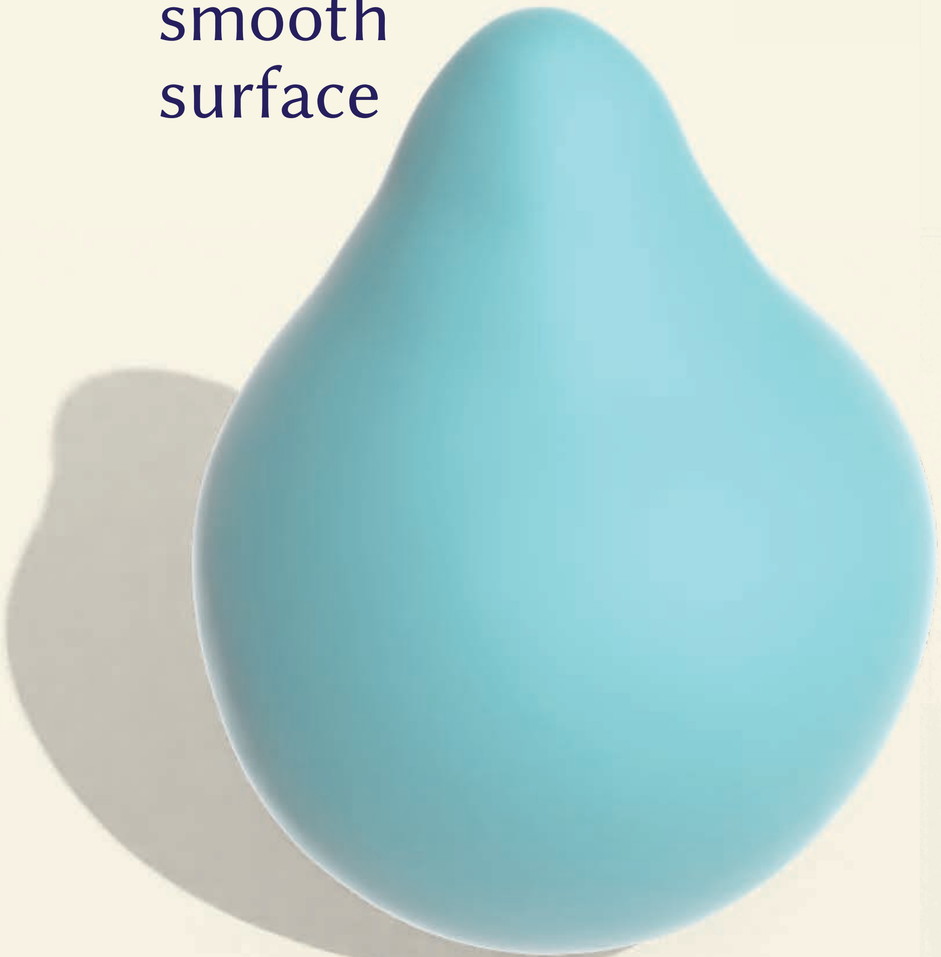
**Texturing 3d models**

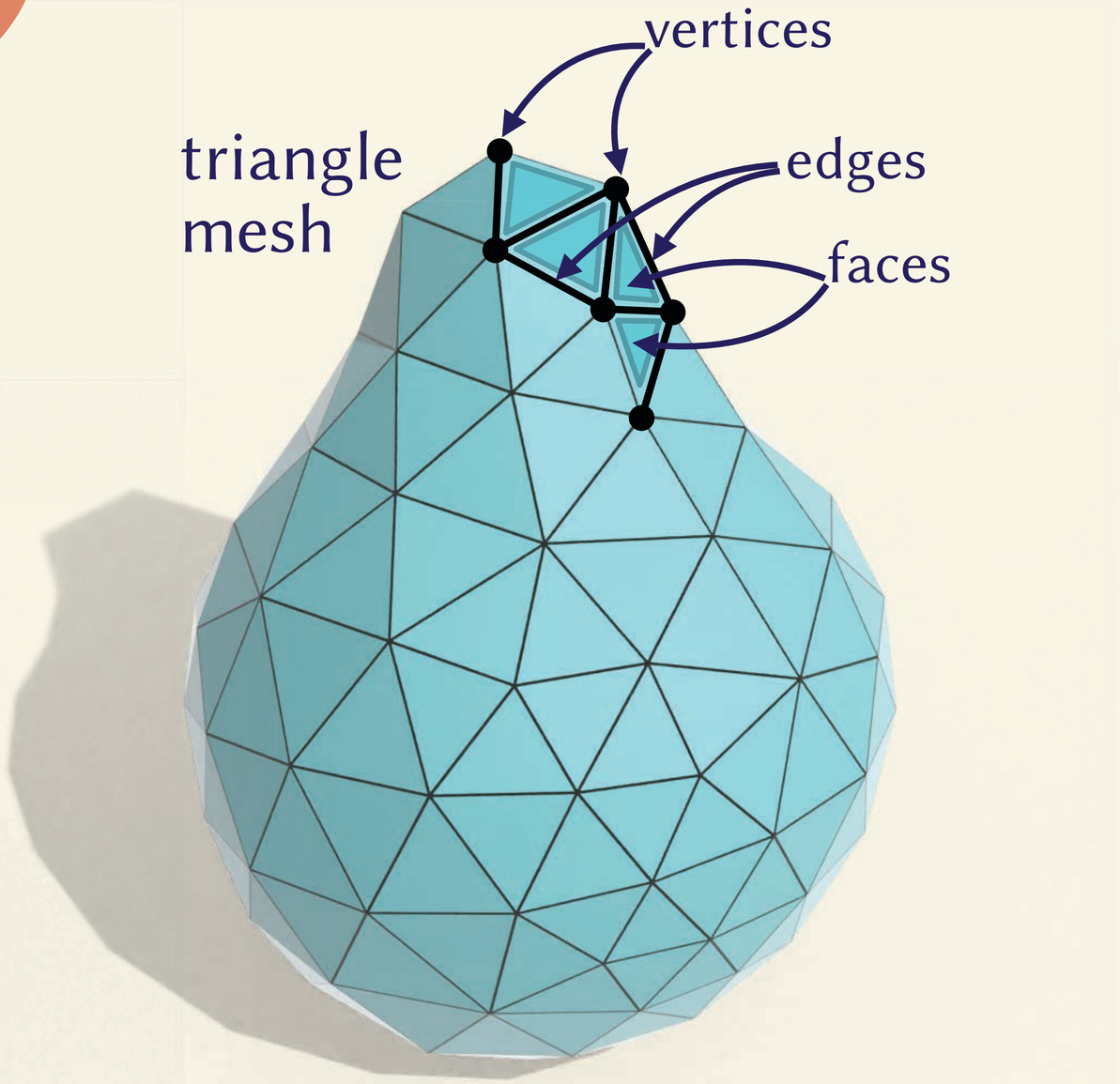


[ Timen 2012 ]

# Triangle meshes

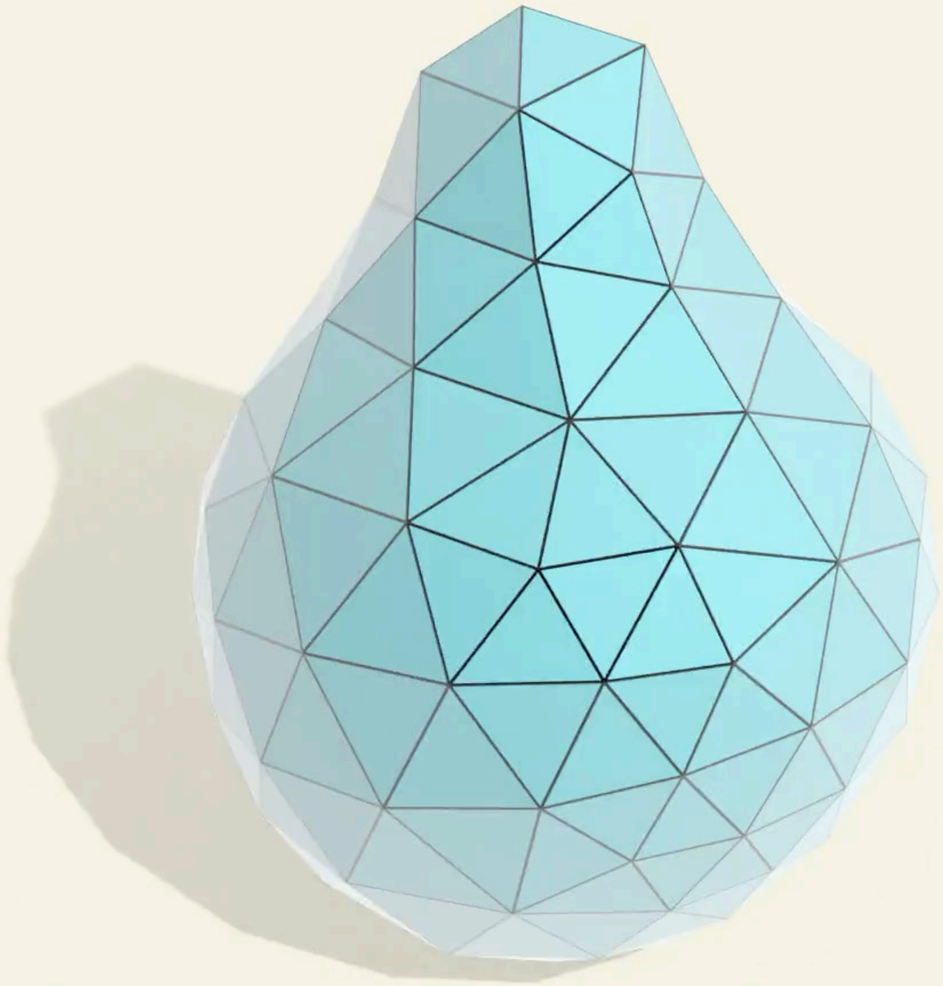
smooth
surface





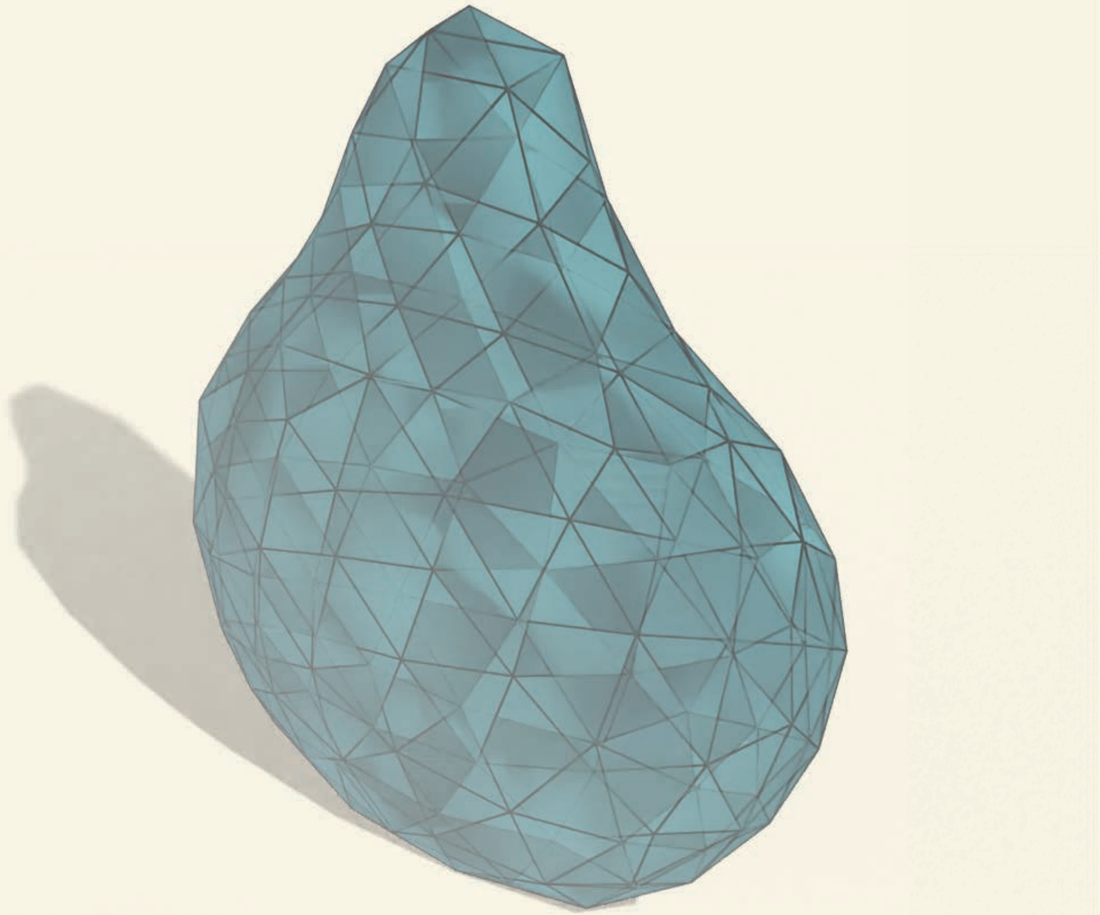
# Parameterizing triangle meshes

- Map every vertex into the plane
  - Collapse?



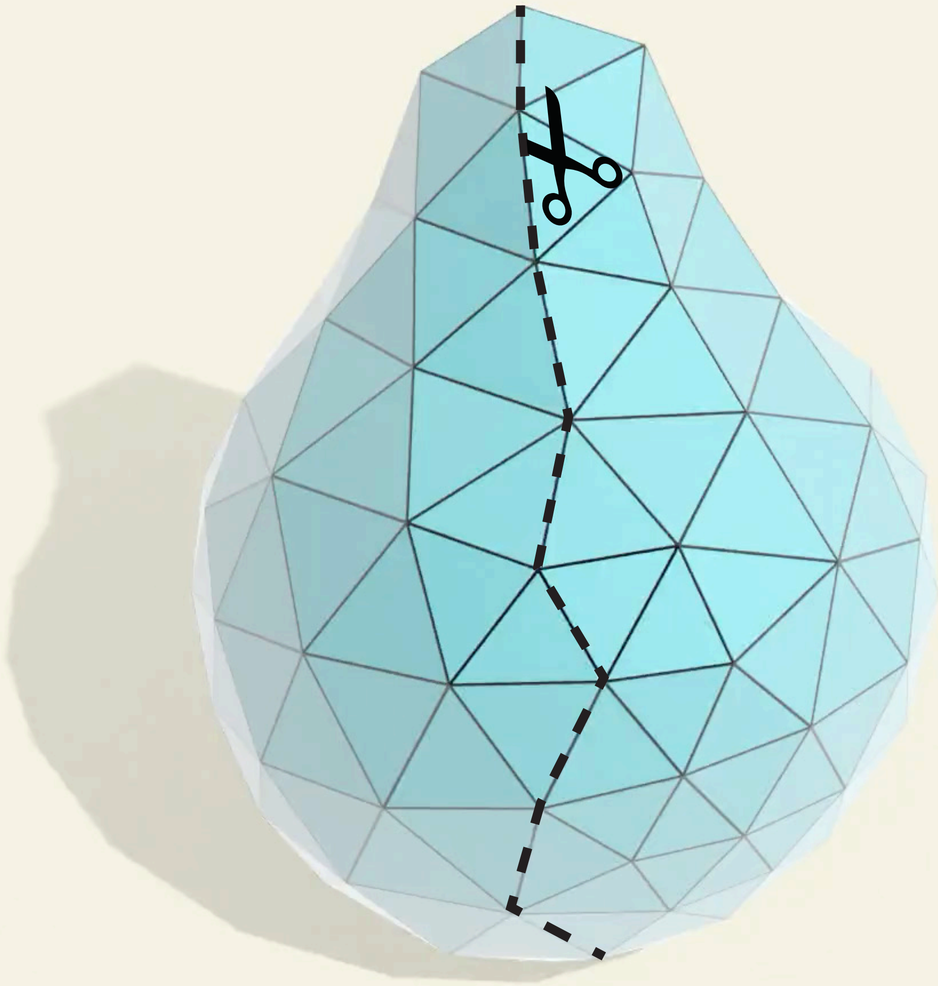
# Parameterizing triangle meshes

- Map every vertex into the plane
  - ~~Collapse?~~ Not injective

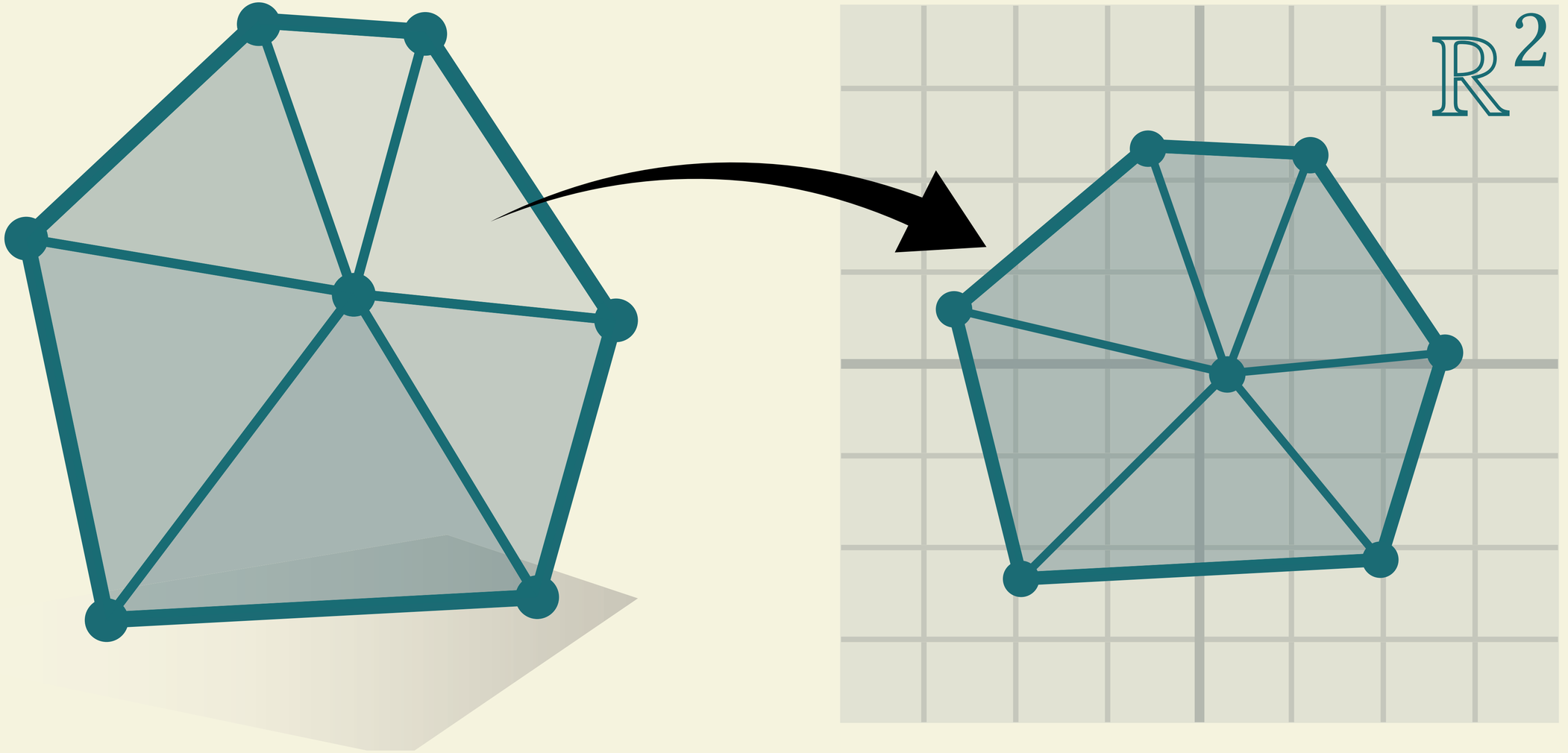


# Parameterizing triangle meshes

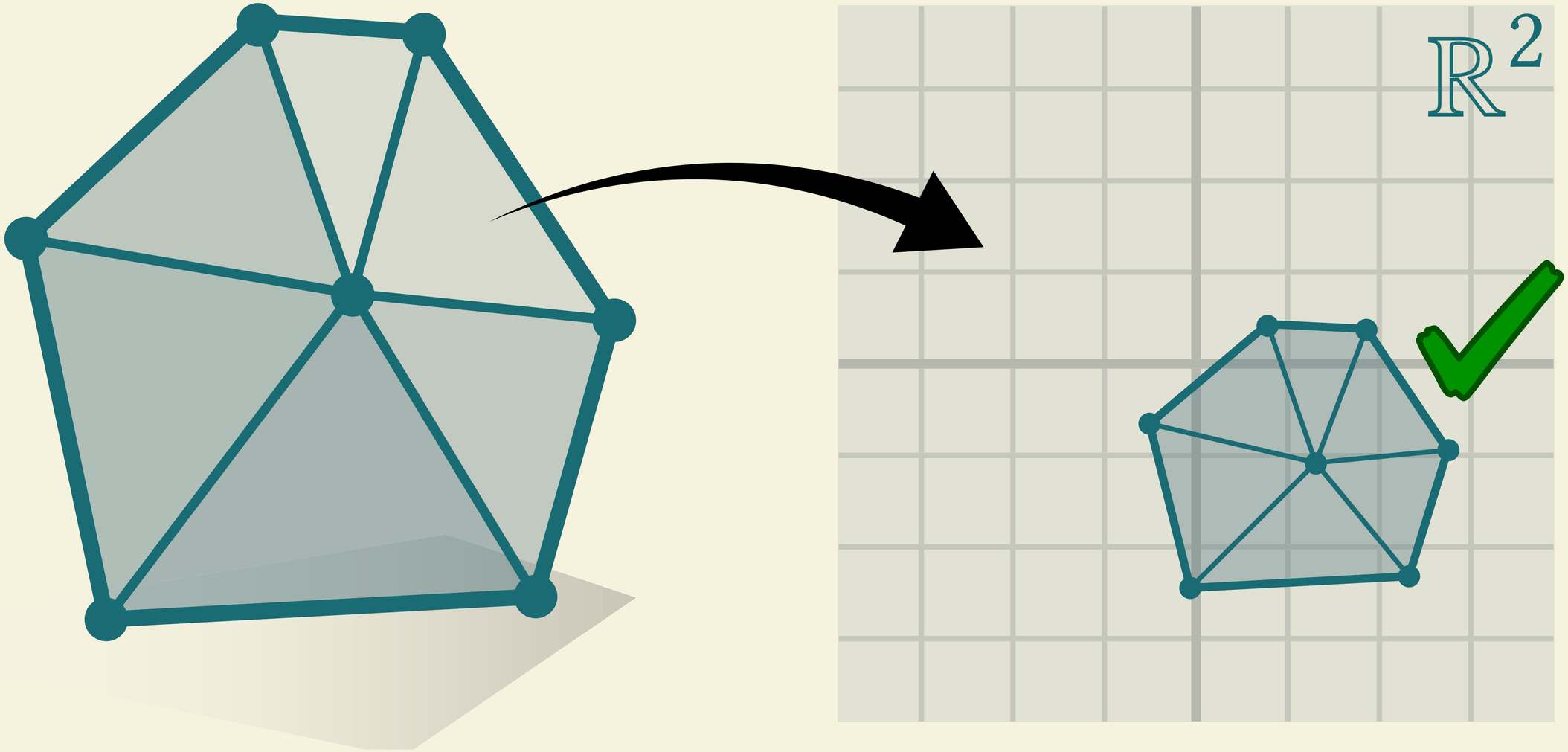
- Map every vertex into the plane
  - ~~Collapse?~~ Not injective
  - Cut open and unfold



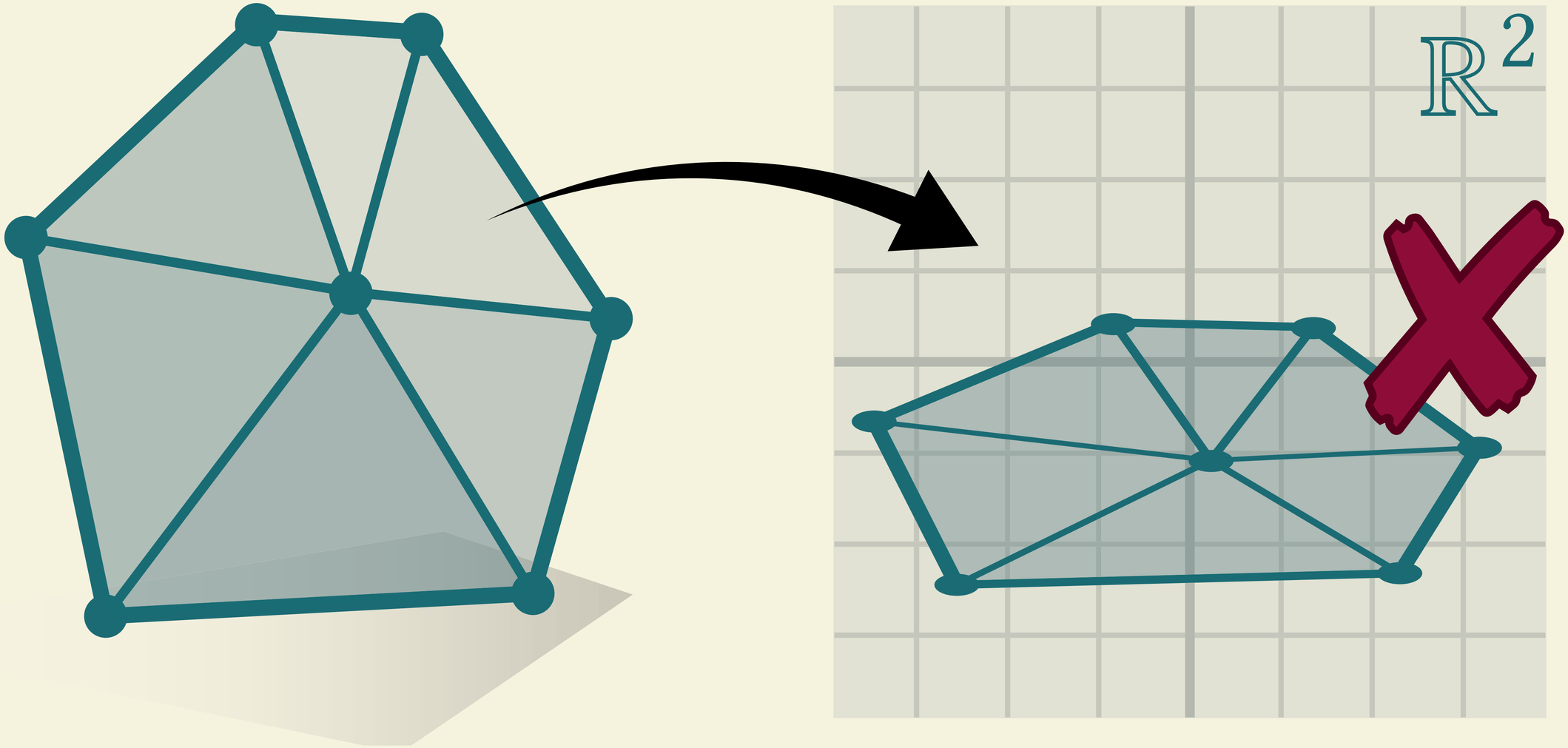
# How can we compute a good unfolding?



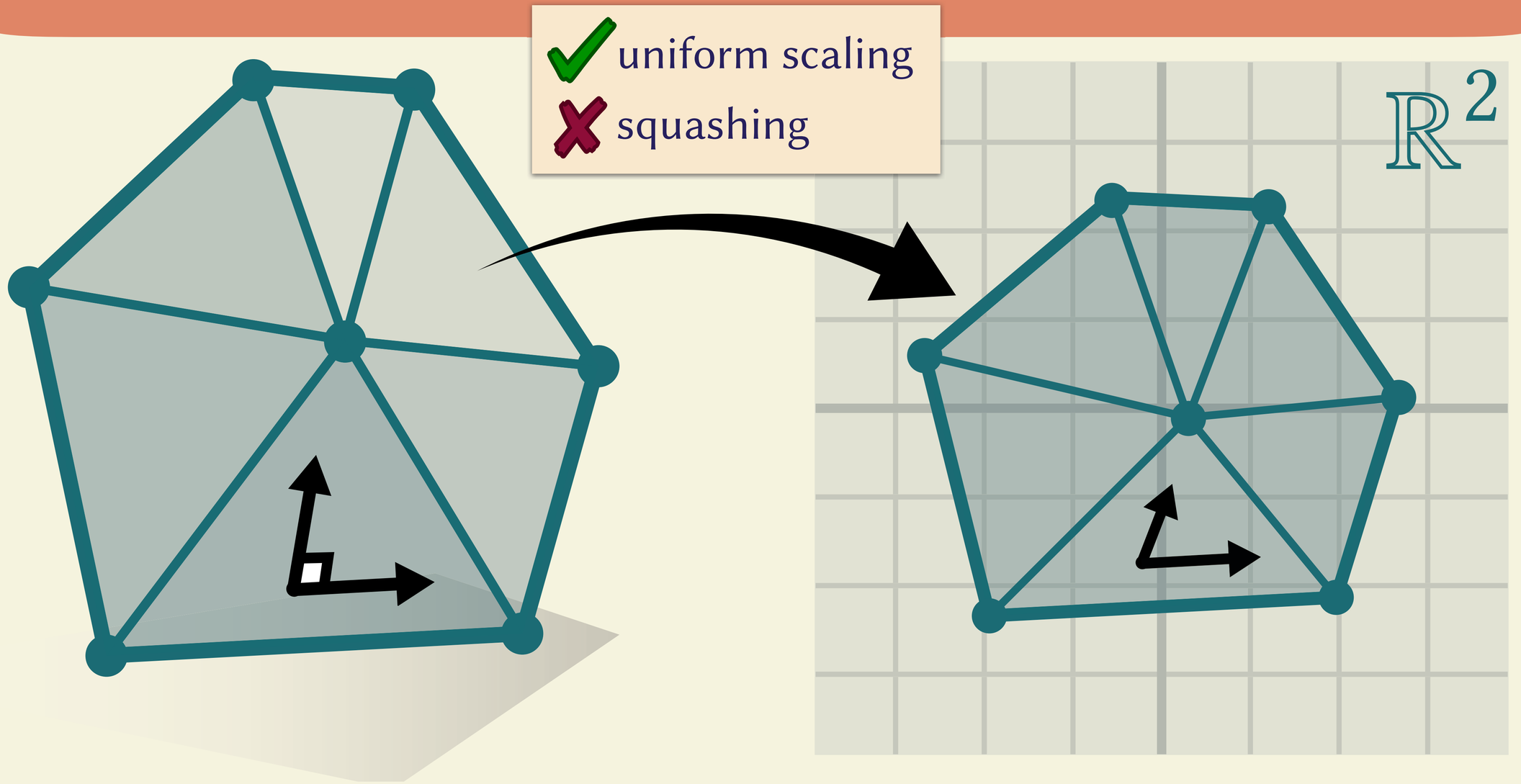
# How can we compute a good unfolding?



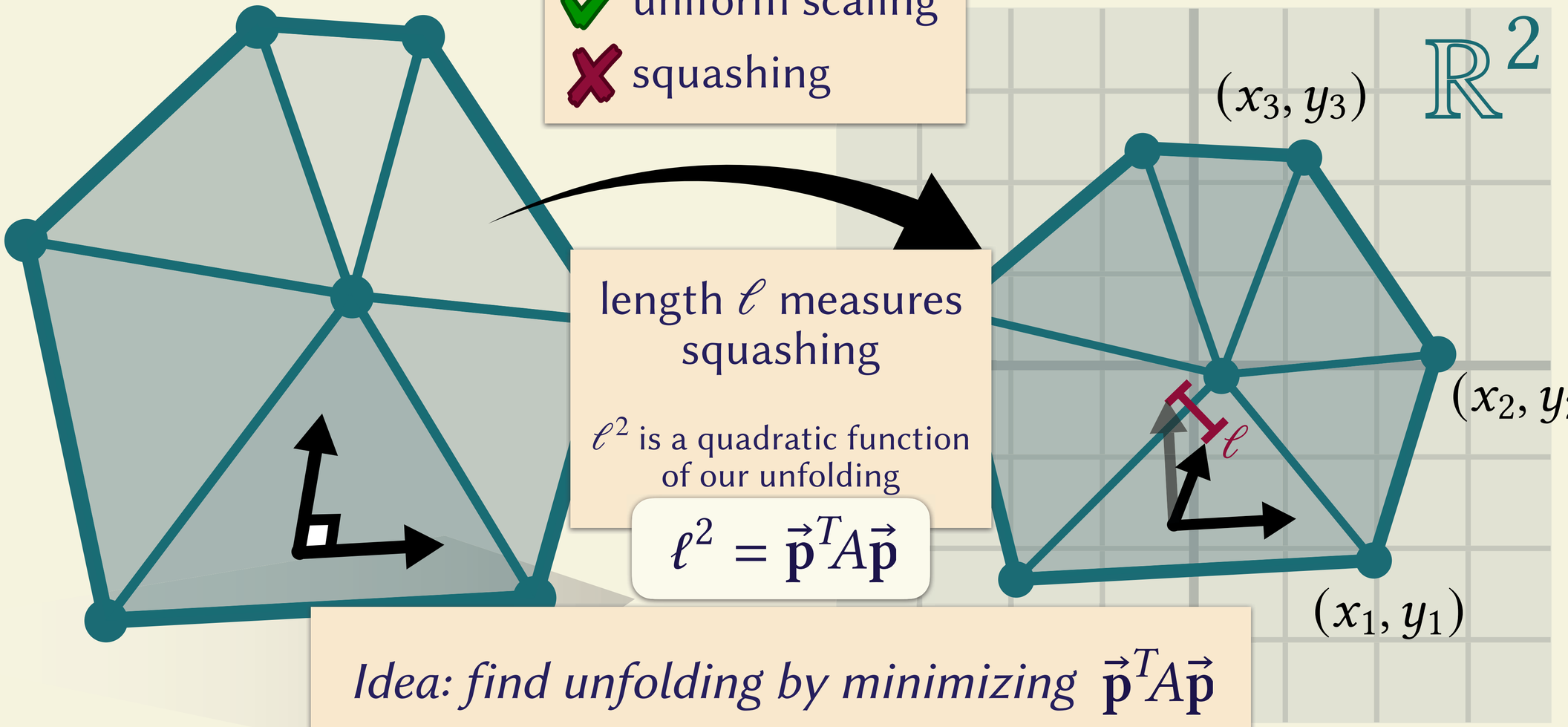
# How can we compute a good unfolding?



# How can we compute a good unfolding?



# How can we compute a good unfolding?

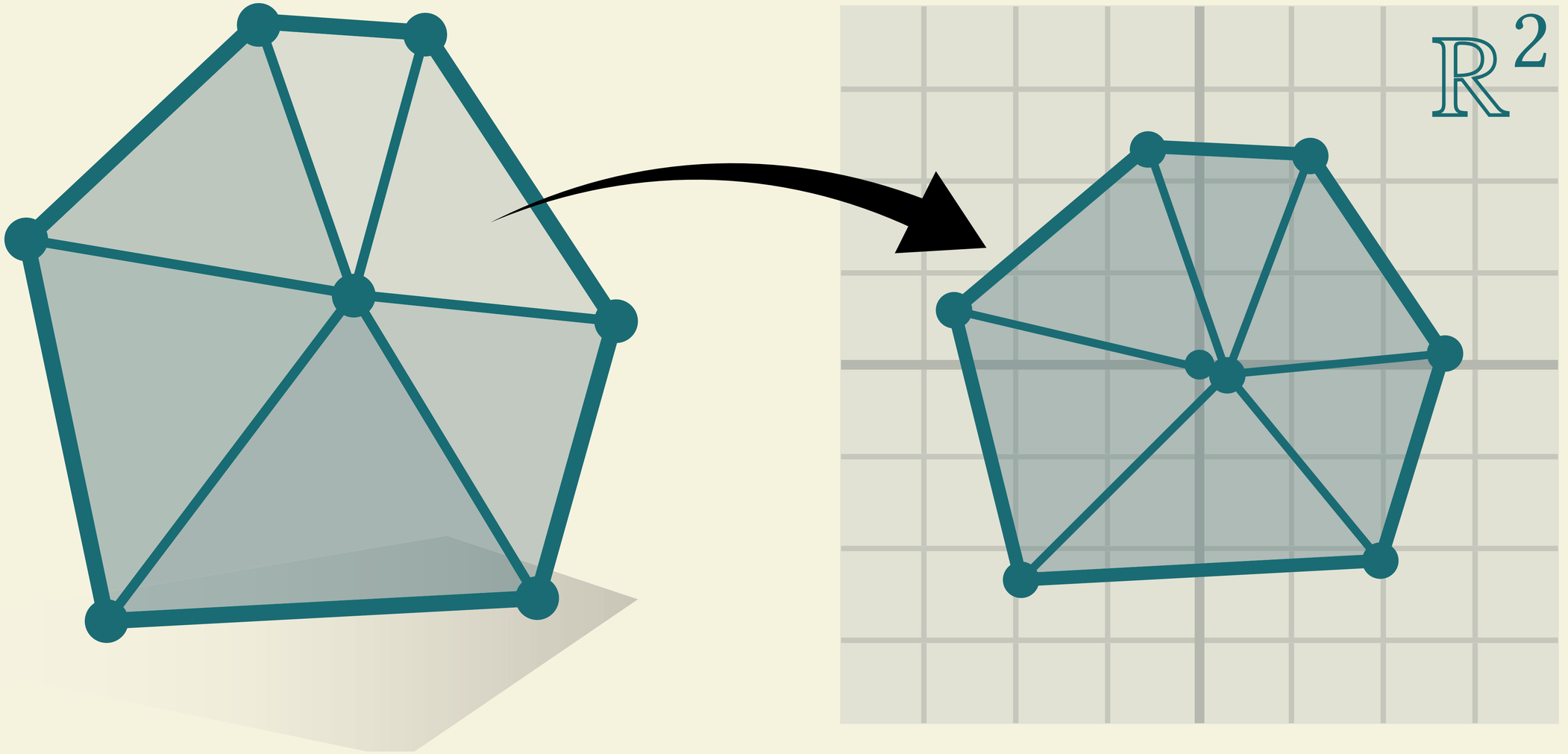



*Idea: find unfolding by minimizing* $\vec{\mathbf{p}}^T A \vec{\mathbf{p}}$

# Something is not quite right

*Idea: find unfolding by minimizing* $\vec{\mathbf{p}}^T A \vec{\mathbf{p}}$

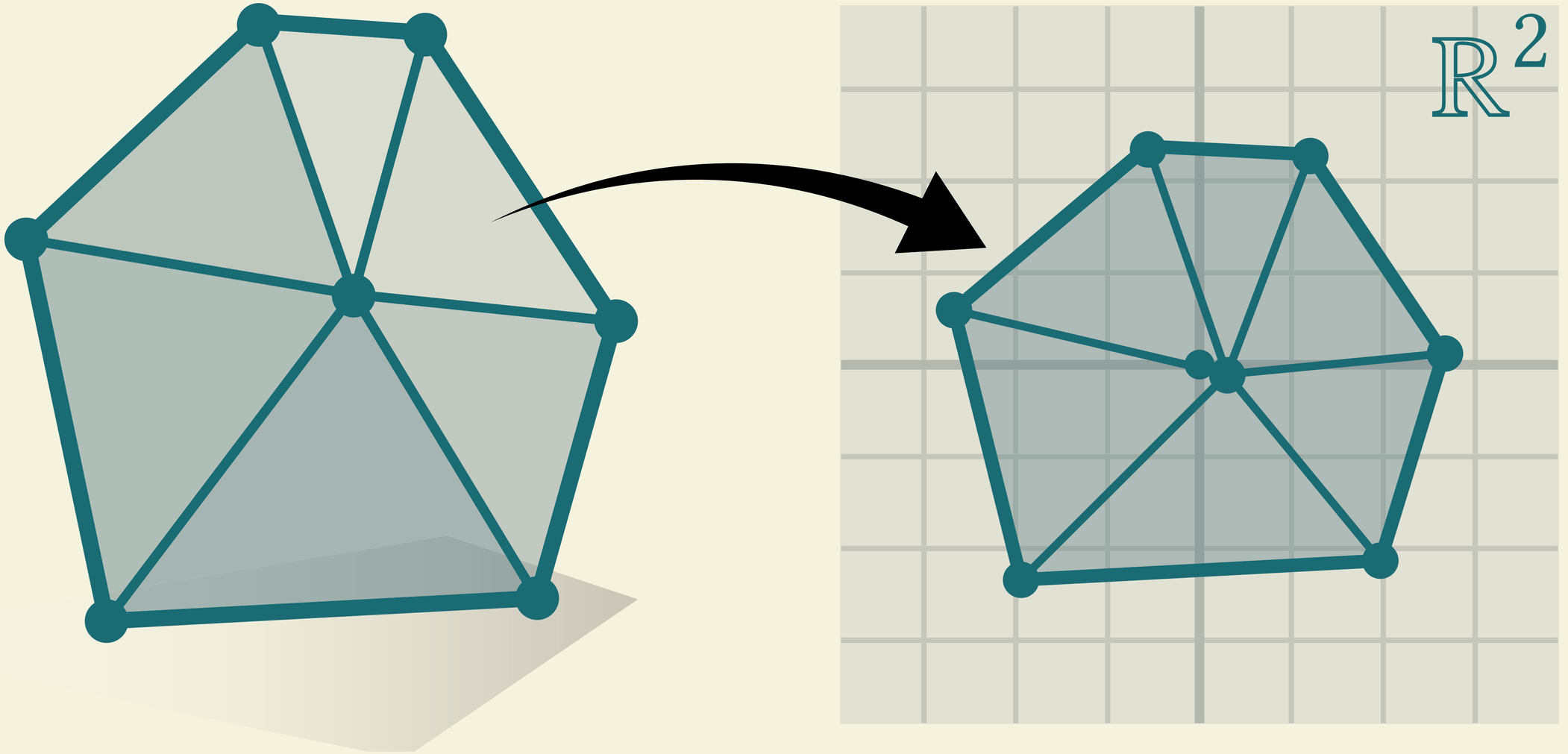
solution is always $\vec{\mathbf{p}} = \vec{0}$



# Eliminating the trivial solution

*Idea: find unfolding by minimizing* $\vec{\mathbf{p}}^T A \vec{\mathbf{p}}$

subject to the constraint that $\vec{\mathbf{p}}^T \vec{\mathbf{p}} = 1$



# Aside: How do we solve this?

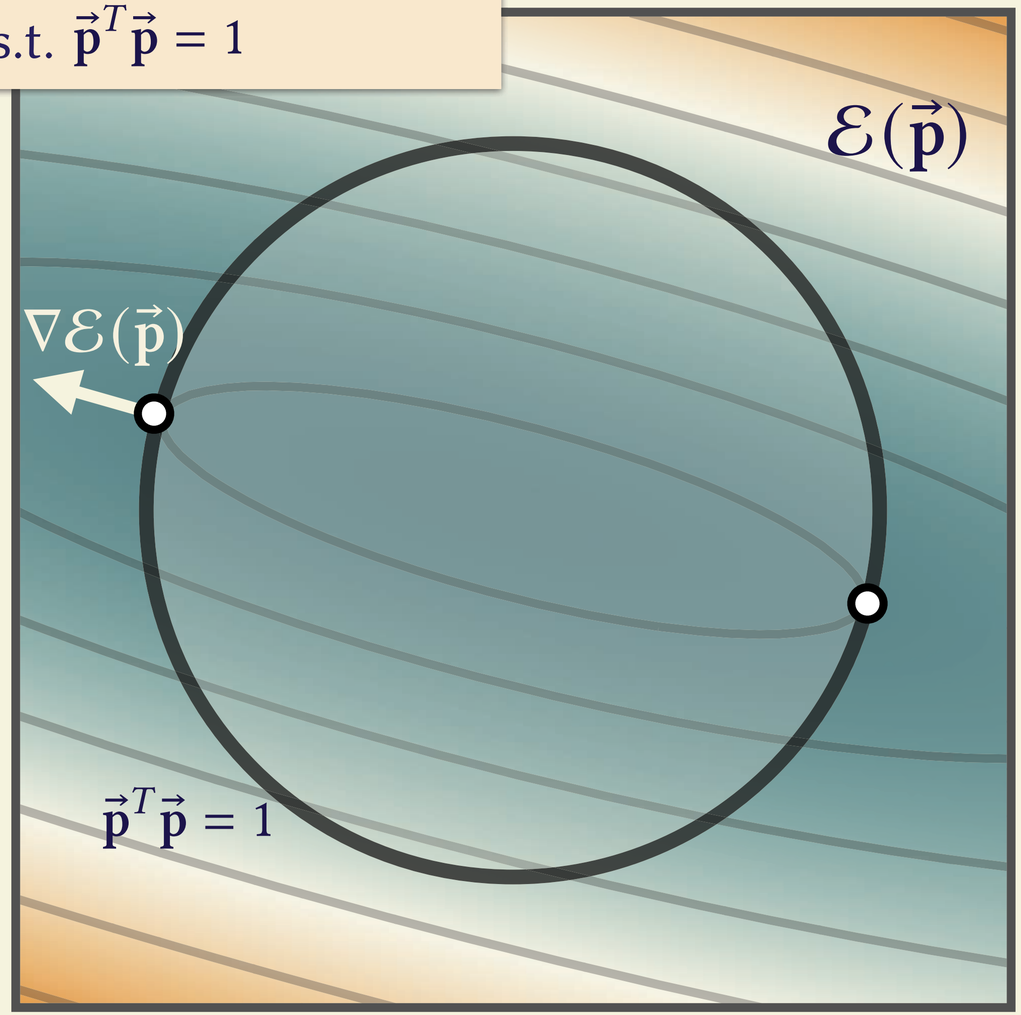
minimize $\vec{\mathbf{p}}^T A \vec{\mathbf{p}} =: \mathcal{E}(\vec{\mathbf{p}})$

s.t. $\vec{\mathbf{p}}^T \vec{\mathbf{p}} = 1$

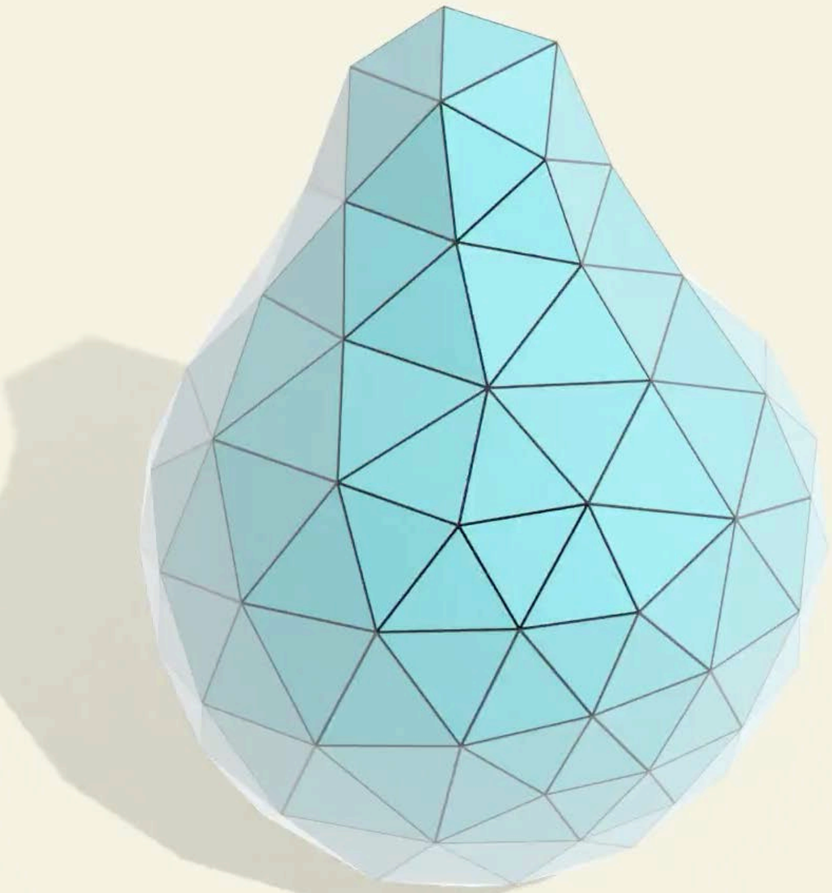
$$A\vec{\mathbf{p}} = \lambda\vec{\mathbf{p}}$$

*eigenvector problem*

(method of *Lagrange multipliers*)



# Now we can finally compute parameterizations

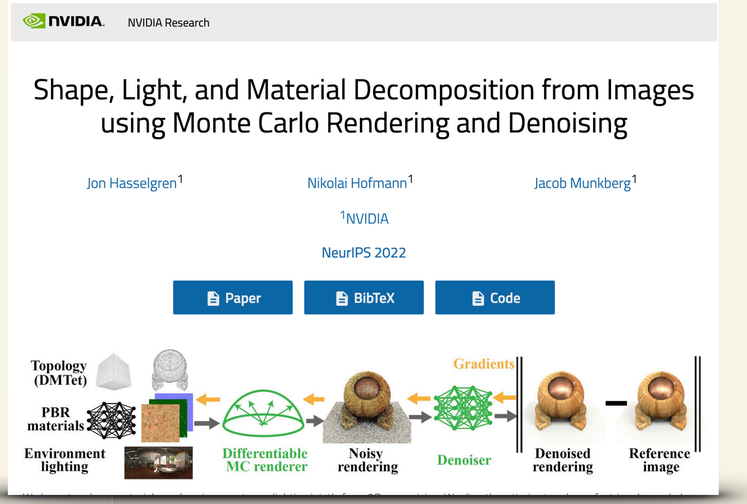











Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. 2002. Least squares conformal maps for automatic texture atlas generation. ACM Trans. Graph. 21, 3 (2002). https://doi.org/10.1145/566654.566590

Patrick Mullen, Yiying Tong, Pierre Alliez, and Mathieu Desbrun. 2008. Spectral conformal parameterization. Symposium on Geometry Processing (SGP '08), 1487–1494.

# Room for improvement

## artist control

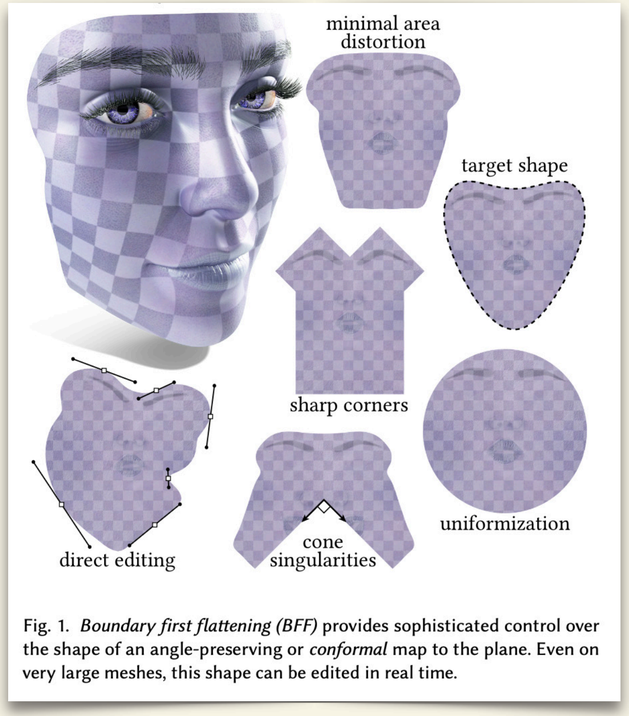
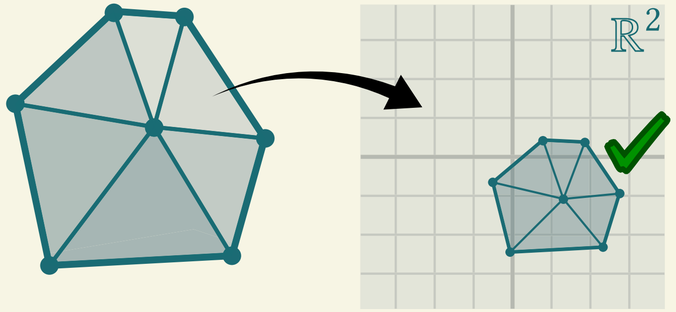


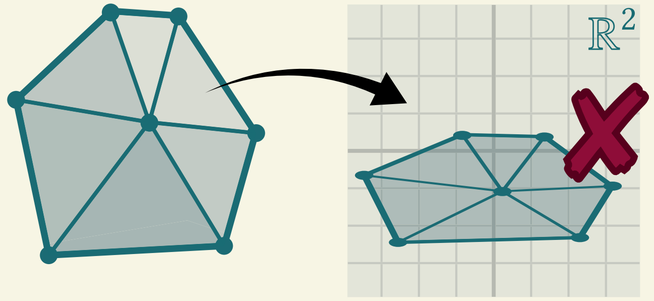

Fig. 1. *Boundary first flattening (BFF)* provides sophisticated control over the shape of an angle-preserving or *conformal* map to the plane. Even on very large meshes, this shape can be edited in real time.

Rohan Sawhney and Keenan Crane. 2017. Boundary First Flattening. ACM Trans. Graph. 37, 1, Article 5 (2018). https://doi.org/10.1145/3132705

## distortion measurement

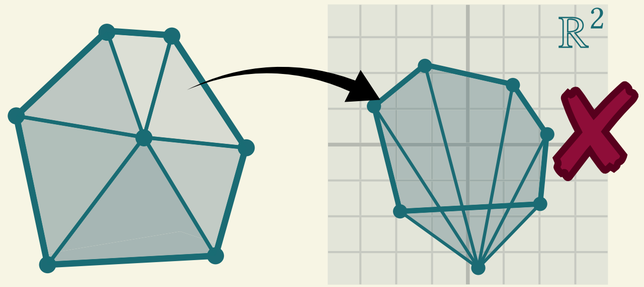



"area distortion"

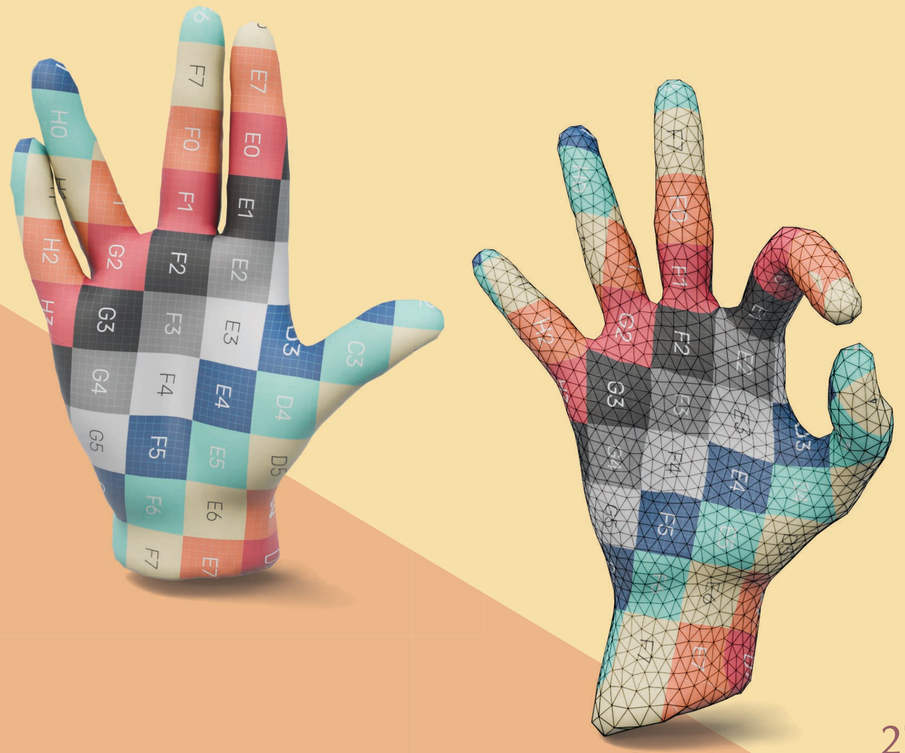



"angle distortion"

## injectivity



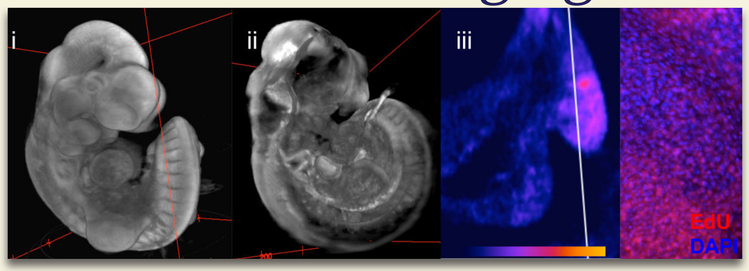
# Shape Correspondence



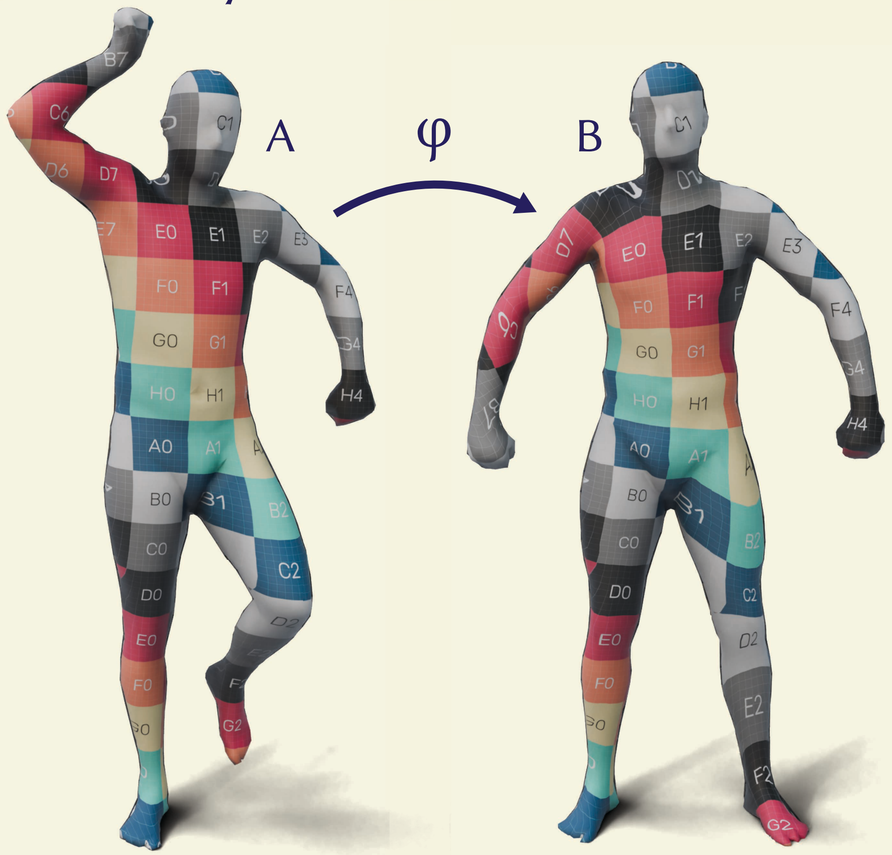
# Problem:
# Mapping between Surfaces

required to study data over collections of shapes

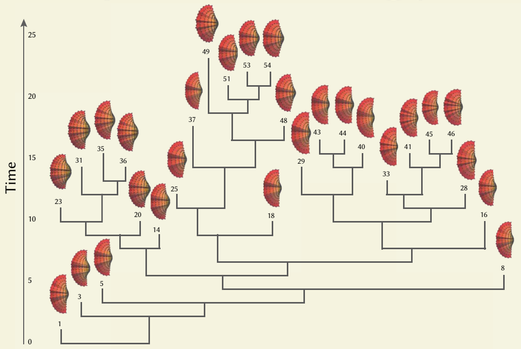
## medical imaging




[ Hallgrimsson et al. 2015 ]




## paleontology



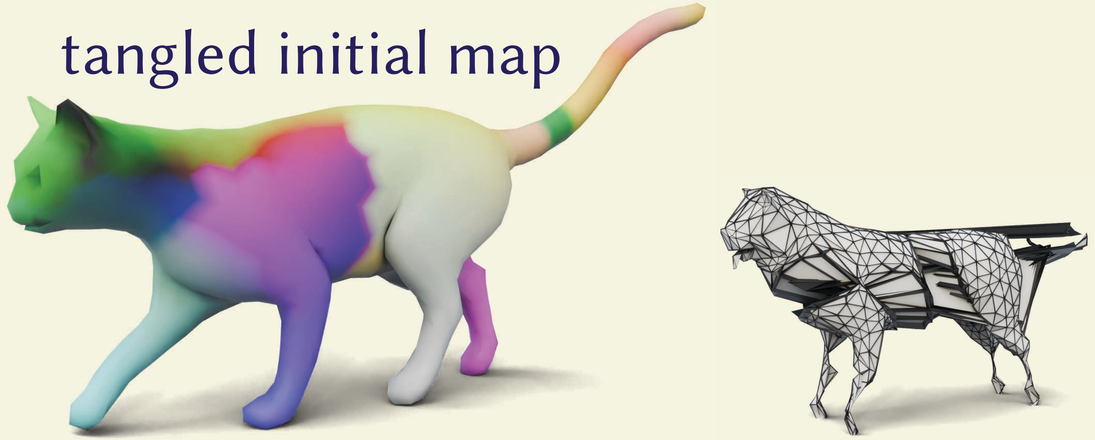

[ Polly & Motz 2016 ]

Etienne Corman, Yousuf Soliman, Robin Magnet, and Mark Gillespie. 2026.
Implicit Minimal Surfaces for Bijective Correspondences.
ACM Trans. Graph. 45, 4, Article 162. https://doi.org/10.1145/3811368

# Map Untangling

## Fixing Flips

tangled initial map





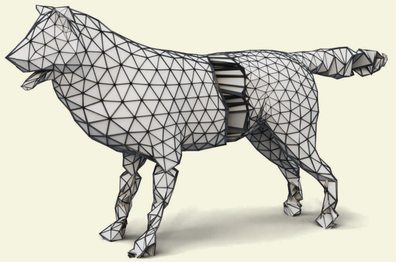


source

RHM

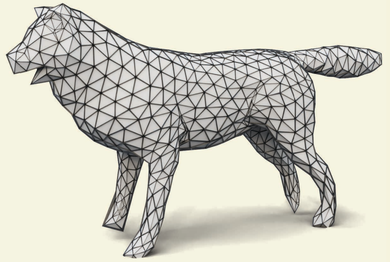
[ Ezuz et al. 2019 ]



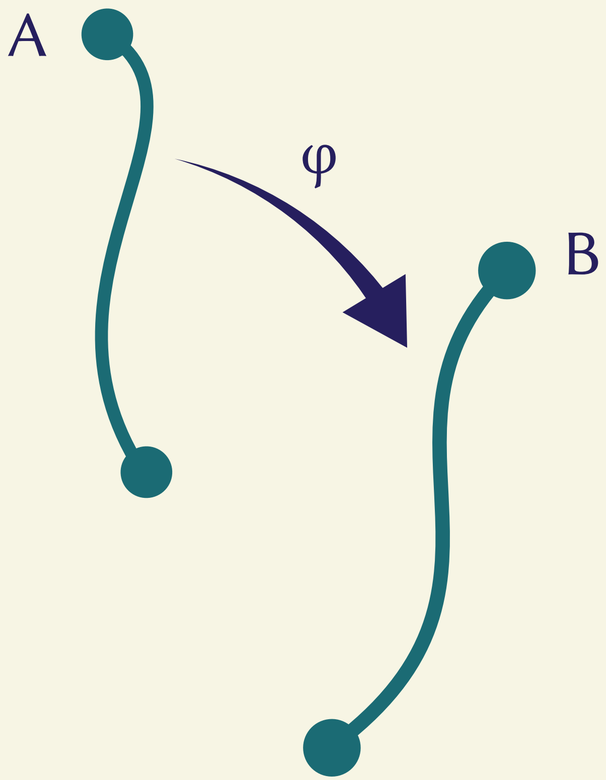


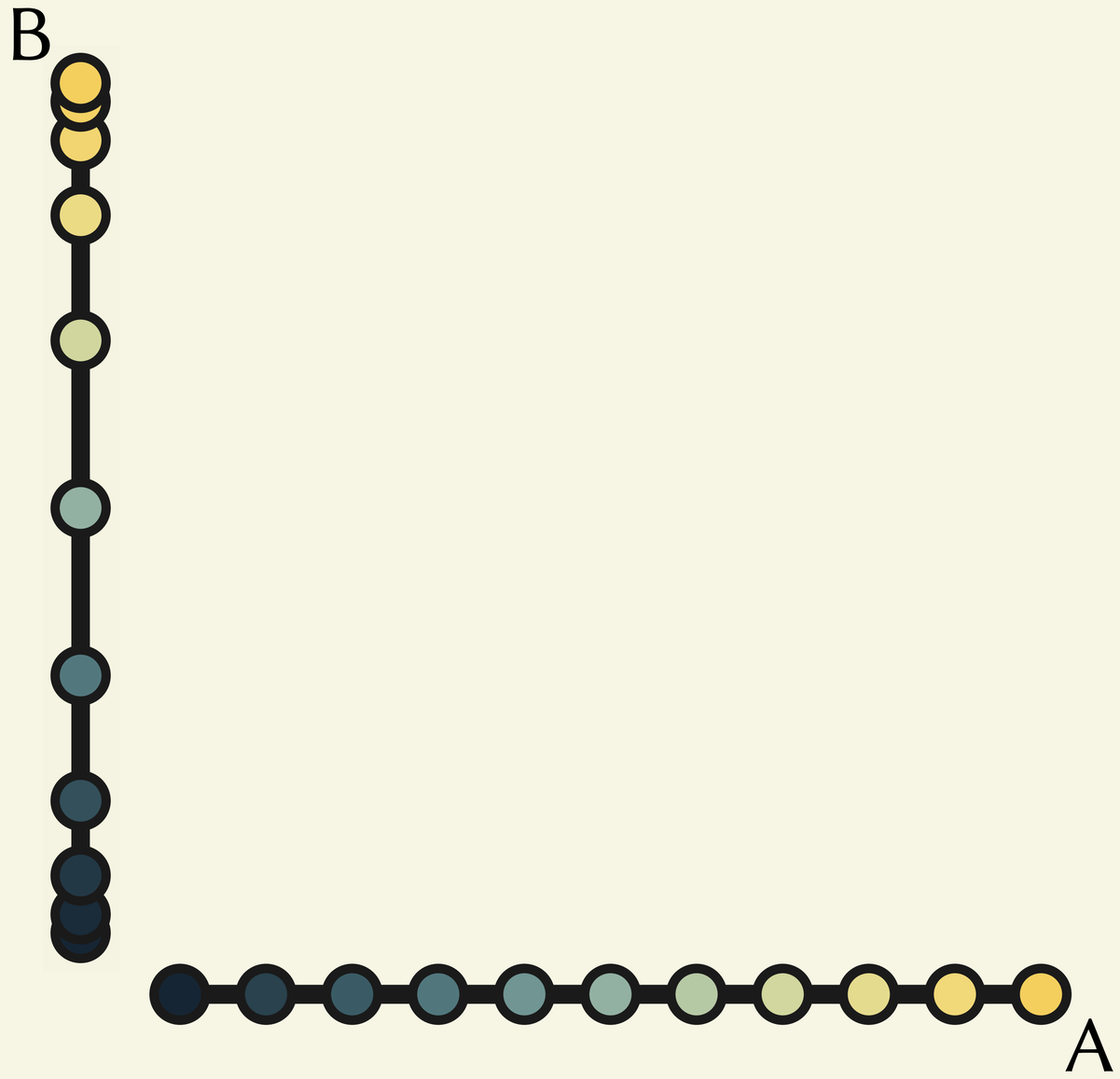
ours



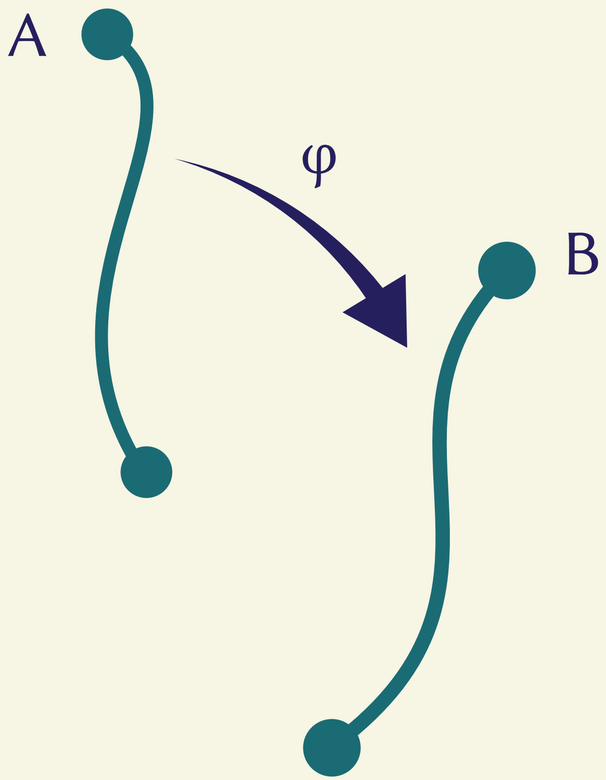


# Example:
# 1D Curves

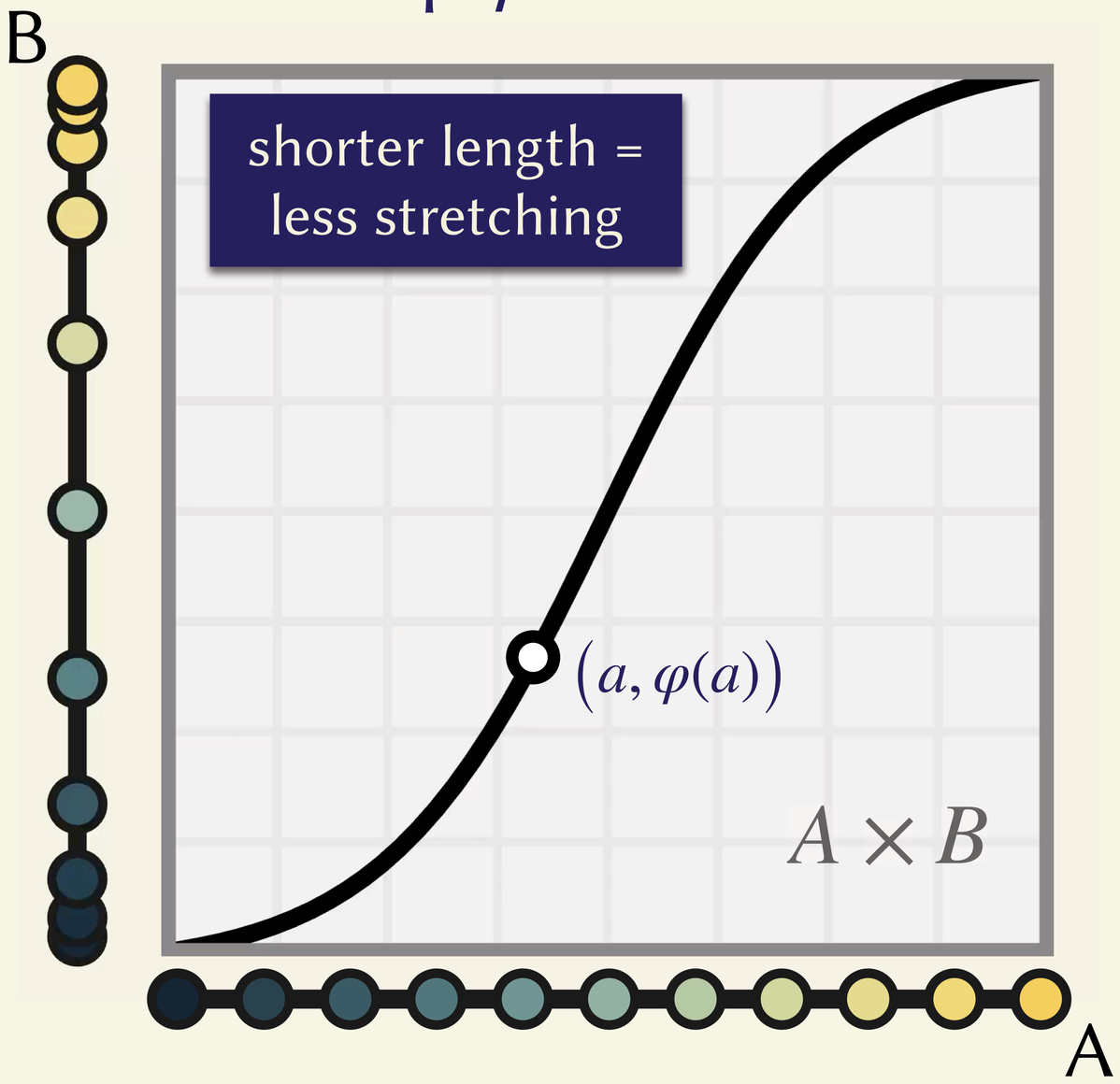


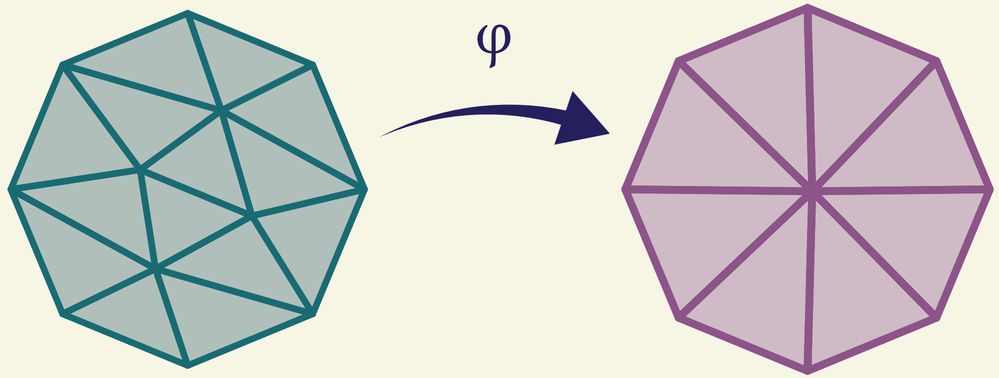
# Example: 1D Curves



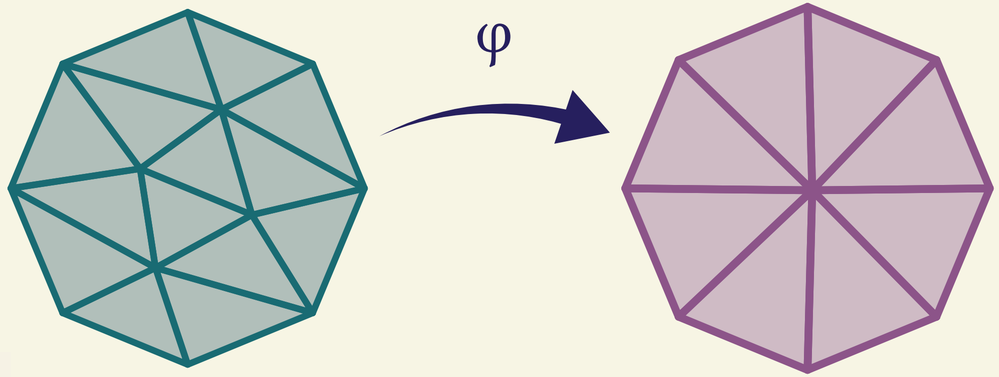

map $\varphi : A \to B$



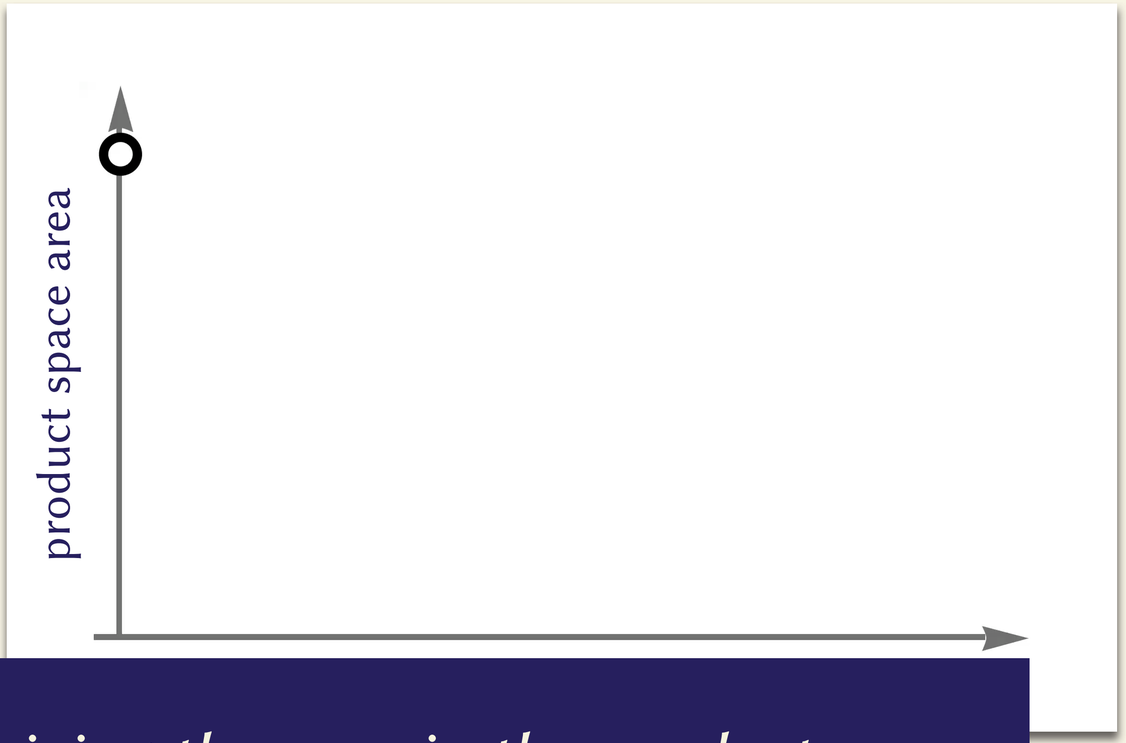
# Example:
# 2D Disks



# Example: 2D Disks






*Find a good map by minimizing the area in the product space*

# Map Distortion



source hand

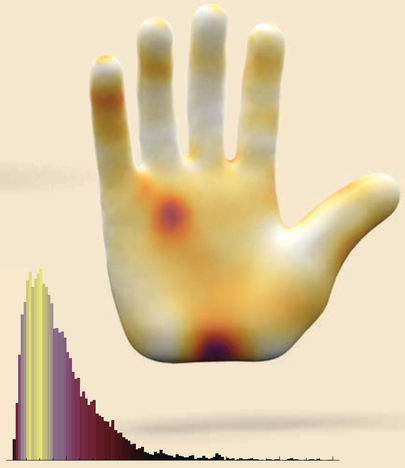


map
[ ours ]

distortion
(area)



distortion
(symmetric Dirichlet)



our simpler optimization yields lower distortion in both senses



map
[ Schmidt et al. 2020 ]

distortion
(area)

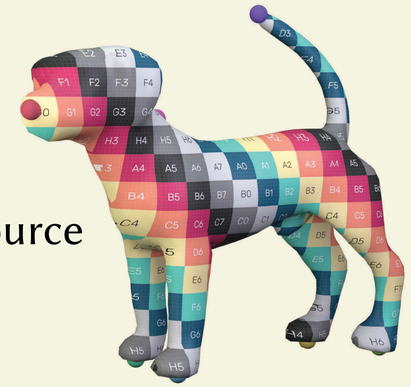


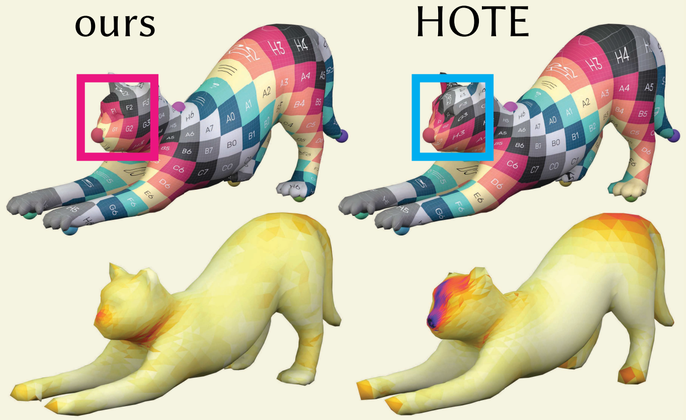
distortion
(symmetric Dirichlet)



# vs Hyperbolic Orbifold Tutte Embedding

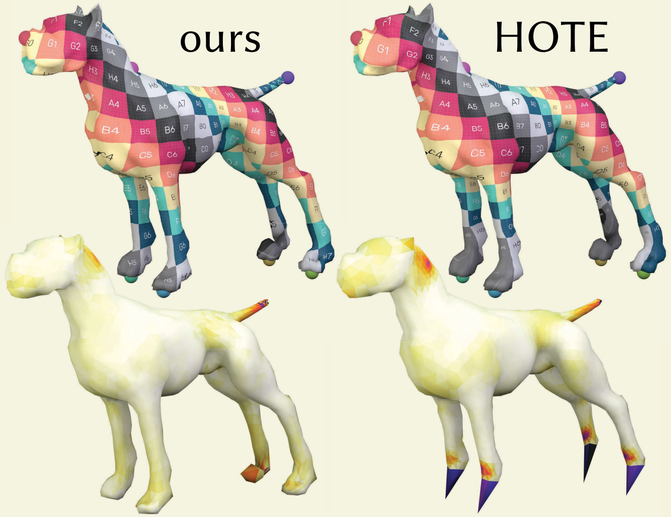
## [Aigerman and Lipman 2016]

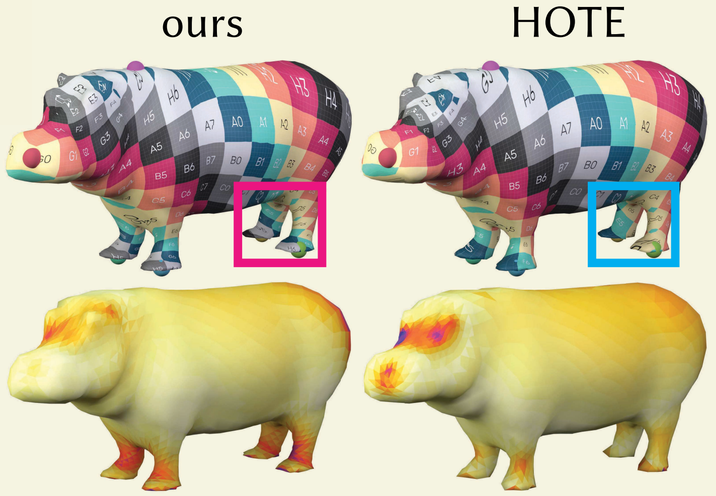






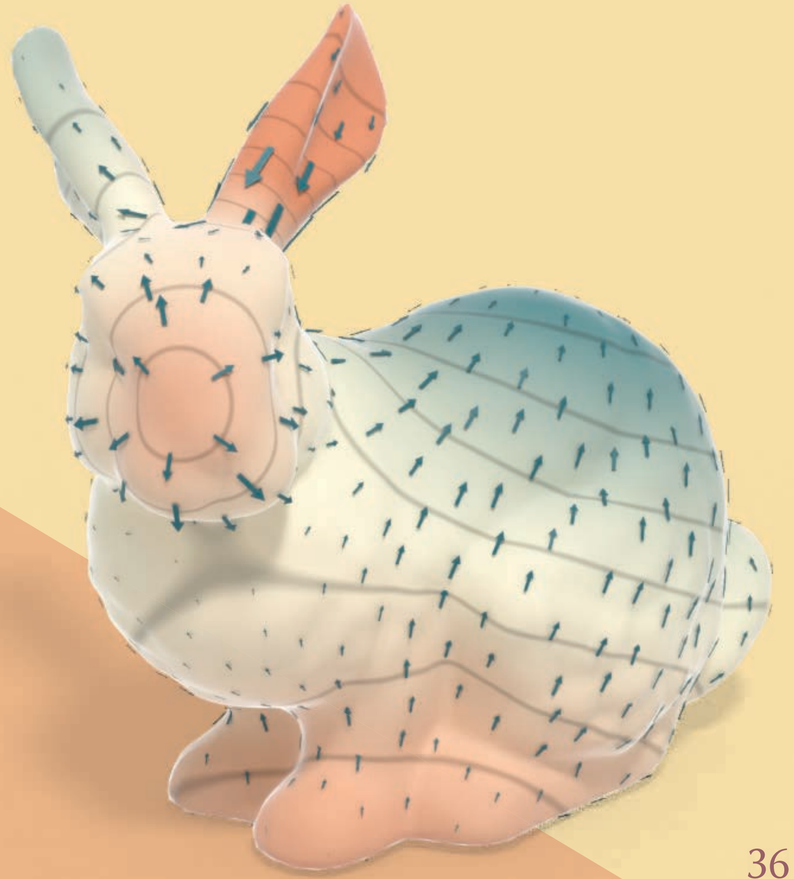





Symmetric Dirichlet Energy

| | ours | HOTE |
|---|---|---|
| cat | 2.7 | 13.5 |
| hippo | 8.9 | 25582.5 |
| dog | 3.4 | 14.8 |

# Vector Field Design



# Tangent vectors



# Vector fields



# Vector fields are used all over the place


Fluid Cohomology

HANG YIN, University of California, San Diego
MOHAMMAD SINA NABIZADEH, University of California, San Diego
BAICHUAN WU, University of California, San Diego
STEPHANIE WANG, University of California, San Diego
ALBERT CHERN, University of California, San Diego

Method with fixed harmonic part | Our method | Velocity-based method

Fig. 1. With torus-shaped obstacles in the fluid domain, the cohomology of the fluid velocity becomes nontrivial. By incorporating the correct time evolution of harmonic part, our method can faithfully capture the vortex shedding phenomenon, unlike the velocity-based projection method (left). We compare our result against ground truth which is produced by a velocity-based method using pressure projection (right).

The vorticity-streamfunction formulation for incompressible inviscid fluids is the basis for many fluid simulation methods in computer graphics, including the recent vortex-particle methods and smoke simulation methods in CAD methods. We point out that current setups in the vorticity-streamfunction formulation are insufficient at simulating fluids on general non-simply-connected domains. The resulting velocity fields from the current boundary, boundaries, and nonzero genus can all make the fluid domains multiply connected. These scenarios introduce nontrivial cohomology components to the fluid velocity fields, and the recovery of the harmonic part of the velocity fields have been previously overlooked. In this paper, we derive the missing equations of motion for the fluid cohomology components. We elucidate the physical significance of our equations, and show that our new formulation is reproducing physically correct behaviors of fluid flows on domains with general topology.

CCS Concepts: • Computing methodologies → Physical simulation; • Mathematics of computing → Algebraic topology; Partial differential equations; • Applied computing → Physics.

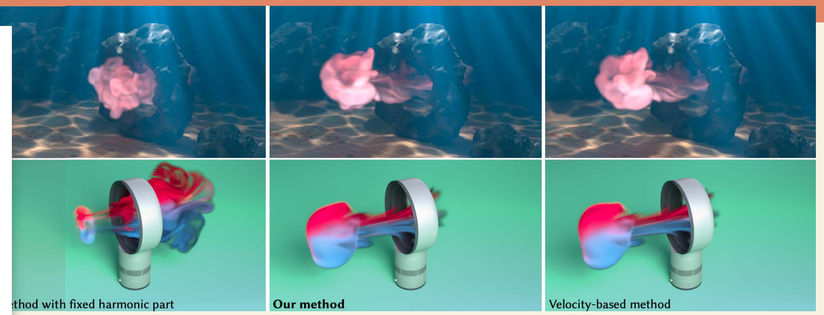
Additional Key Words and Phrases: Fluid dynamics, homology theory

ACM Reference Format:
Hang Yin, Mohammad Sina Nabizadeh, Baichuan Wu, Stephanie Wang, and Albert Chern. 2021. Fluid Cohomology. ACM Trans. Graph. 42, 4, Article 1 (August 2023), 19 pages. https://doi.org/10.1145/3450626

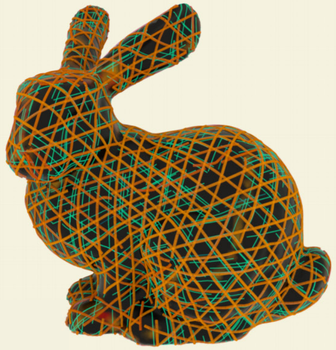
1 INTRODUCTION

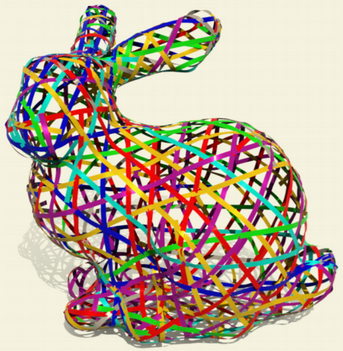
Vortex methods in the fluid simulation in computer graphics is to approximate the solutions to the incompressible Euler equations which govern the dynamics of the velocity fields of incompressible Inviscid fluids. One approach is to instead simulate the equations in the vorticity-streamfunction formulation of fluid dynamics. This is a system of equations that describes the time evolution of the curl of the velocity field is reconstructed from the vorticity through a streamfunction [Bridson 2015, §14.]. Chorin and Marsden 1990, §3.] Vortex methods are popular for their ability to capture fluid flow's symbols over the preservation of local vorticity. The velocity fields derived from the streamfunction are automatically divergence free. Researchers have adopted this formulation to simulate vortex dynamics [Cantin et al. 1995; Elcott et al. 2007; Zhang et al. 2012] and two-phase flows [Ando et al. 2015a]. Other usages of the

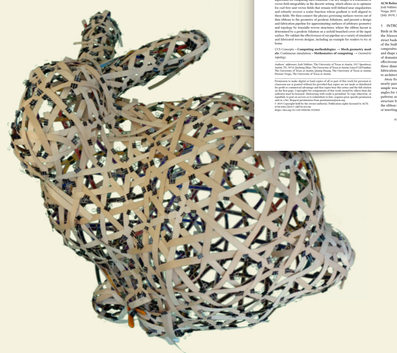
ACM Trans. Graph., Vol. 42, No. 4, Article 1. Publication date: August 2023.




fluid simulation



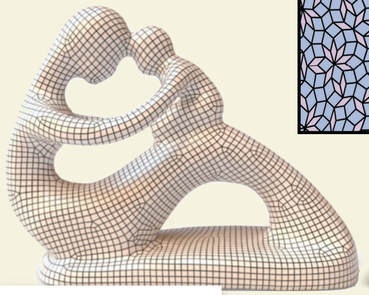


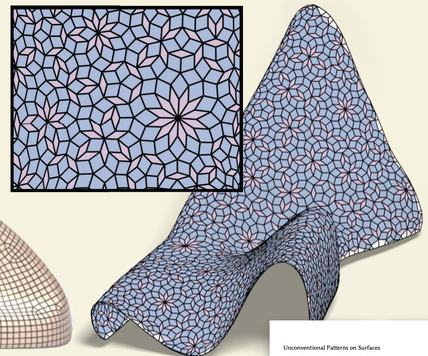



Weaving Geodesic Foliations

JOSH VEKHTER, The University of Texas at Austin
JIAYI EVA YANG, The University of Texas at Austin
LUISA F. GIL PANDINO, The University of Texas at Austin
QIXING HUANG, The University of Texas at Austin
ETIENNE VOUGA, The University of Texas at Austin

computational fabrication





mesh generation


Integer-Grid Maps for Reliable Quad Meshing

David Bommes¹ Marcel Campen² Hans-Christian Ebke² Pierre Alliez³ Leif Kobbelt²
¹Inria Sophia Antipolis - Méditerranée ²RWTH Aachen University ³Inria Nancy

Greedy Rounding + Splitting | Our Reliable Approach

Figure 1: Quilt-State-of-the-art parameterization based quad generators, working with greedy rounding and splitting, perform well if the sought cut-graph is easy to discern automatically (a-c). In absence of a dominant cut-graph the proposed algorithm produces a valid integer-grid map (d) which can be used to extract a quad mesh (e). Our algorithm produces a valid integer-grid map for any target sizing and thus in addition to uniform quad remeshing can be applied to coarse quad layout abstraction as well. The target edge length, indicated by lines, is identical for the left and the right result.

Abstract

Quadrilateral remeshing approaches based on global parameterization have received significant attention in recent years. The idea of computing a smooth cross field (or its dual counterpart, a pair of scalar functions) whose singularities are then used to construct a coarse quad layout has inspired many powerful algorithms. Apart from their strengths, these methods also have a few weaknesses. Most notably, they tend to yield meshes that, for intermediate to coarse target edge lengths, are rather irregular. Even worse, they frequently fail to produce a valid result altogether, i.e., they are not able to robustly maintain a user-prescribed sizing function. In this paper we propose a remeshing approach that addresses these issues. It is motivated by a novel quality measure for integer-grid maps which assesses the expected distortion of a corresponding triangle mesh within the class of so called Integer-Grid Maps that are guaranteed to induce valid meshes. We present a practical algorithm for IGMs that are to the extent possible aligned with a given cross field, and which feature only few irregularities. Moreover, we show how to explicitly control the sizing behavior of the result. Equipped with these tools, we obtain remeshers that combine good control over size and shape of resulting elements with a high degree of quad regularity and alignment.

CR Categories: I.3.5 [Computational Geometry and Object Modeling]: Boundary representations; Geometric transformations

Keywords: remeshing, quadrangulation, parameterization, mixed-integer optimization

Links: DL PDF Video Slides

1 Introduction

The automatic conversion of objects given as triangulated surface meshes into a quadrilateral representation (referred to as quad remeshing) is a challenging task which has attracted lots of attention in the past years. Besides the fact that quad-dominant or pure quad meshes are very appealing due to their favorable performance in numerical simulation, their key benefit is not their mere existence but that they are very well suited to accurately represent feature curves and directional information. Typically, quad edges naturally align to the well-behaved edge flow of a single curve, isocurves between pairs of curves are typically nearly orthogonal and exhibit good tangential continuity across singularities [Ray et al. 2006].

Often quad remeshing is a single building block embedded within a complex application pipeline like, e.g., an animation or simulation framework. In such a context, a remeshing algorithm is fundamentally required to be able to determine a mesh with characteristics (size, shape, alignment) as specified by the user or an adaptive simulation scheme in each of the thousands of time steps automatically, robustly, and in a timely manner. It is thus feasible to manually react on failure cases in the simulation and a

© ACM, 2013. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in ACM Transactions on Graphics, http://dx.doi.org/10.1145/2461912.2462014.

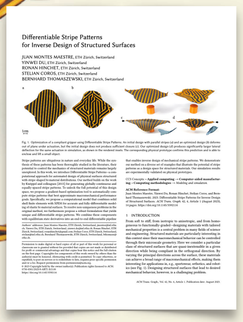




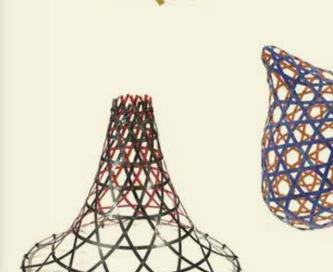

Unconventional Patterns on Surfaces

ABEL MATSUO, Utrecht University, The Netherlands
ANIR YAAMAN, Utrecht University, The Netherlands

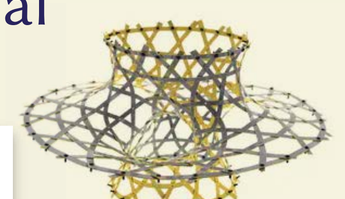
robot grippers


Differentiable Stripe Patterns for Inverse Design of Structured Surfaces

JUAN MONTES MAESTRE, ETH Zürich, Switzerland
YINWEI DU, ETH Zürich, MIT, Switzerland
RONAN HINCHET, ETH Zürich, Switzerland
STELIAN COROS, ETH Zürich, Switzerland
BERNHARD THOMASZEWSKI, ETH Zürich, Switzerland

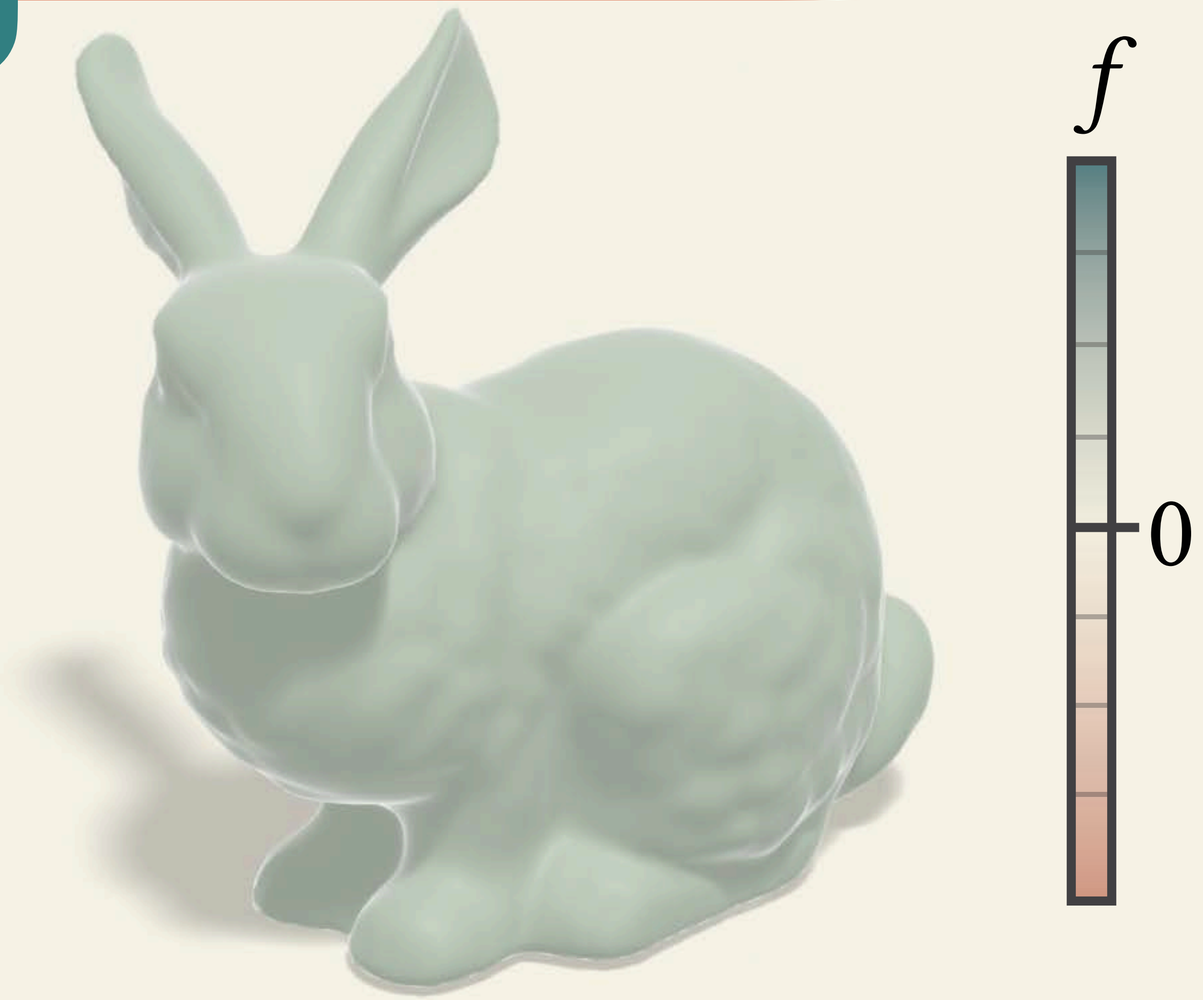

3D Weaving with Curved Ribbons

YINGYING REN, EPFL, Switzerland
JULIAN PANETTA, UC Davis, USA
TIAN CHEN, EPFL, Switzerland
FLORIN ISVORANU, EPFL, Switzerland
SAMUEL POINCLOUX, EPFL, Switzerland
CHRISTOPHER BRANDT, EPFL, Switzerland
ALISON MARTIN, Weaver and Independent Researcher, Italy
MARK PAULY, EPFL, Switzerland



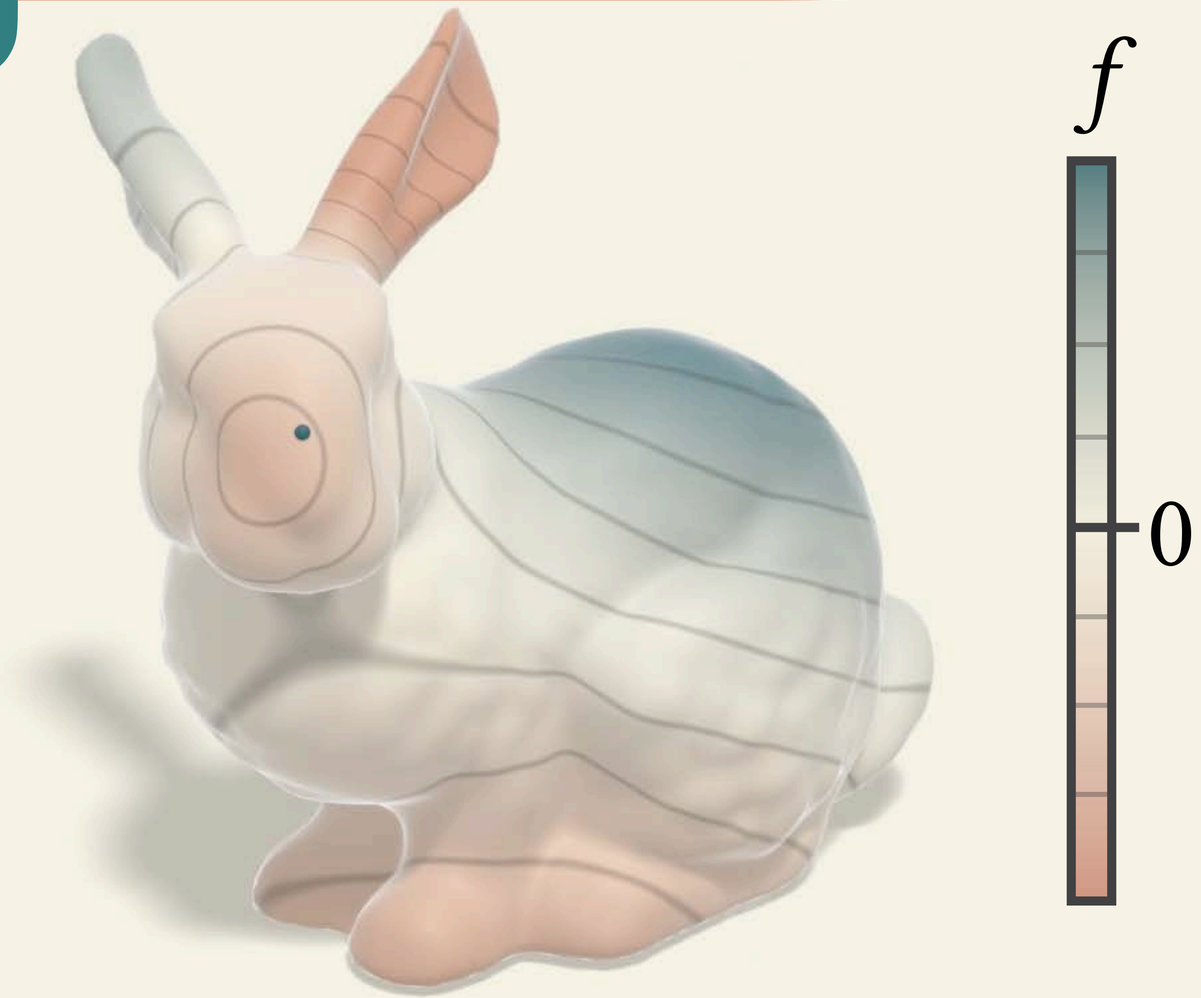
# How do we get a vector field?

gradient fields



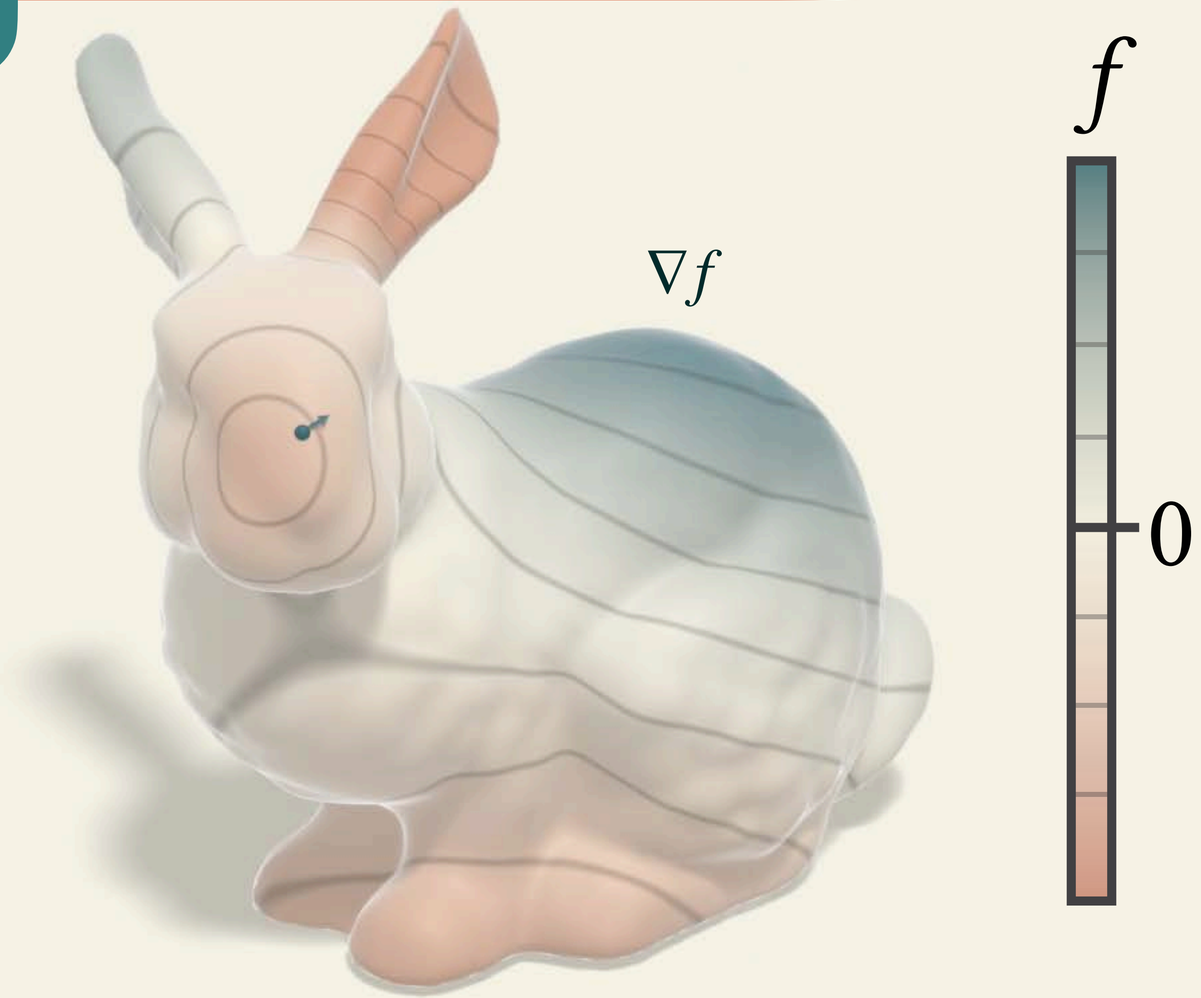
# How do we get a vector field?

gradient fields



# How do we get a vector field?

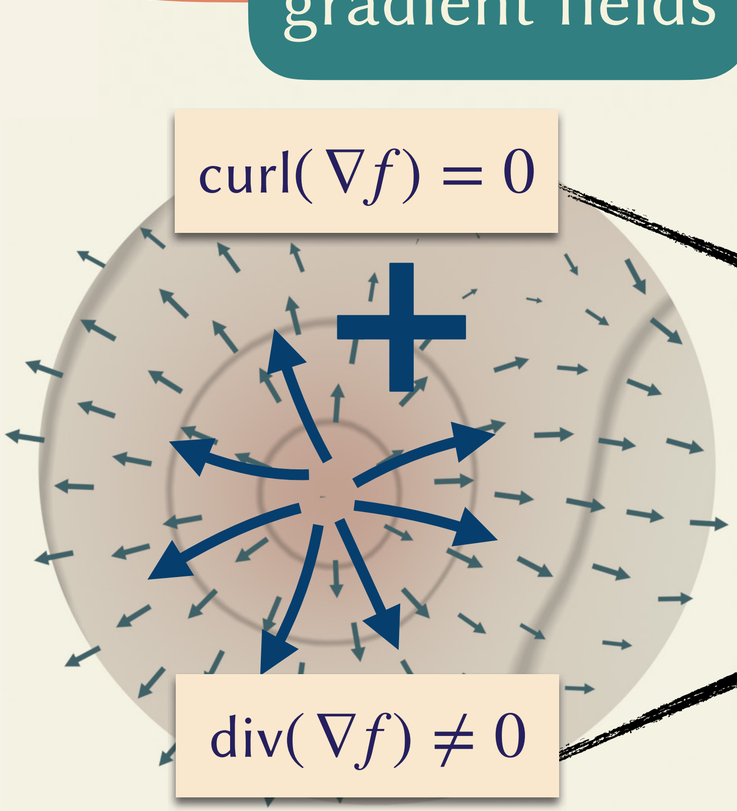
gradient fields



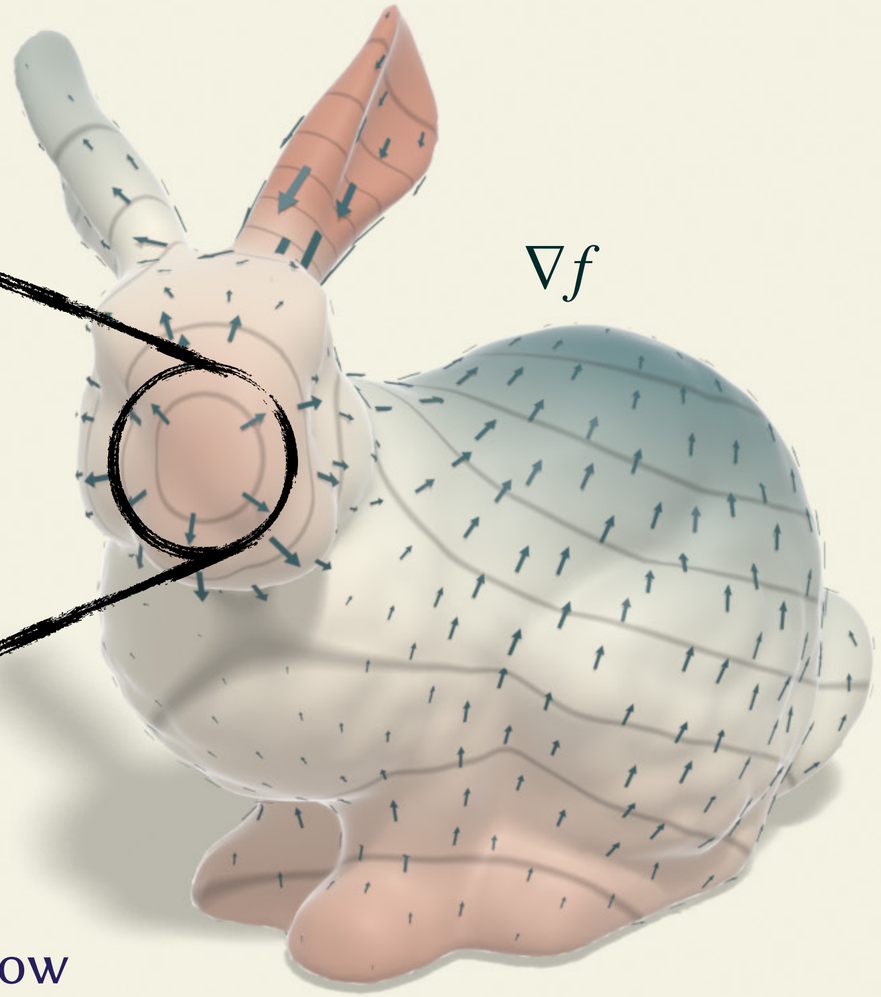
# How do we get a vector field?

gradient fields

$\text{curl}(\nabla f) = 0$



$\text{div}(\nabla f) \neq 0$




$\text{curl}(\nabla f)$ measures rotation

$\text{div}(\nabla f)$ measures inflow/outflow

on the bunny,
*every* field with
curl 0 is a gradient

$f$



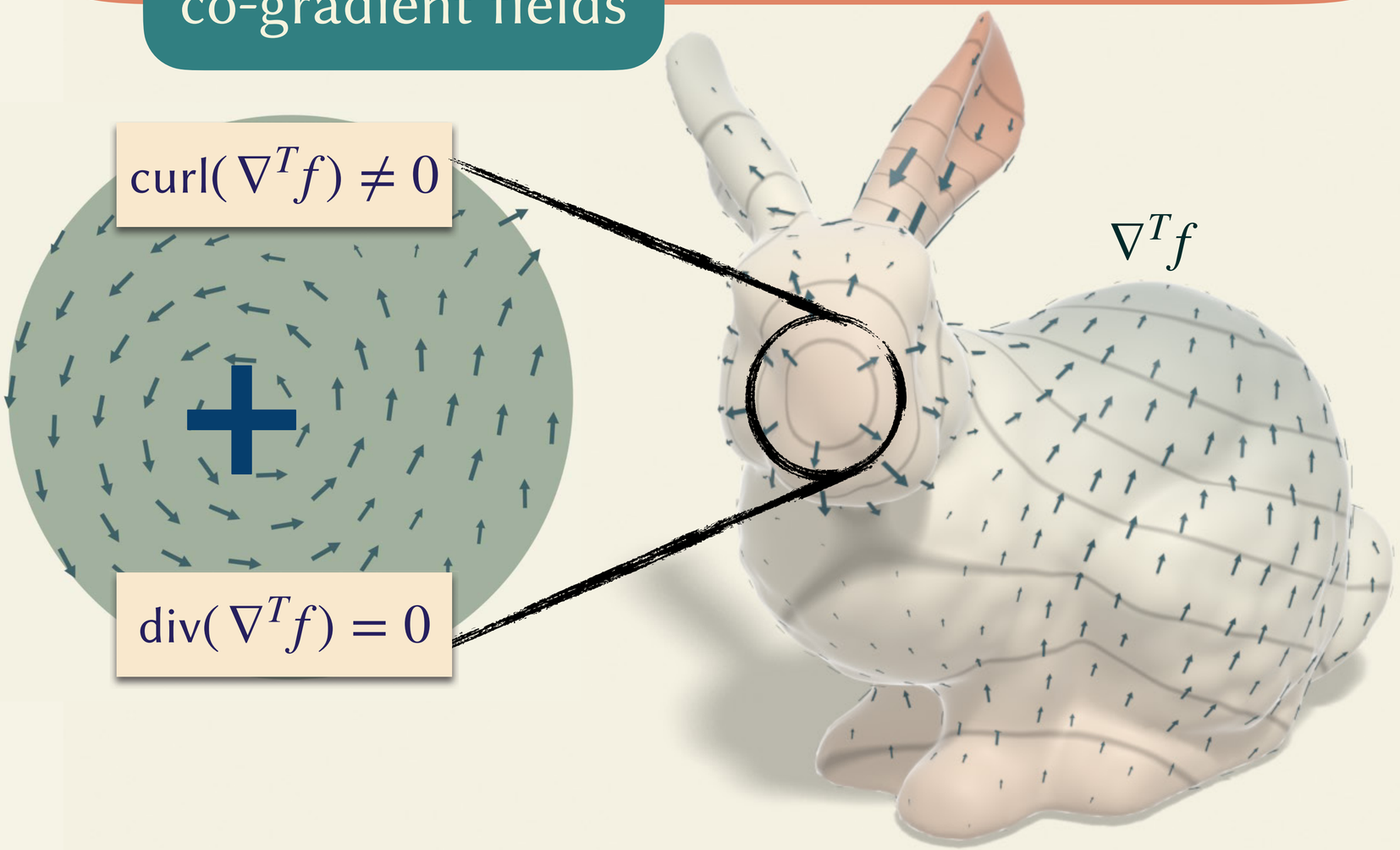

the divergence
determines the field

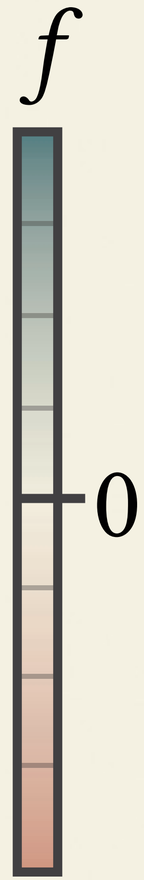
# How do we get a vector field?

co-gradient fields

on the bunny, *every* field with divergence 0 is a co-gradient

the curl determines the field



# Decomposing a vector field

$V$ =  + $\nabla f$

determined by $\mathrm{div}(V)$

$\nabla^T g$

determined by $\mathrm{curl}(V)$

# The story is more complicated on surfaces with handles

$\mathrm{curl}(V) = 0$

but *not* a gradient!



# The story is more complicated on surfaces with handles

$\text{curl}(V) = 0$

but *not* a gradient!

$\int V \, dx > 0$

but $\int \nabla f \, dx$ is always 0

$\text{div}(V) = 0$

but *not* a co-gradient!




only possible on surfaces with handles

essentially 1 field per handle

all other harmonic fields are rotated/scaled copies of this one

important link between vector fields and overall shape topology

*de Rham isomorphism*

# Decomposing a vector field

full version

Helmholtz-Hodge decomposition

$$V = \nabla f + \nabla^T g + h$$



gradient

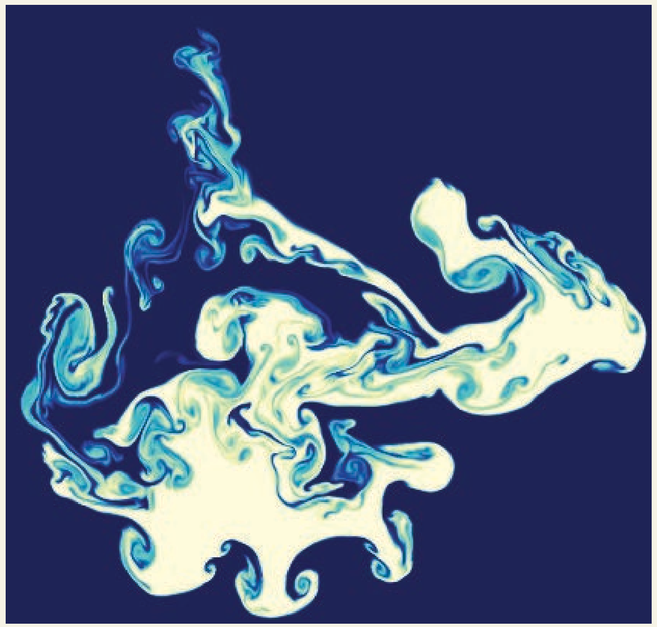
co-gradient

harmonic

# Application: incompressible fluid simulation

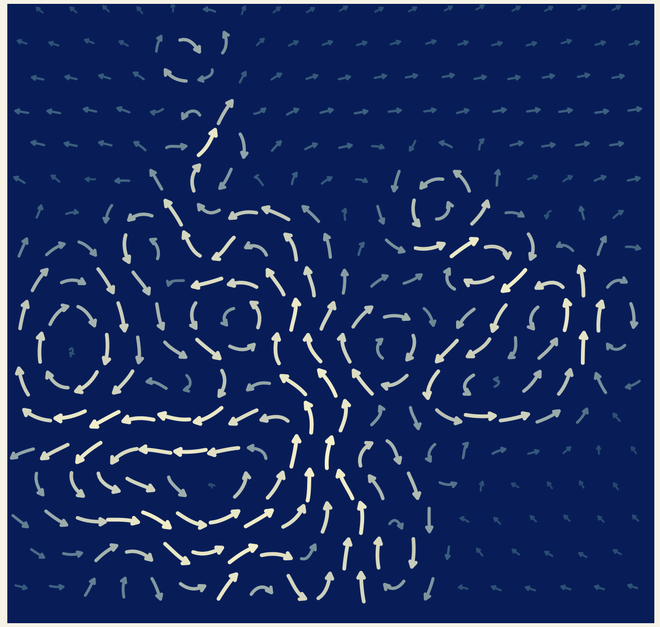
$$V = \nabla f + \nabla^T g + h$$

gradient | co-gradient | harmonic



fluid state

$\Longrightarrow$



velocity vector field

incompressible fluid

$\Longleftrightarrow$

divergence-free velocity field

$\Longleftrightarrow$

gradient component = 0

Mohammad Sina Nabizadeh, Stephanie Wang, Ravi Ramamoorthi, and Albert Chern. 2022. Covector fluids. ACM Trans. Graph. 41, 4, Article 113 (2022). https://doi.org/10.1145/3528223.3530120

# Application: shape deformation

$$V = \nabla f + \nabla^T g + h$$

gradient | co-gradient | harmonic



Curvature-Driven Conformal Deformations

ETIENNE CORMAN, Université de Lorraine, CNRS, Inria, LORIA, France

In this paper, we introduce a novel approach for computing conformal deformations of 3D shapes. Conformal deformations, which preserve angles and energies, serve as fundamental tools in geometry processing, essential for tasks such as surface fairing, deformation, and approximation using developable surfaces. Despite their importance, computing conformal maps, or embedding, especially for the latter, has been a challenging endeavor. The complexity arises from inherent numerical instabilities in curvature estimation and the intricate nature of differentiating these energies. To address these challenges, we concentrate on conformal deformations, leveraging the insight that they can be represented by a vector field. This vector field choice renders curvature-based energies easily applicable, mitigating previous manipulation difficulties. Our key contribution lies in identifying a linear relationship between infinitesimal conformal deformations and changes in curvature. We use this insight to deform surfaces of arbitrary genus, aiming to minimize bending energies or prescribe Gaussian curvature while sticking to positional constraints.

Additional Key Words and Phrases: conformal geometry, Willmore energy, Gaussian curvature



Fig. 1. Minimizing bending energy in Eq. (1) on spot for $p$ equal to 1 (top, Willmore flow), 2 (middle) and 3 (bottom).

1 INTRODUCTION

Curvature flows stand as indispensable tools in computer graphics for surface manipulation. They iteratively deform surfaces by minimizing an energy while complying with given constraints, such as area or volume preservation, positional constraints. They essentially define surfaces as the minimizers of specific energies. Our contribution is a novel approach for computing a specific category of curvature flows. Our focus revolves around two primary types of curvature-based flows in computer graphics: fairing flows, aimed at surface smoothing, and intrinsic flows, for which computing an embedded representation is often unnecessary.

The first category fairs a surface by iteratively molds surfaces to make it as round as possible by minimizing the surface bending. This is quantified by the magnitude of the shape operator $\mathbb{I}$, encapsulating the variations in surface normals across all directions. These techniques have found extensive applications in diverse domains, including surface fairing [Desbrun et al. 1999], geometric optimization [Hoskins et al. 2003; Ovsjanikov et al. 2003; Tournois et al. 1997], inpainting [Bobenko and Schröder 2005; Lévy 2005], and form finding [Jiang et al. 2015; Vouga et al. 2012]. Typically, these bending energies aim to decrease the norm of the shape operator [Eigen and Schulze 1997; Joshi and Séquin 2007; Lott and Pullin 1988; Moreton and Séquin 1992].

$$\mathcal{E}_b(\Gamma) = \int_\Gamma \|\mathbb{I}\|^p \, da, \quad p \in \mathbb{N}. \tag{1}$$

Author's address: Etienne Corman, etienne.corman@cnrs.fr, Université de Lorraine, CNRS, Inria, LORIA, Nancy, 54000, France.

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

2024-0187

May 2024.

When $p = 1$ this reduces to the celebrated Willmore energy [Joshi and Séquin 2007; Willmore 1965], however larger values of $p$ create a stronger smoothing effect on regions with higher curvature (see Figure 1). Crane et al. [2010; 2019] have also explored higher powers of curvature for surface fairing, yielding similar deformations. However, their method is confined to this very particular energy and is incompatible with intrinsic flows.

The second category: the intrinsic flows, is commonly used in the computer graphics community to "flatten" the surface not to map but from a parameterization point of view [Sawhney et al. 2017]. For instance, The Ricci flow, introduced by Hamilton [1988] can be easily utilized in growing the Poincaré conformal, falls within this category. Combinatorial counterparts of such flows [Chow and Luo 2003; Jin et al. 2008] have found successful applications in conformal mesh parameterization. In computer graphics, these flows naturally arise when computing texture atlases [Sheffer et al. 2006], remeshing quadrangulations [Ben-Chen et al. 2008; Kälberer et al. 2008], or surface approximation using developable patches for manufacturing [Yuan et al. 2023]. Here, the focus is on modifying the Gaussian curvature, i.e., the determinant of the shape operator. As this information is independent of the immersion, computations can be performed in the plane, followed by an integration operation [Springborn et al. 2008] — a focus of interest in this paper and illustrated by Figure 2. While an immersion can be computed from the deformed metric [Chern et al. 2018], this process results in the loss of the relationship between the deformation and the curvature, which is crucial for shape design tasks.

Considering only conformal deformations significantly enhances numerical stability while maintaining the capacity to represent a large family of deformations. Prior research has highlighted the potential for curvature flows to induce extreme surface deformations



represent local deformation as a vector field

need to be able to integrate up local deformations to recover full deformation

$\Longleftrightarrow$

gradient component = harmonic component = 0

Etienne Corman. 2024. Curvature-Driven Conformal Deformations.
ACM Trans. Graph. 43, 4, Article 139 (2024). https://doi.org/10.1145/3658145

# Application: knitting pattern design

$$V = \nabla f + \nabla^T g + h$$

gradient

co-gradient

harmonic


Curl Quantization for Automatic Placement of Knit Singularities

RAHUL MITRA*, Boston University, USA
MATTÉO COUPLET, Boston University, USA
TONGTONG WANG, LIGHTSPEED, China
MEGAN HOFFMANN, Northeastern University, USA
KUI WU, LIGHTSPEED, USA
EDWARD CHIEN, Boston University, USA



Fig. 1. *A knitting time function is computed over an input model, and the curl signal of its gradient is measured in the two orthogonal knitting directions, course and wale. Our method automatically places singularities in regions of high curl while satisfying all structural manufacturing constraints. The orthogonal stripe patterns are intertwined to generate a smooth knit graph suitable for both artifact-free yarn-level rendering and machine knitting.*

We develop a method for automatic placement of knit singularities based on curl quantization, extending the knit-planning frameworks of Mitra et al. (2019) and Wang et al. (2021). Given a surface, we first compute a vector field on an underlying knitting time function, and face course and wale singularities in regions of high curl for the normalized time function gradient and its 90° rotation. We show that this strategy allows us to easily maintain the structural constraints necessary for machine knitting, e.g., the helix-free constraint, and to satisfy user constraints such as stripe alignment and singularity placement. Our short preliminary approach alleviates the need for a manual cut-and-sew step in the design of seamless garments, as we demonstrate on a variety of examples.

CCS Concepts: • **Computing methodologies** → **Shape analysis**. • **Applied computing** → **Computer aided manufacturing**.

Additional Key Words and Phrases: Computational Knitting, Vector Fields

**ACM Reference Format:**
Rahul Mitra, Mattéo Couplet, Tongtong Wang, Megan Hoffmann, Kui Wu, and Edward Chien. 2025. Curl Quantization for Automatic Placement of Knit Singularities. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3721238.3730715

Authors' addresses: Rahul Mitra, rahulm@bu.edu, Boston University, Boston, MA, USA; Mattéo Couplet, mcouplet@bu.edu, Boston University, Boston, MA, USA; Tongtong Wang, wangtongt93@gmail.com, LIGHTSPEED, Shenzhen, China; Megan Hoffmann, hoffmann.me@northeastern.edu, Northeastern University, Boston, MA, USA; Kui Wu, wukui@lightspeed.com, LIGHTSPEED, Los Angeles, CA, USA; Edward Chien, echien@bu.edu, Boston University, Boston, MA, USA.

Please use nonwhite paper or ACM Reprint client to enable CC licensing.
This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
SIGGRAPH Conference Papers '25, August 10–14, 2025, Vancouver, BC, Canada.
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1540-2/25/08
https://doi.org/10.1145/3721238.3730715

## 1 INTRODUCTION

Knitting is a versatile additive manufacturing technique that creates garments by interlocking yarn loops through a CNC knitting machine (Computer Numerical Control knitting machine). It has been widely used in everyday applications, including clothing, home

the running of a singularity matching procedure in post-processing (Wolcin et al. 2020). Our global optimization also produces smooth knit graphs that are suitable for yarn-level rendering and machine knitting, without the visual artifacts of competing methods. Furthermore, we extend our method to the popular cut-and-sew garment design paradigm. We validate our method by computing singularity placements for a variety of manufacturable prototypical models in the 3D and cut-and-sew settings.

target knit stripes

generated knit pattern

*Idea:* represent pattern via vector field

final object shape
determined by field curl

result
(simulation)

result
(machine-knit)

Rahul Mitra, Mattéo Couplet, Tongtong Wang, Megan Hoffman, Kui Wu, and Edward Chien. 2025. Curl Quantization for Automatic Placement of Knit Singularities. (SIGGRAPH Conference Papers '25). Article 158, https://doi.org/10.1145/3721238.3730715

# Hodge decomposition is everywhere

Allows us to design algorithms adapted to the geometry and topology of our surfaces



$V$

$\nabla f$

gradient

$\nabla^T g$

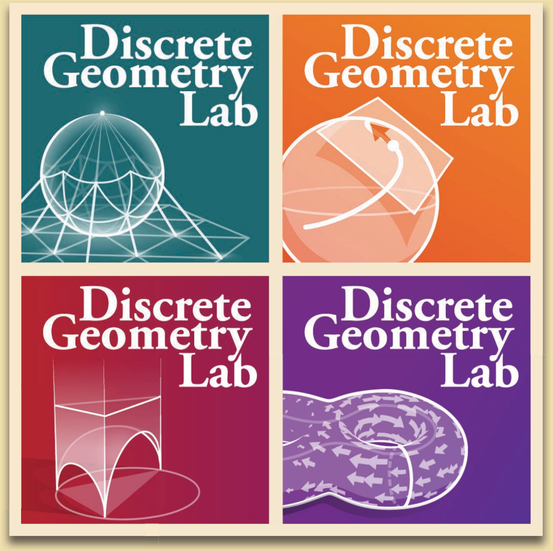
co-gradient

$h$

harmonic

# Thanks for listening

✉ mark.gillespie@utah.edu
🌐 geometry.cs.utah.edu

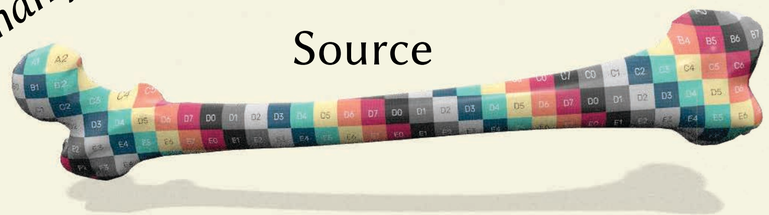





# Supplemental Slides

# vs Adaptive Triangulations
## [ Schmidt et al. 2023 ]

*human femur*




*bear*




*deer*




*pig*




Symmetric Dirichlet Energy

| | ours | AT |
|---|---|---|
| bear | 1.94 | 2.16 |
| deer | 2.02 | 2.22 |
| pig | 2.41 | 2.55 |

# Map Untangling

## Fixing Collapses



source mesh



ZoomOut
[Melzi et al. 2019]



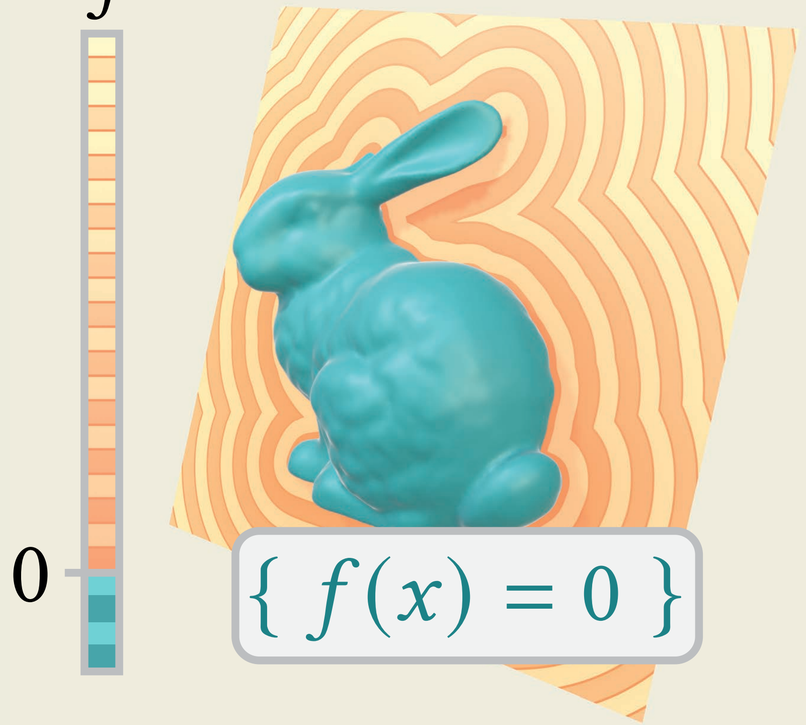
ours

# Implicit Encoding

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

$f$

$f : \mathbb{R}^3 \to \mathbb{R}$

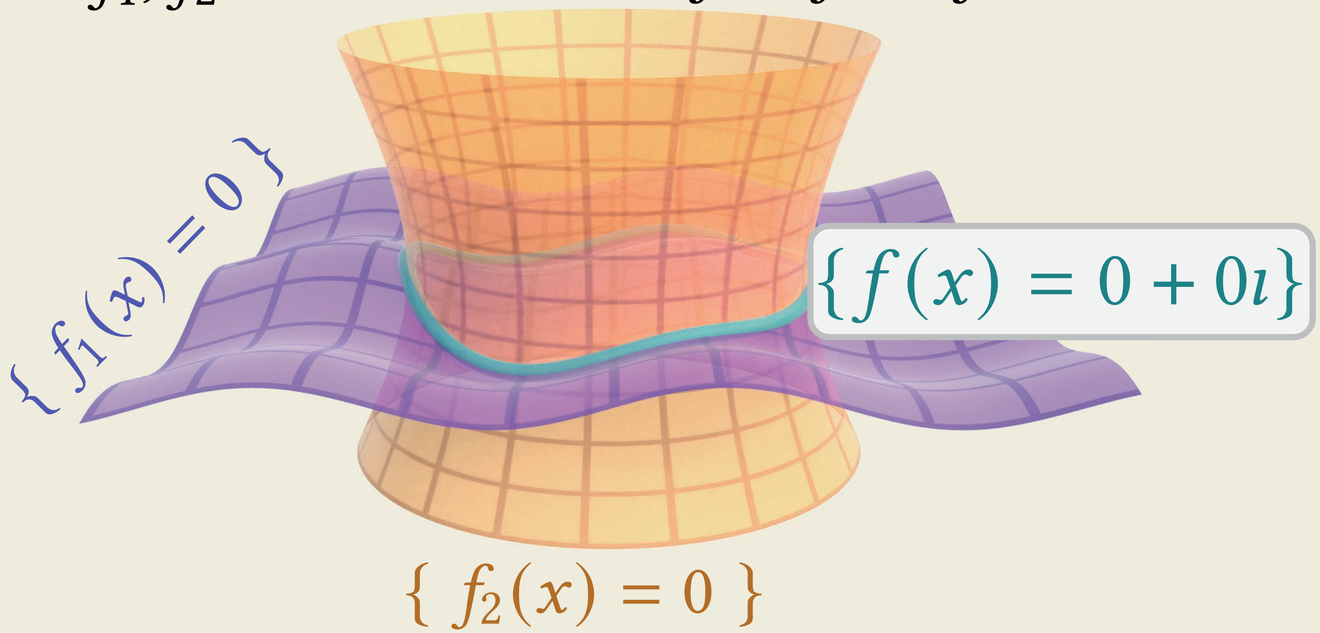



How about a 2D surface in a 4D space?

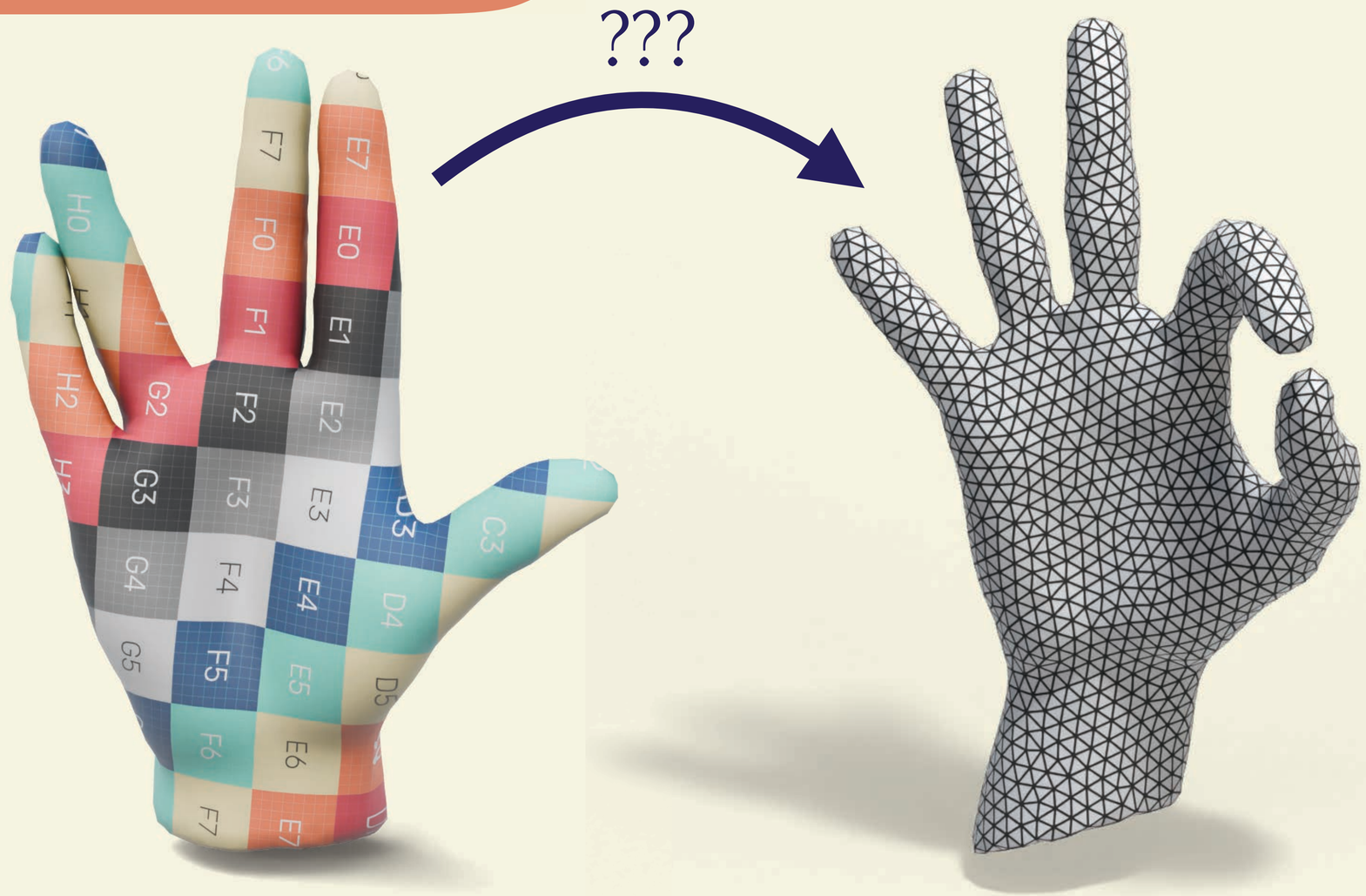
$\Longrightarrow$ encode as the zero level set of a *complex* function

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

$f_1, f_2 : \mathbb{R}^3 \to \mathbb{R} \iff f = f_1 + \imath f_2 : \mathbb{R}^3 \to \mathbb{C}$



# Algorithm Sketch

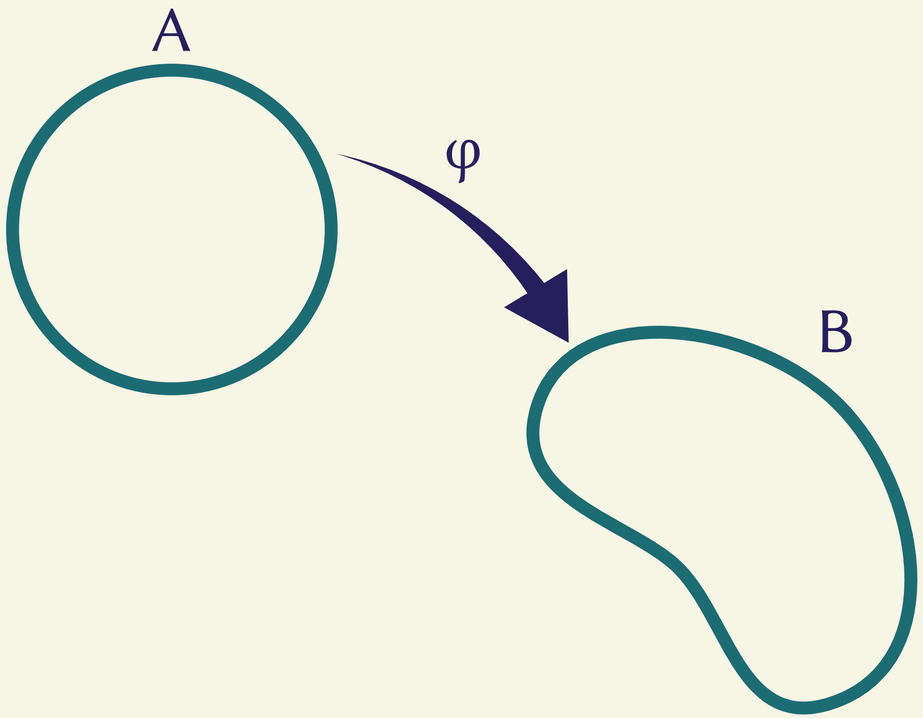


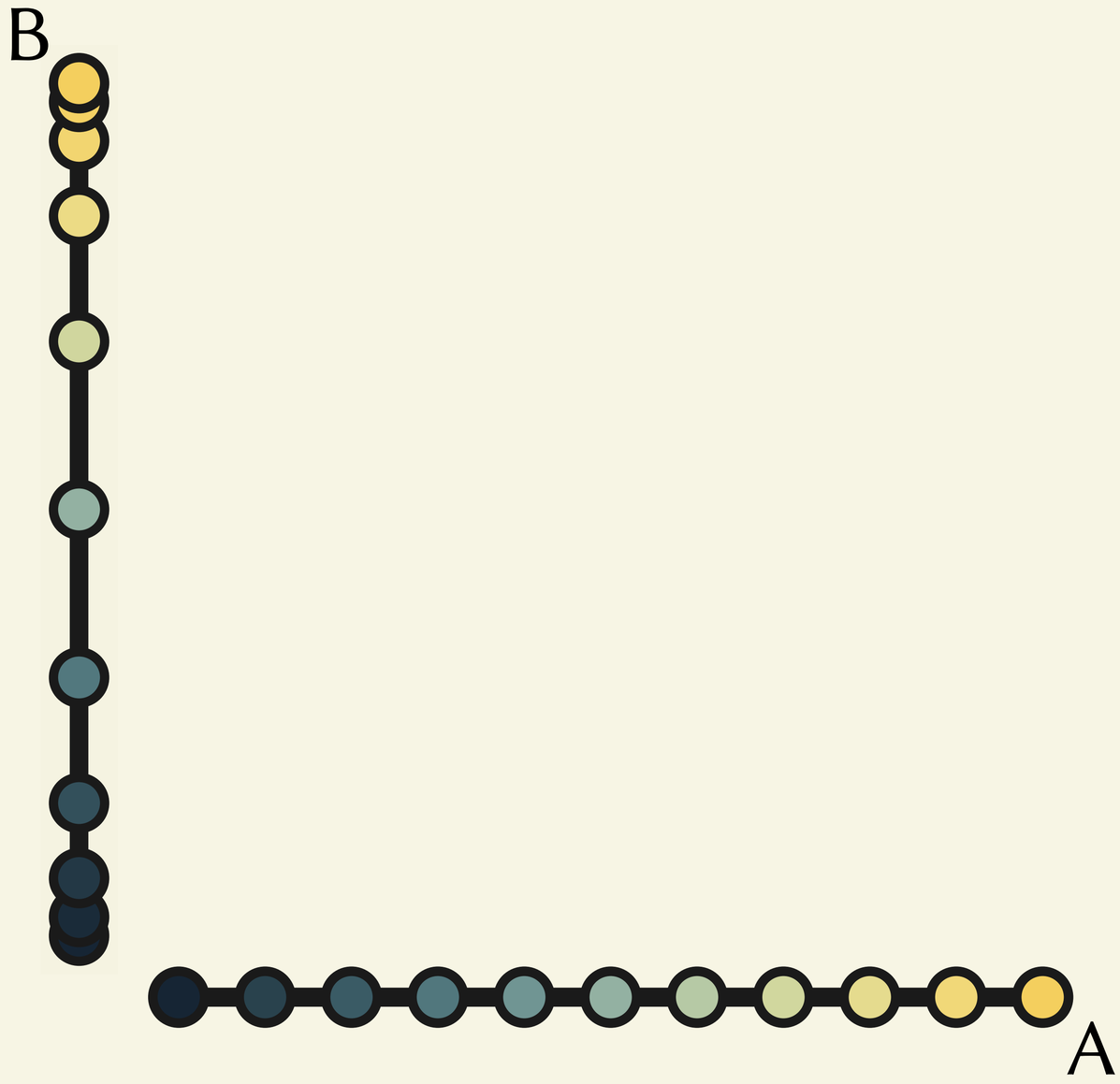
# Algorithm Sketch

I. Find An Initial Map  II. Fit Implicit Map  III. Optimize Implicit Map

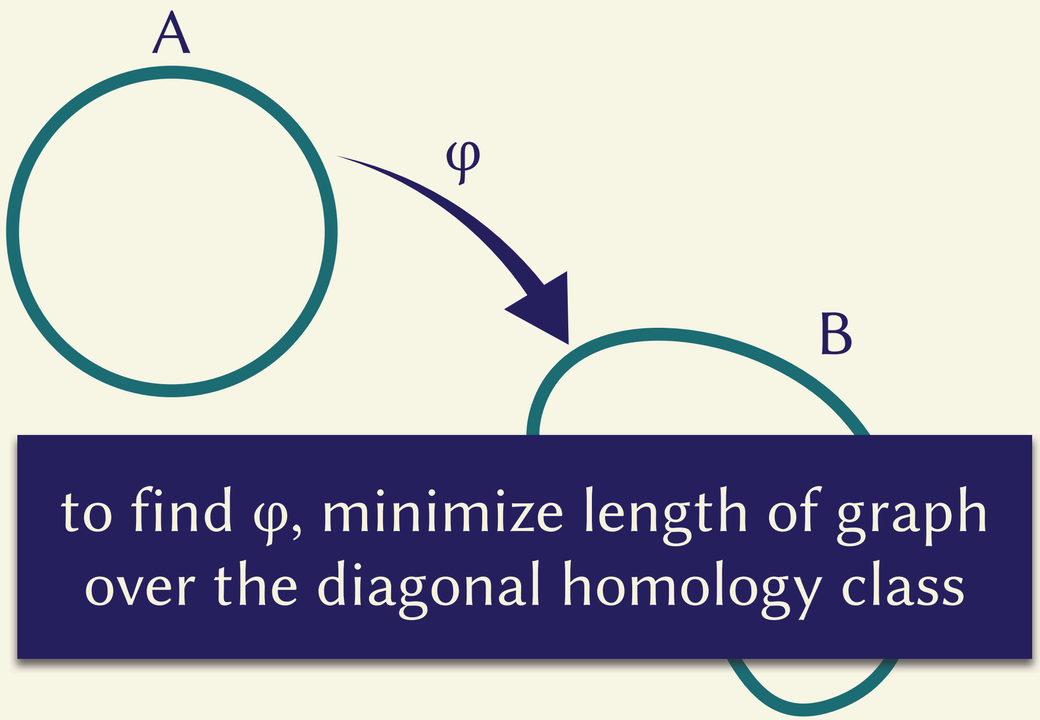


# Example:
# 1D Correspondence



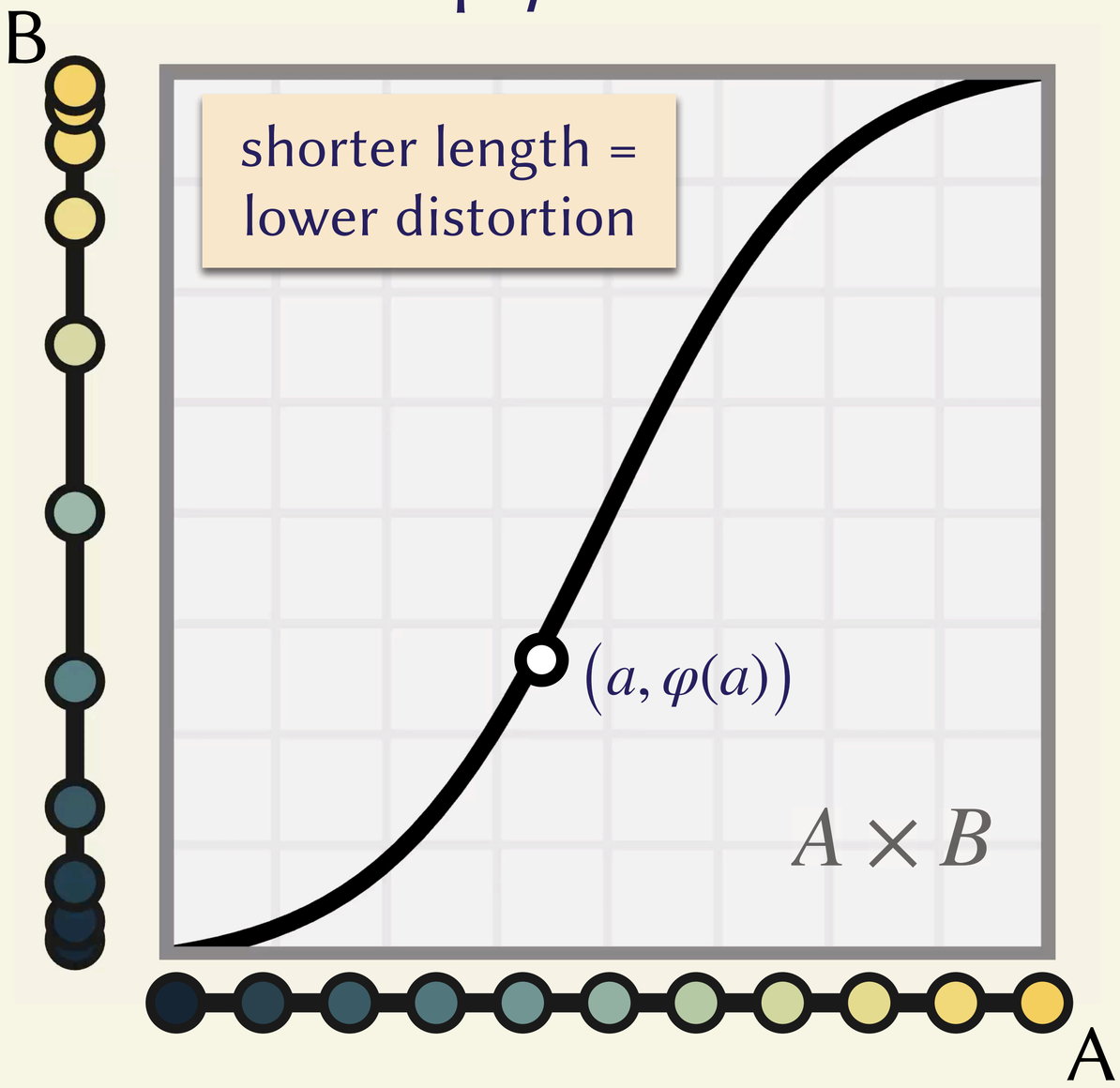



# Example:
# 1D Correspondence




to find φ, minimize length of graph over the diagonal homology class

map $\varphi : A \to B$

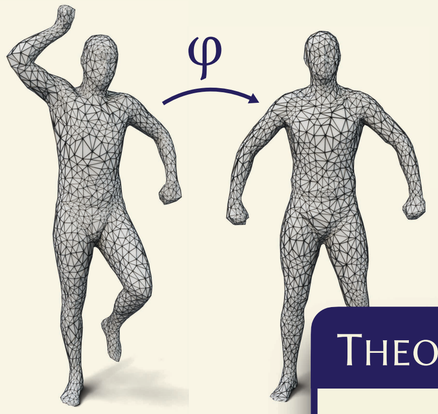


# Mapping between 2D surfaces

to find φ, minimize area of $\Sigma$ over the diagonal homology class $[\Delta] = [\mathrm{pt}_A \times B + A \times \mathrm{pt}_B]$

homology class of *orientation-preserving* maps!

$$\varphi : A \to B \qquad \Sigma_\varphi = \{(a, \varphi(a))\} \subset A \times B$$




$$\mathrm{Area}[\Sigma_\varphi] = \int_A \sqrt{(1+\sigma_1^2)(1+\sigma_2^2)}\, dA$$

($\sigma_1, \sigma_2$ singular values of $d\varphi$)

THEOREM [CSMG 2026]:

$$\mathrm{Area}[A] + \int_A \underbrace{|\det d\varphi|}_{\text{AREA DISTORTION}} dA \le \mathrm{Area}[\Sigma_\varphi] \le \mathrm{Area}[A] + \int_A \underbrace{\tfrac{1}{2}|d\varphi|^2}_{\text{DIRICHLET ENERGY}} dA$$

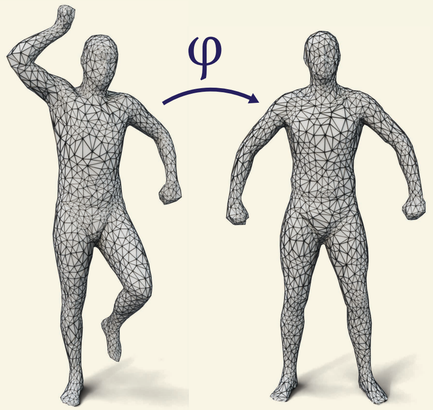
# Mapping between 2D surfaces

to find φ, minimize area of $\Sigma$ over the diagonal homology class
$[\Delta] = [\mathrm{pt}_A \times B + A \times \mathrm{pt}_B]$

homology class of *orientation-preserving* maps!

$\varphi : A \to B$

$$\Sigma_\varphi = \{(a, \varphi(a))\} \subset A \times B$$




What does $\mathrm{Area}[\Sigma_\varphi]$ measure?

$$\mathrm{Area}[\Sigma_\varphi] = \mathrm{Area}\left[(\mathrm{Id} \times \varphi)(A)\right]$$

$$= \int_A \sqrt{\det(\mathbb{I} + g_A^{-1}\varphi^* g_B)}\, \mathrm{d}A$$

$$= \int_A \sqrt{\left(1+\sigma_1^2\right)\left(1+\sigma_2^2\right)}\, \mathrm{d}A$$

($\sigma_1, \sigma_2$ singular values of $d\varphi$)