252-0538-00L, Spring 2018

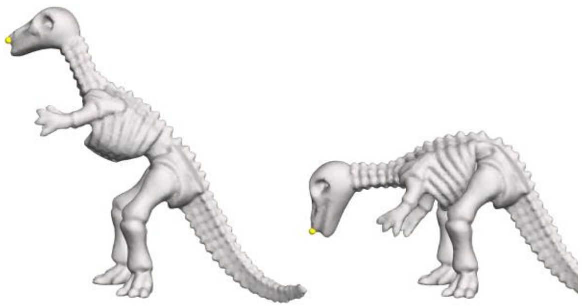
Shape Modeling and Geometry Processing

Mappings

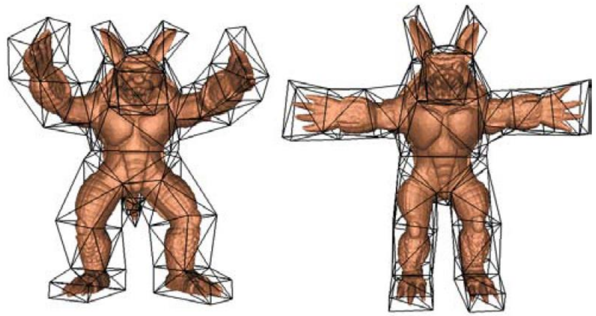
Representation and Distortion

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Mappings
Representation and Distortion

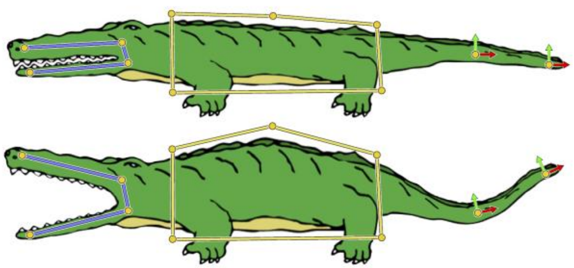


[Sorkine & Alexa 07]



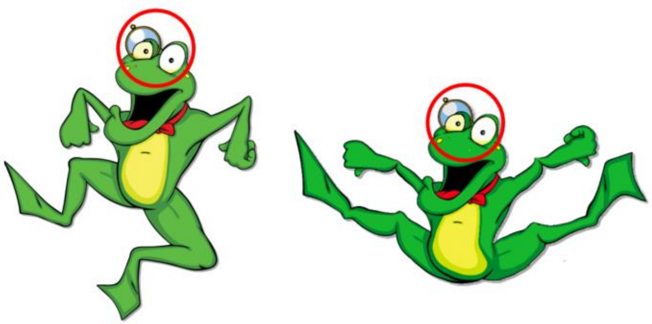
[Ju et al. 05]

[Poranne & Lipman 14]



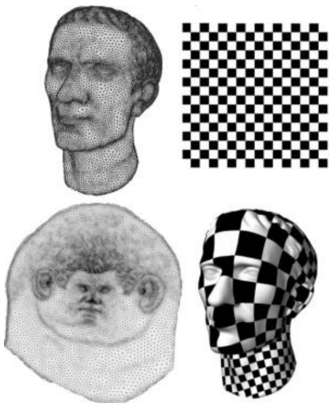
[Jacobson 07]

[Weber et al. 09]

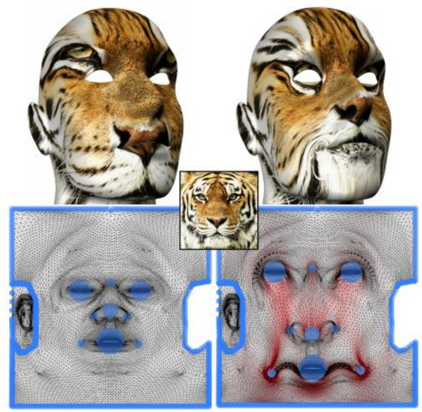


[Lipman et al. 07]

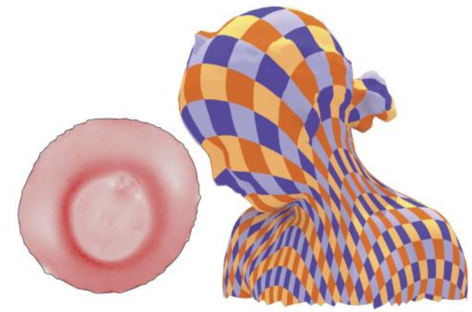
Mappings
Representation and Distortion



[Lévy et al. 02]

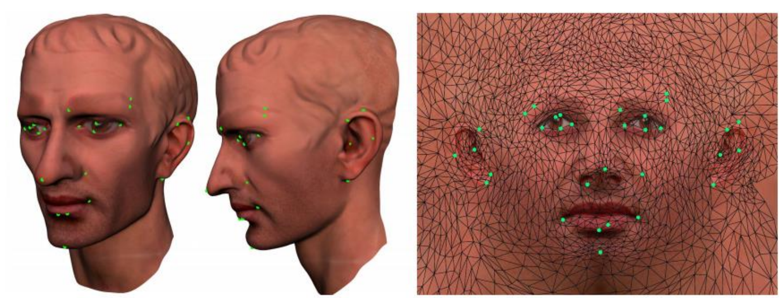


[Schuler et al. 13]



[Fu et al. 15]

[Mullen et al. 08]



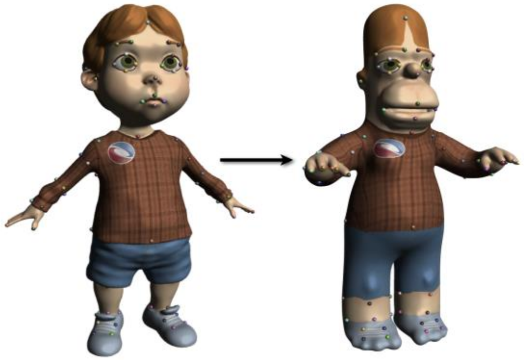
[Weber et al. 12]

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

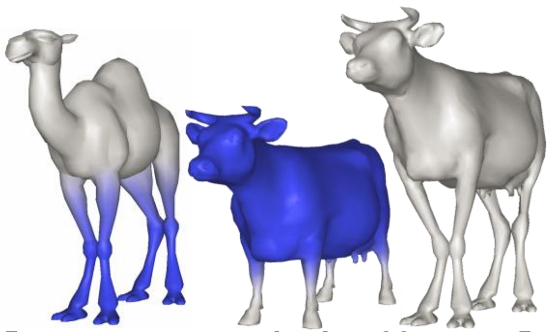
Mappings
Representation and Distortion

[Kim et al. 11]

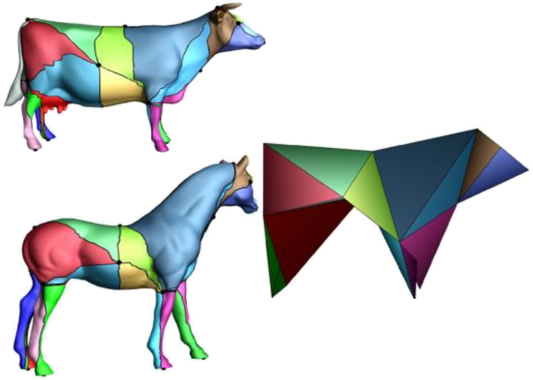
[Ovsjanikov et al. 12]



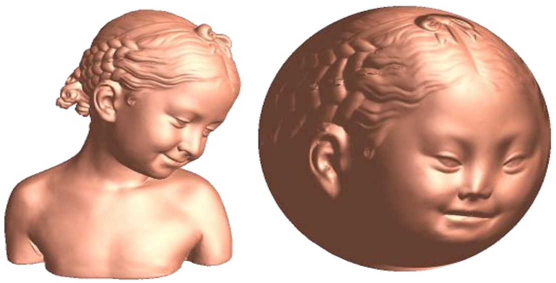
[Panozzo et al. 13]



[Kraevoy and Sheffer 04]



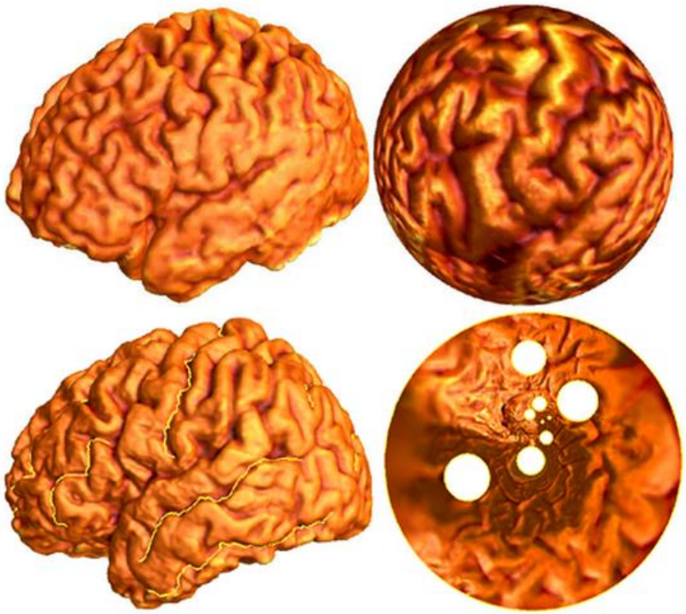
[Schreiner et al. 04]

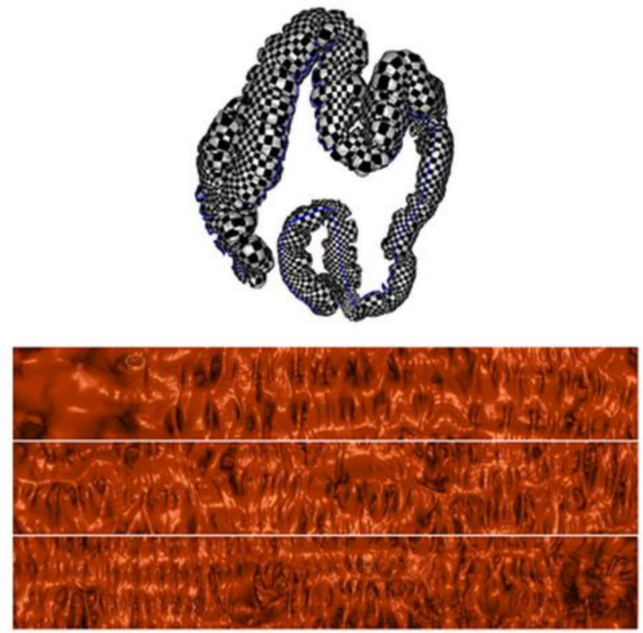


[Jin et al. 08]

“Real” Applications

Brain/colon mapping

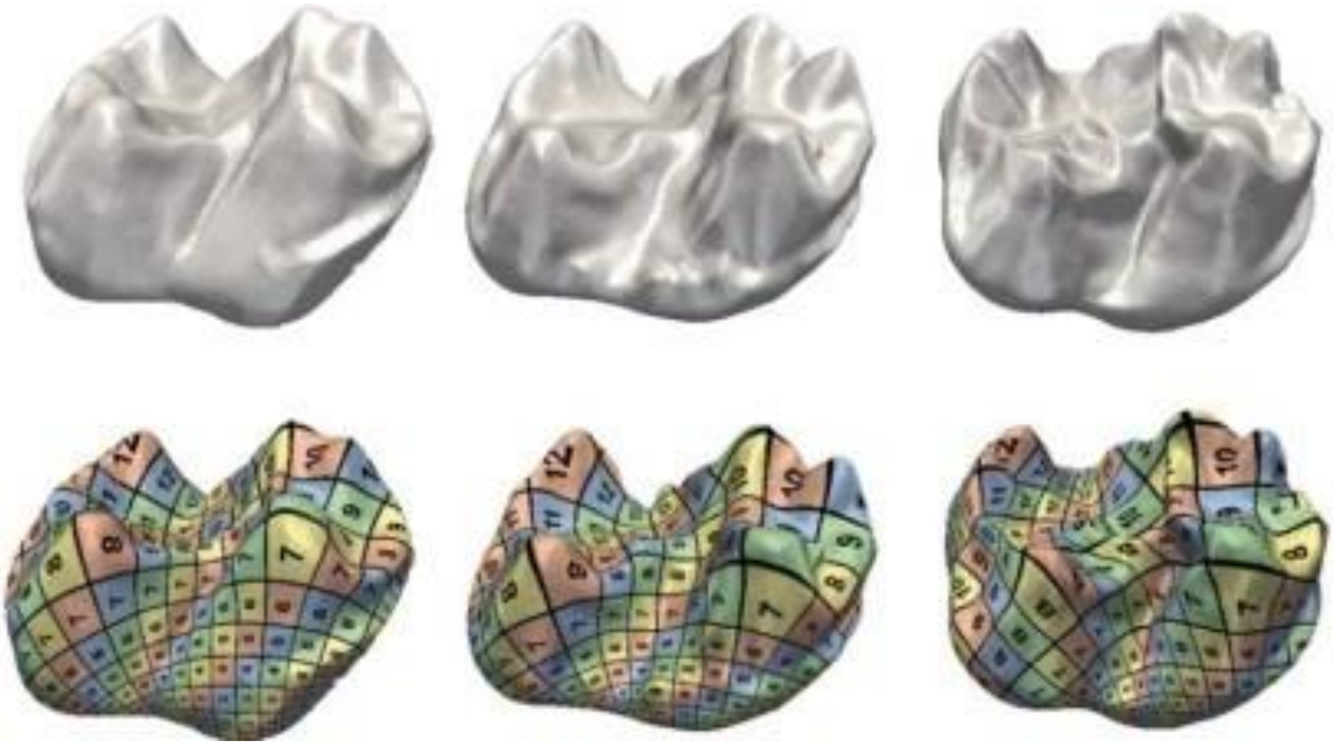




[Gu et al.]

"Real" Applications

Biological Morphology

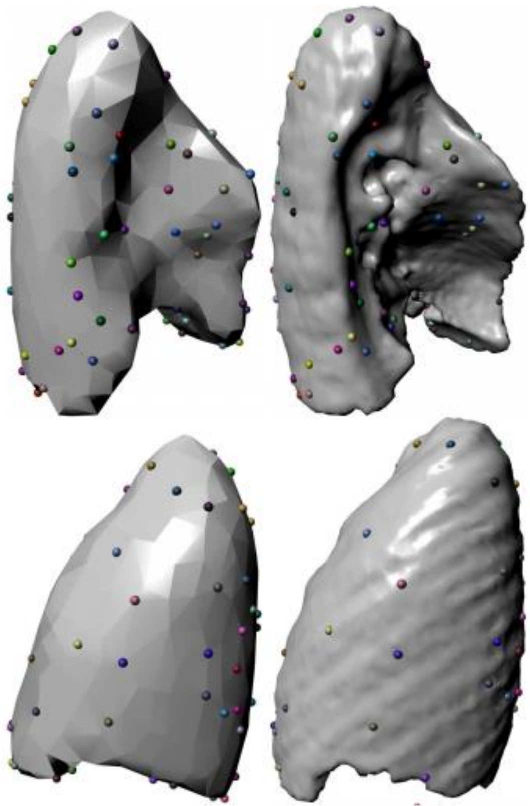


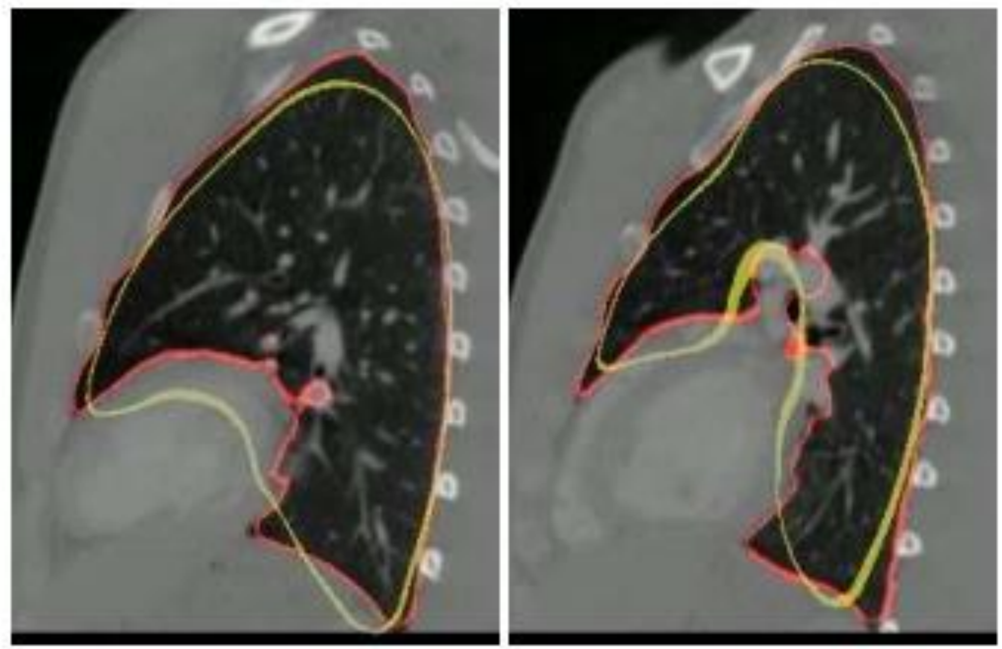
[Boyer et al. 2012]

ETH zürich

“Real” Applications

Medical segmentation/registration





[Levi and Gotsman 12]

Definition

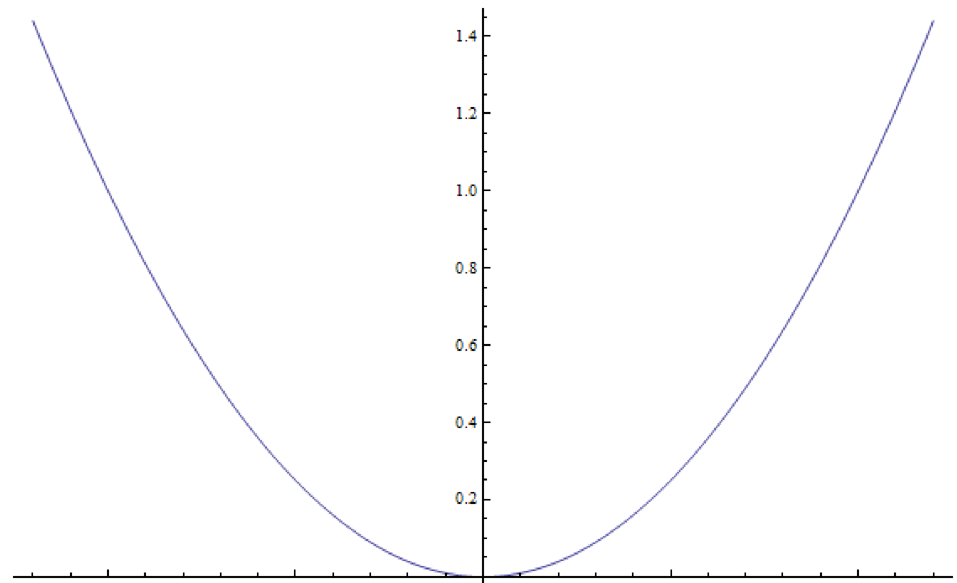
Mapping / Map :

A smooth function between shapes / spaces

Examples

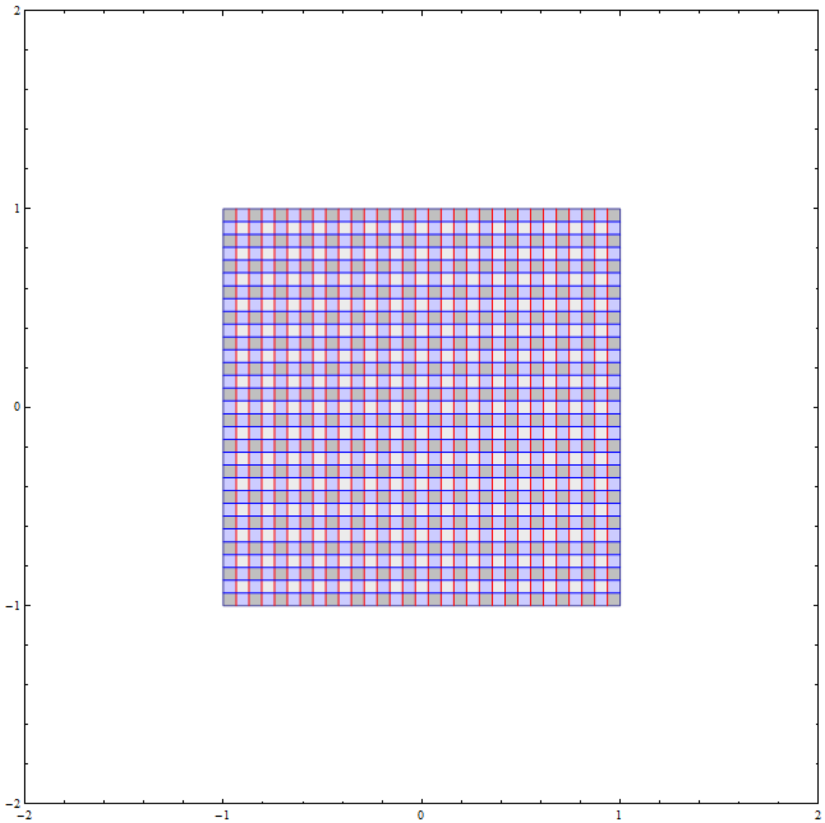
$$f : \mathbb{R} \to \mathbb{R}$$

$$f(x) = x^2$$

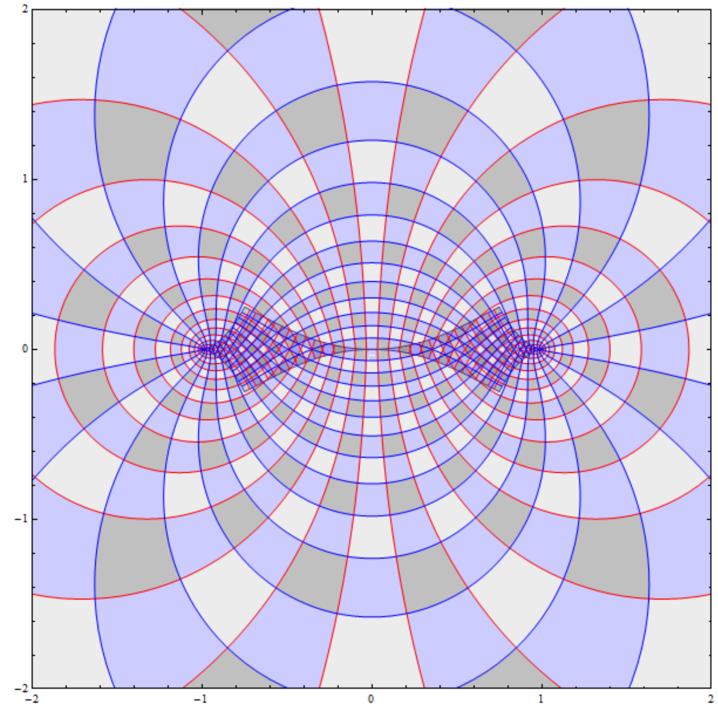


$$f : \mathbb{R}^2 \to \mathbb{R}^2$$

$$f(z) = z + \frac{1}{z}$$

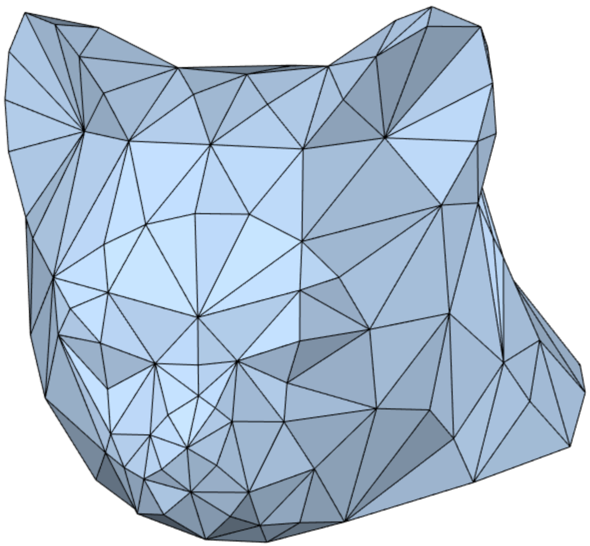


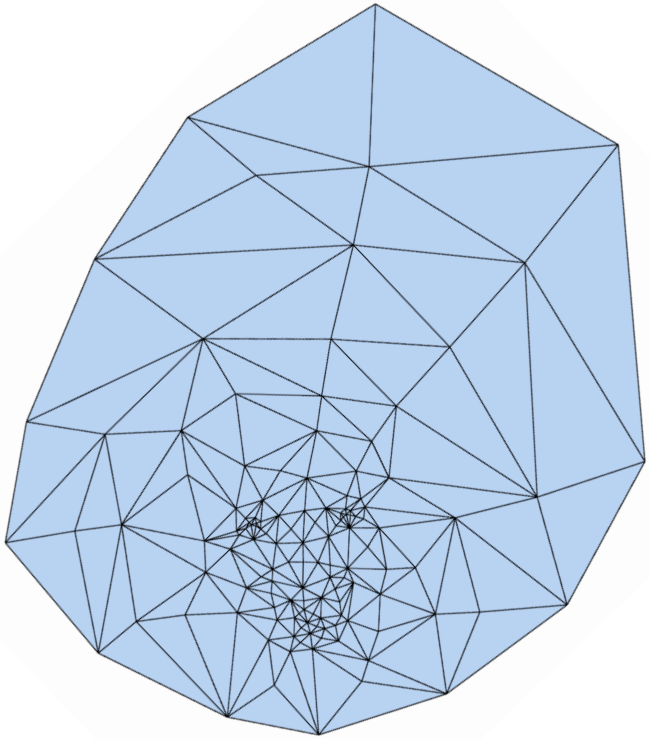
$\Rightarrow$



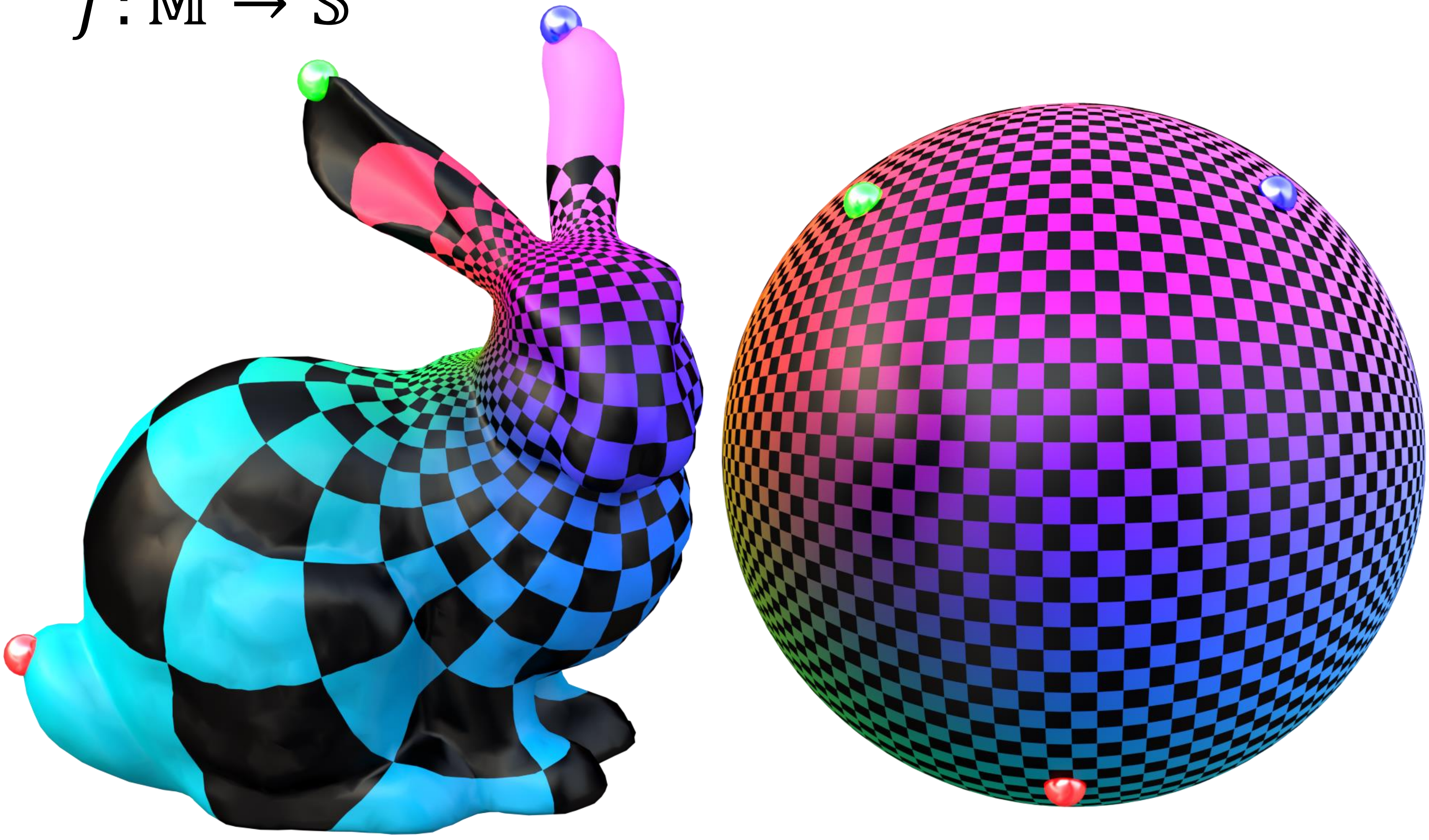
ETH zürich

$$f : \mathbb{M} \rightarrow \mathbb{R}^2$$





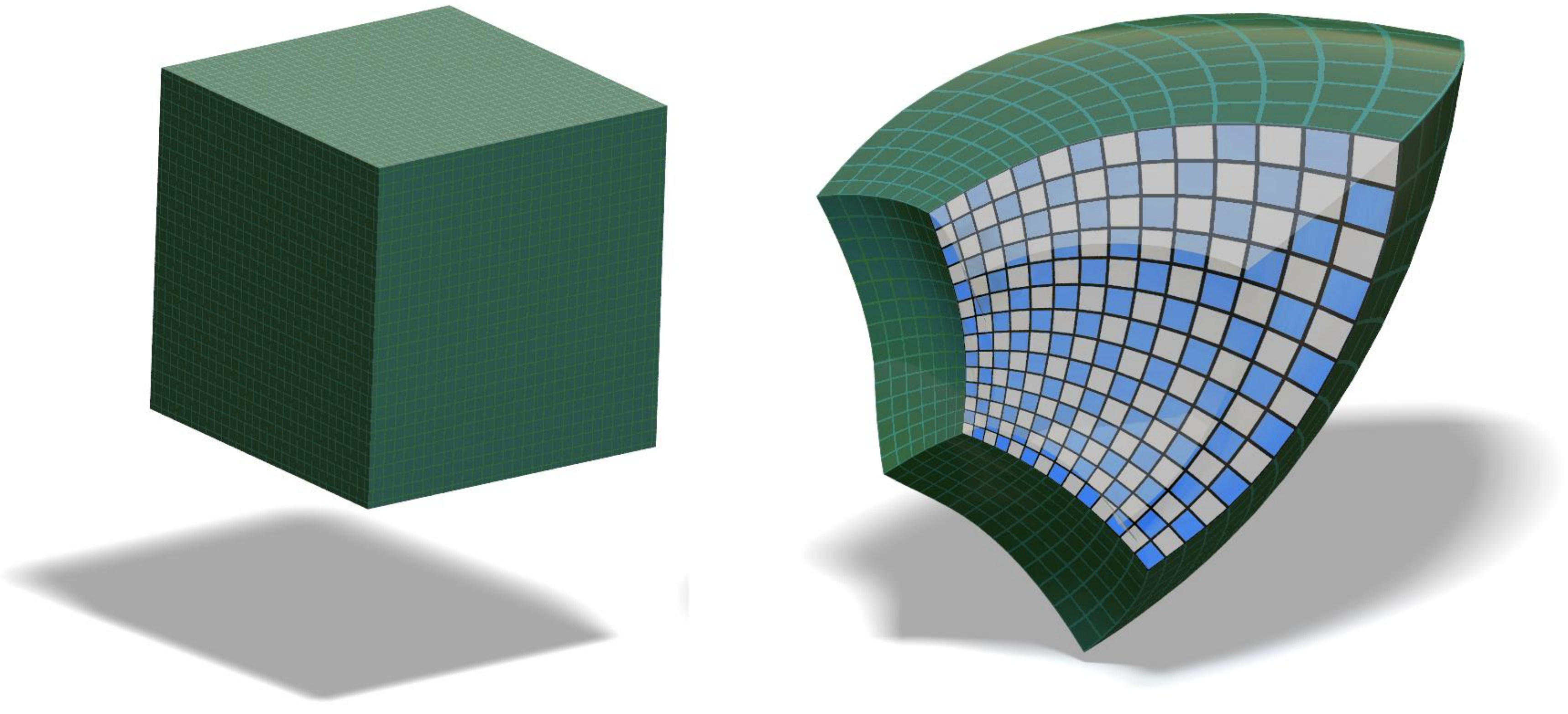
$f: \mathbb{M} \to \mathbb{S}^2$



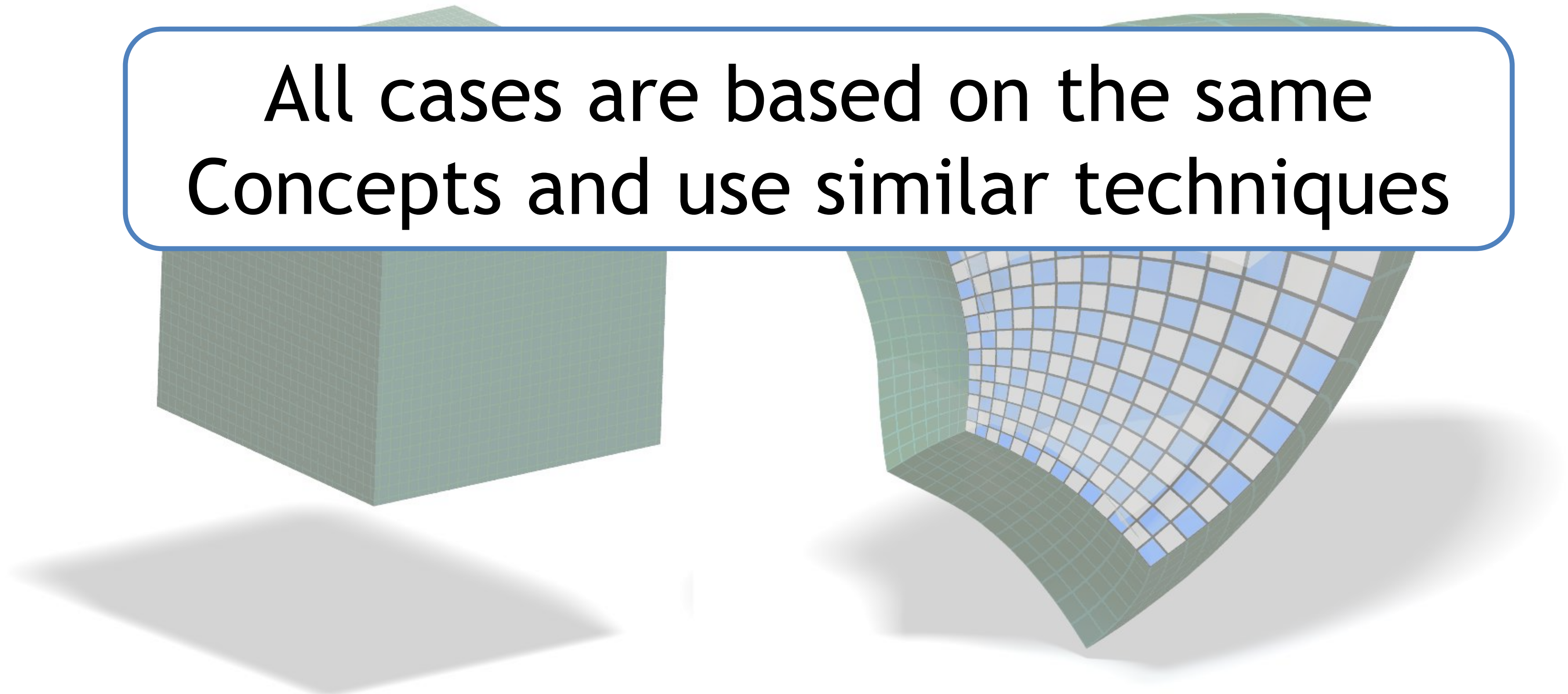
ETH zürich

$$f: \mathbb{M} \rightarrow \mathbb{M}'$$

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



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



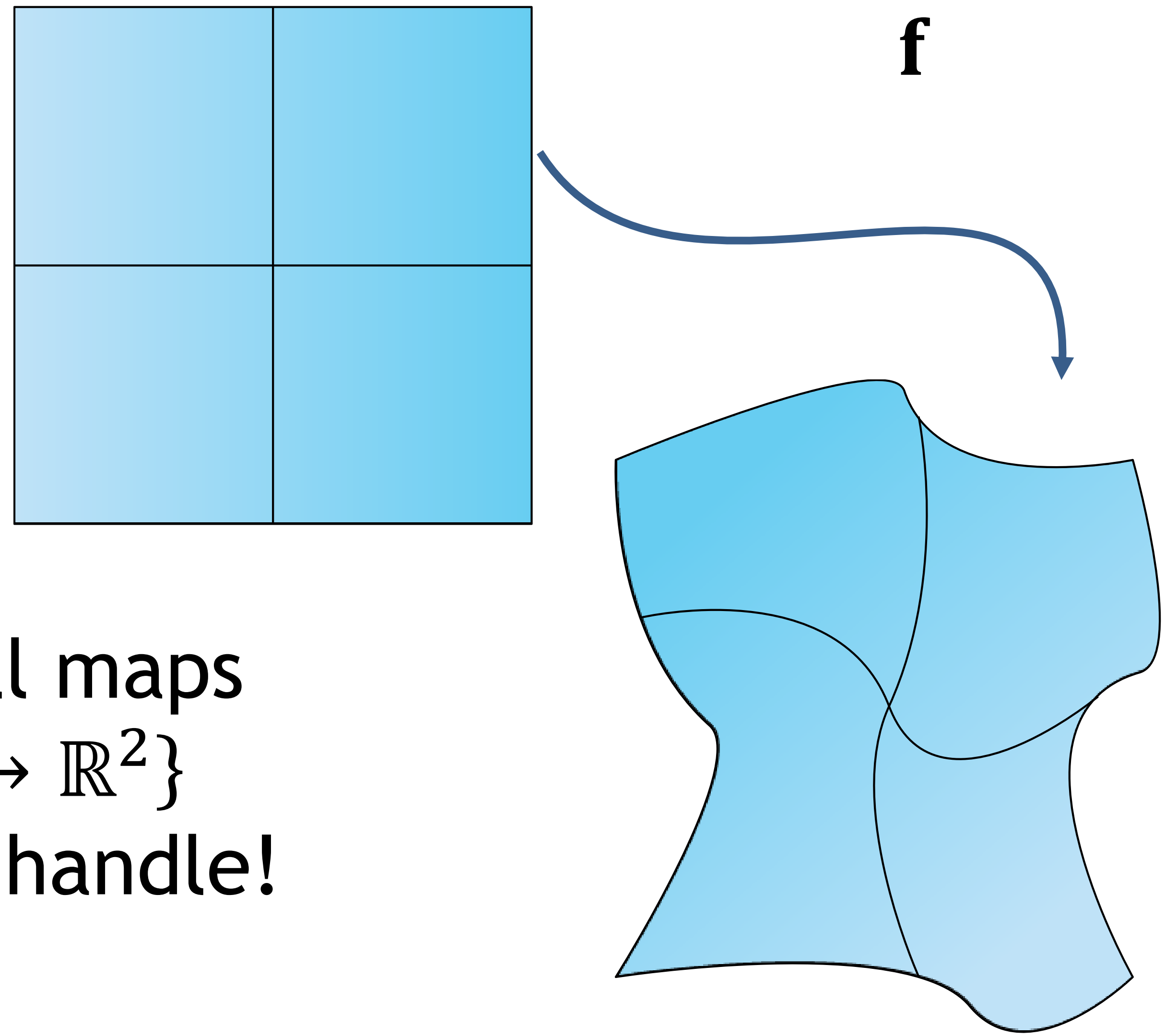
All cases are based on the same Concepts and use similar techniques

2D Maps : Representation

$$\boxed{\mathbf{f}(\mathbf{x}) = \begin{pmatrix} u(\mathbf{x}) \\ v(\mathbf{x}) \end{pmatrix}}$$

2D Maps : Representation

$$\boxed{\mathbf{f}(\mathbf{x}) = \begin{pmatrix} u(\mathbf{x}) \\ v(\mathbf{x}) \end{pmatrix}}$$



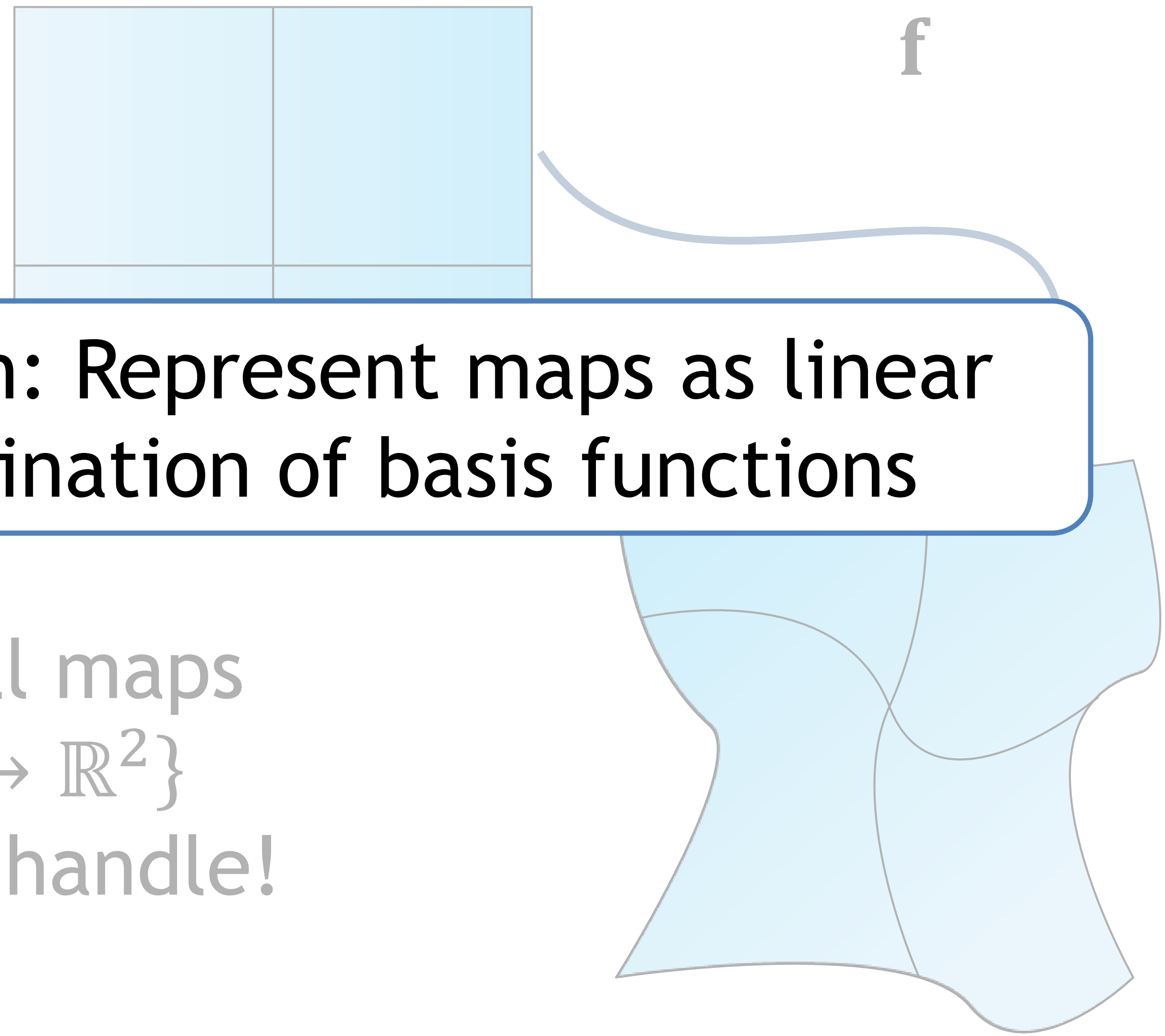
The set of all maps

$\{f(x): \mathbb{R}^2 \to \mathbb{R}^2\}$

is too big to handle!

2D Maps : Representation

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} u(\mathbf{x}) \\ v(\mathbf{x}) \end{pmatrix}$$



Solution: Represent maps as linear combination of basis functions

The set of all maps
$\{f(x)\colon \mathbb{R}^2 \to \mathbb{R}^2\}$
is too big to handle!

ETH zürich

2D Maps : Representation

$$f_1, f_2, f_3, \ldots, f_n$$

2D Maps : Representation

$$f_1, f_2, f_3, \dots, f_n$$

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} u(\mathbf{x}) \\ v(\mathbf{x}) \end{pmatrix}$$

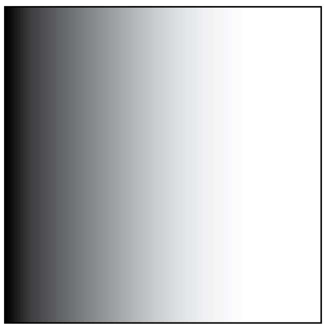
2D Maps : Representation

$$f_1, f_2, f_3, \dots, f_n$$

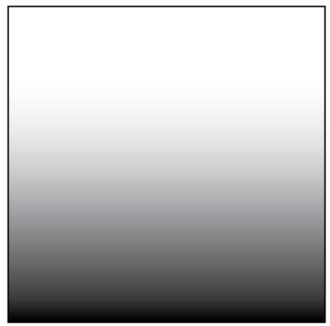
$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} u(\mathbf{x}) \\ v(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \sum a_i f_i(\mathbf{x}) \\ \sum b_i f_i(\mathbf{x}) \end{pmatrix}$$

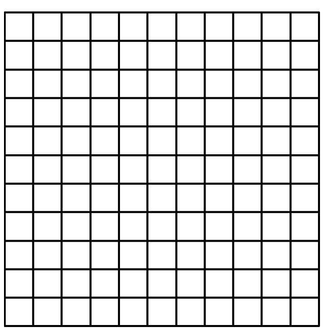
2D Maps : Representation

$f_1(x, y) = x$

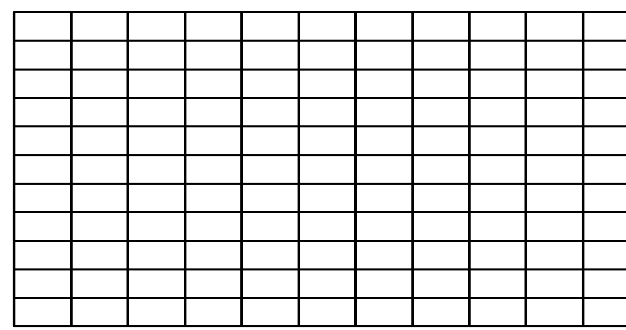


$f_2(x, y) = y$



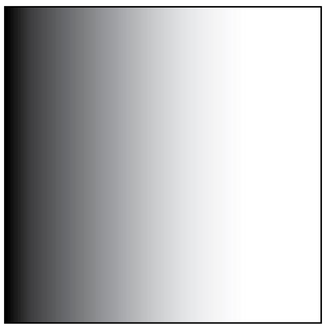


$(x, y) \Rightarrow (2f_1, f_2)$

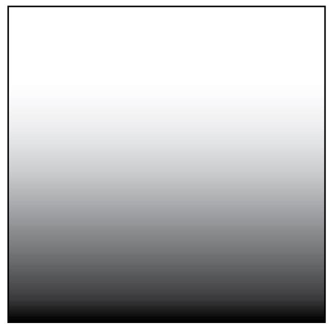


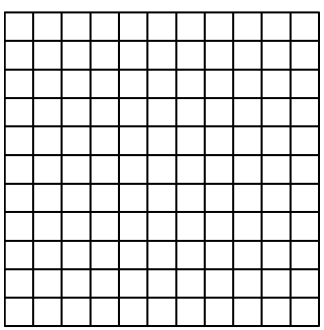
2D Maps : Representation

$f_1(x, y) = x$

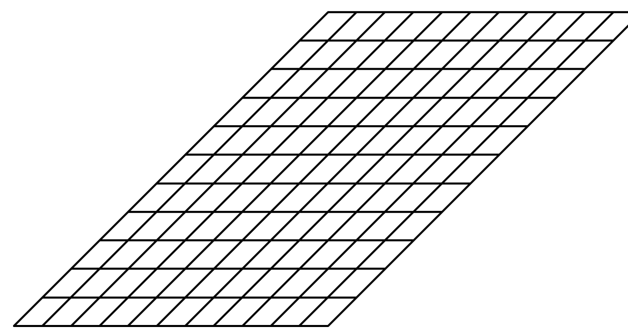


$f_2(x, y) = y$



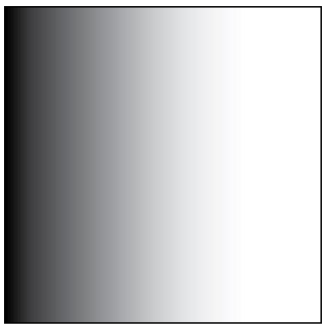
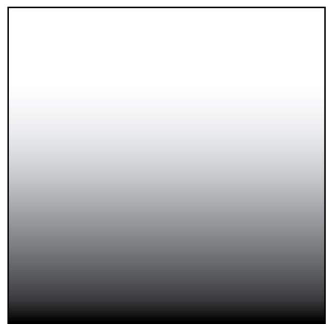
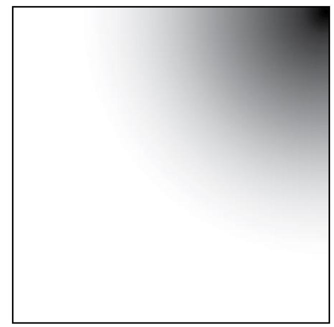


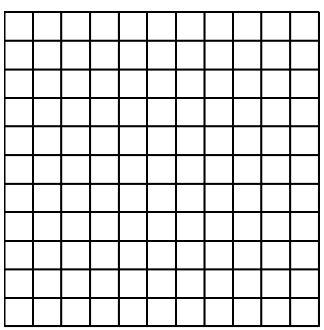
$(x, y) \Rightarrow (f_1 + f_2, f_2)$



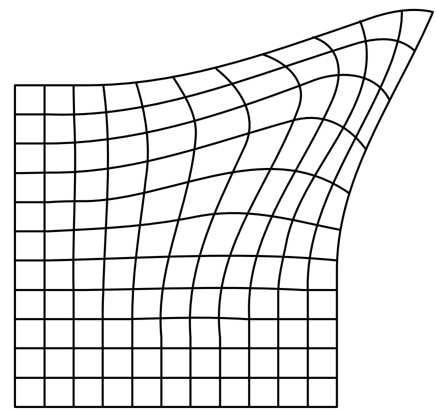
2D Maps : Representation

$f_1(x, y) = x$ $\quad$ $f_2(x, y) = y$ $\quad$ $f_3(x, y)$



$$(x, y) \Rightarrow (f_1 + f_3, f_2 + f_3)$$



Mappings for deformations

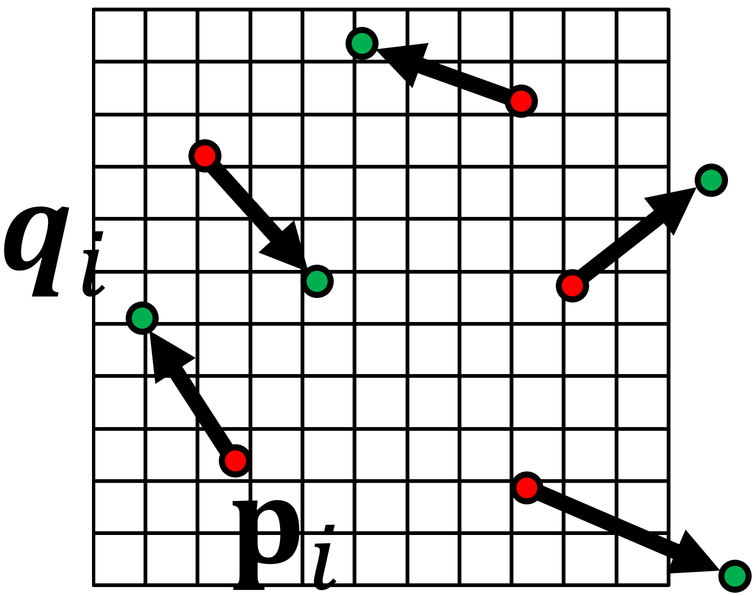
$\mathbf{q}_i$

$\mathbf{p}_i$

Mappings for deformations

$\mathbf{q}_i$

Deformation as an interpolation problem



$$\mathbf{f}(\mathbf{p}_i) = \sum \mathbf{c}_i f_i(\mathbf{x})$$

$$\mathbf{c}_i = ?$$

$$\mathbf{f}(\mathbf{p}_i) = \mathbf{q}_i, \forall i$$

$$\sum \mathbf{c}_i f_i(\mathbf{p}_i) = \mathbf{q}_i, \forall i$$

Example: Thin Plate Spline

Solve the problem

$$\min E_{\mathrm{TPS}}(\mathbf{f}) = \iint \left[\left(\frac{\partial^2 \mathbf{f}}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 \mathbf{f}}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 \mathbf{f}}{\partial y^2} \right)^2 \right]$$

Bending energy

$$\text{s.t. } \mathbf{f}(\mathbf{p}_i) = \mathbf{q}_i, \forall i$$

General solution

$$\mathbf{f}(\mathbf{p}_i) = \mathbf{c}_0 + \mathbf{c}_x \boldsymbol{x} + \mathbf{c}_y \boldsymbol{y} + \sum \mathbf{c}_i \phi(\|\mathbf{x} - \mathbf{p}_i\|)$$

$$\phi(r) = r^2 \log r$$

ETH zürich

Globally VS. Locally Bijective

Globally Bijective $\rightarrow$ Locally Bijective

$\mathbf{f}$ is bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective

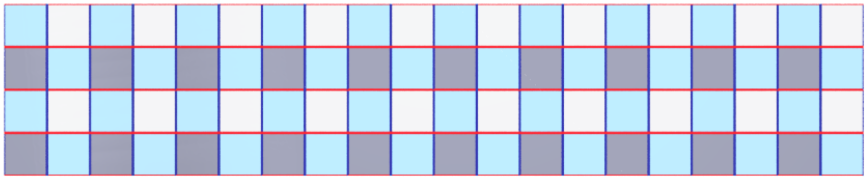
Globally VS. Locally Bijective

Globally Bijective

$\mathbf{f}$ is bijective

Locally Bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective



ETH zürich

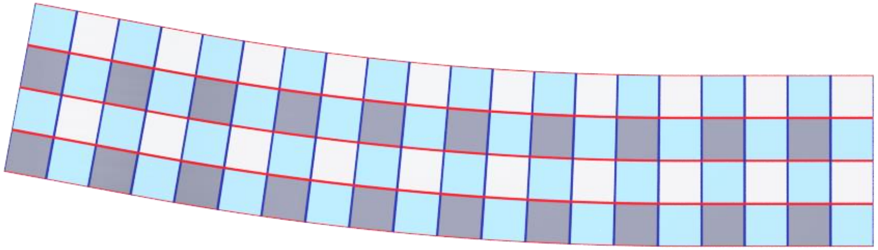
Globally VS. Locally Bijective

Globally
Bijective

$\mathbf{f}$ is bijective

Locally
Bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective



ETH zürich

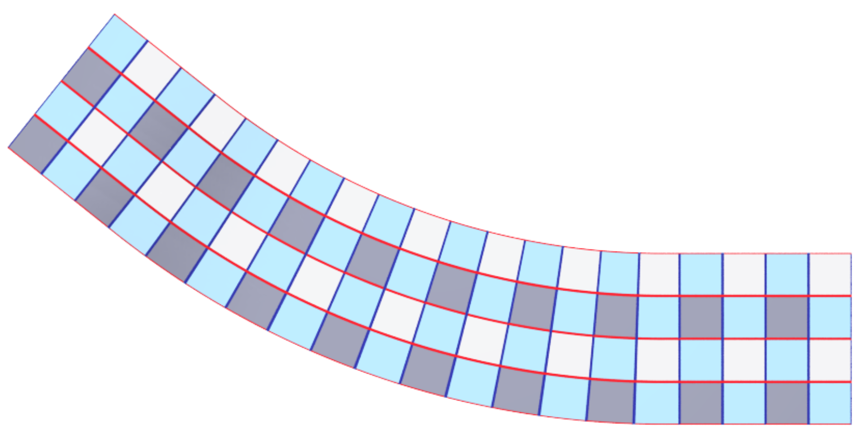
Globally VS. Locally Bijective

Globally Bijective

$\mathbf{f}$ is bijective

Locally Bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective



ETH zürich

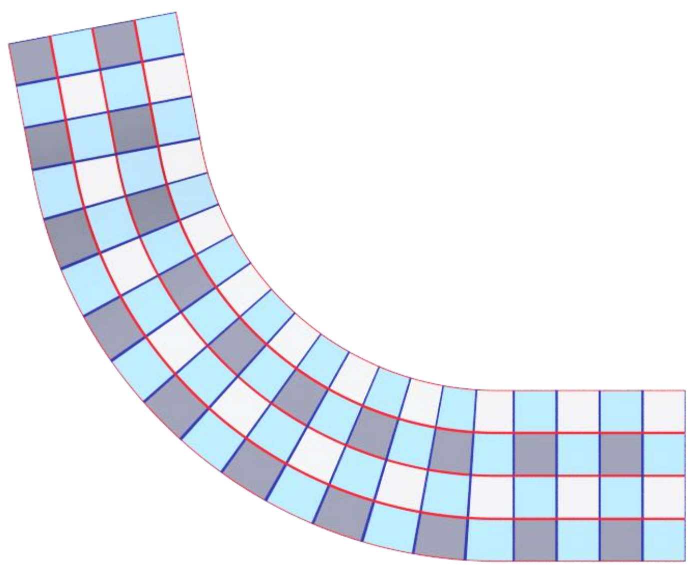
Globally VS. Locally Bijective

Globally Bijective

$\mathbf{f}$ is bijective

Locally Bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective



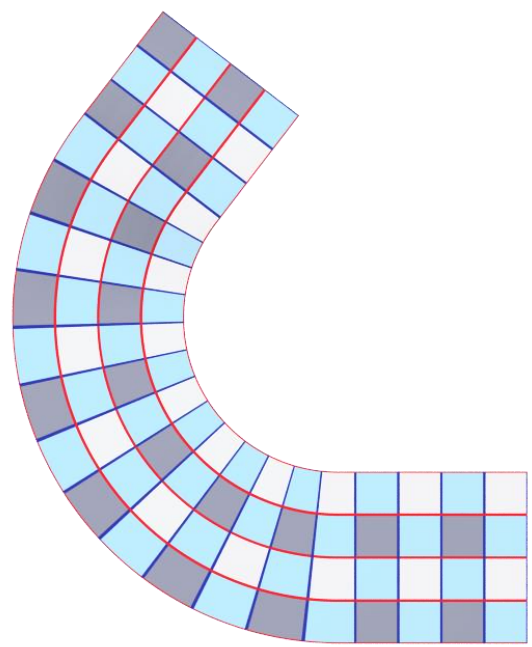
Globally VS. Locally Bijective

Globally Bijective

$\mathbf{f}$ is bijective

Locally Bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective



Globally VS. Locally Bijective

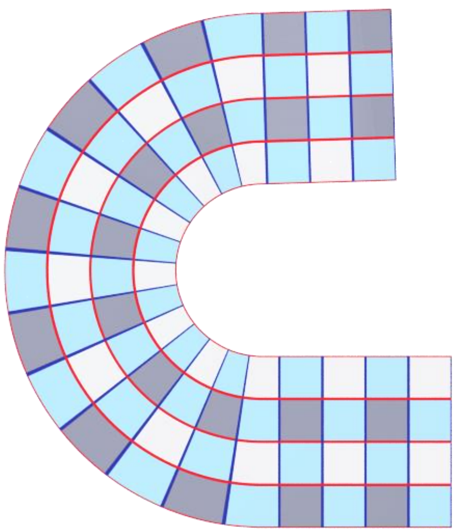
Globally Bijective

Locally Bijective

$\mathbf{f}$ is bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective

Still Bijective!



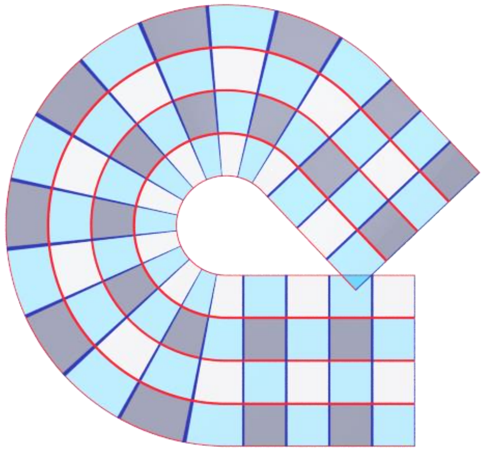
Globally VS. Locally Bijective

Globally Bijective

Locally Bijective

$\mathbf{f}$ is bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective



ETH zürich

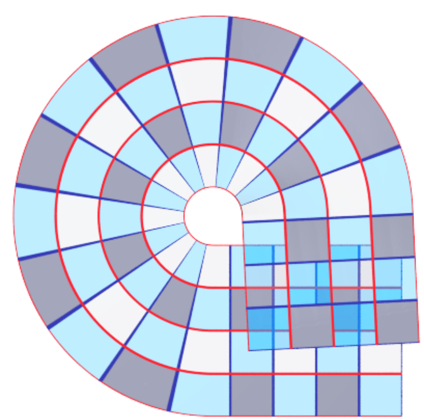
Globally VS. Locally Bijective

Globally Bijective

$\mathbf{f}$ is bijective

Locally Bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective



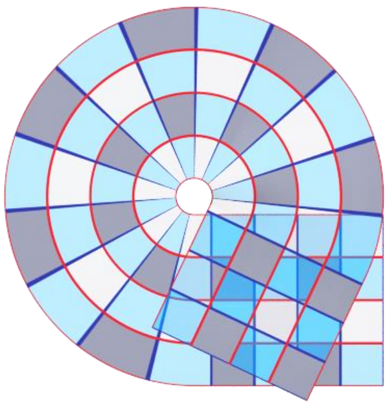
Globally VS. Locally Bijective

Globally Bijective

$\mathbf{f}$ is bijective

Locally Bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective



ETH zürich

Globally VS. Locally Bijective

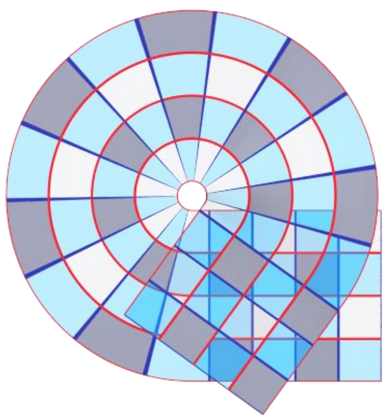
Globally Bijective

$\mathbf{f}$ is bijective

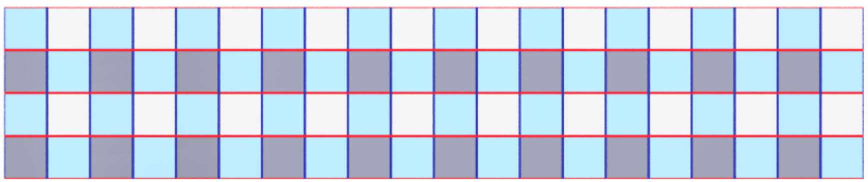
Not Bijective!

Locally Bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective

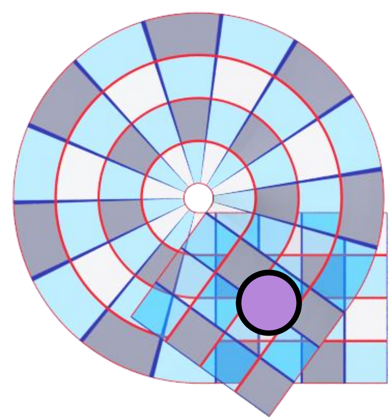


Globally VS. Locally Bijective



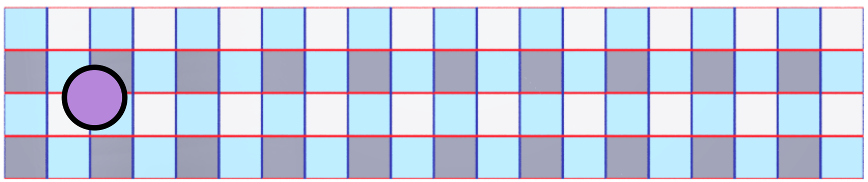
Two Pre-images

Not Bijective!

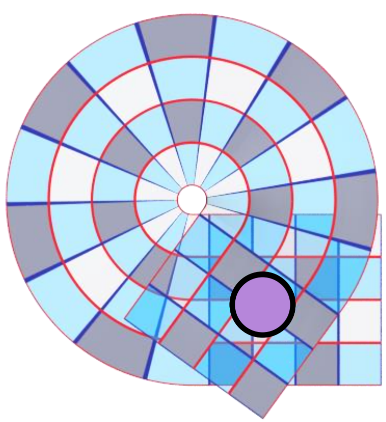


ETH zürich

Globally Bijective VS. Locally Bijective

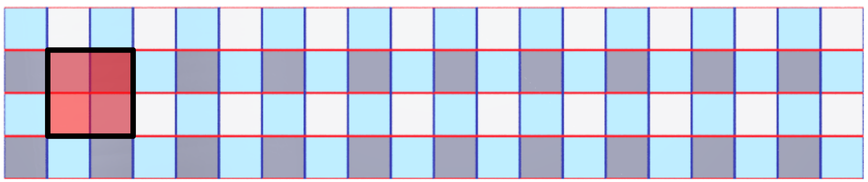


Not Bijective!

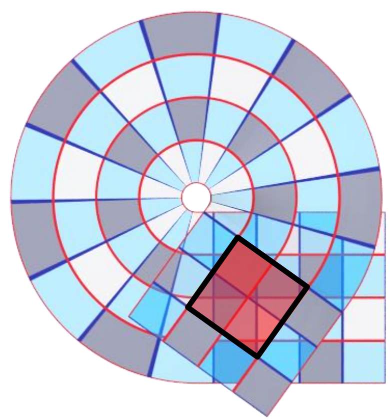


Globally Bijective VS. Locally Bijective

U

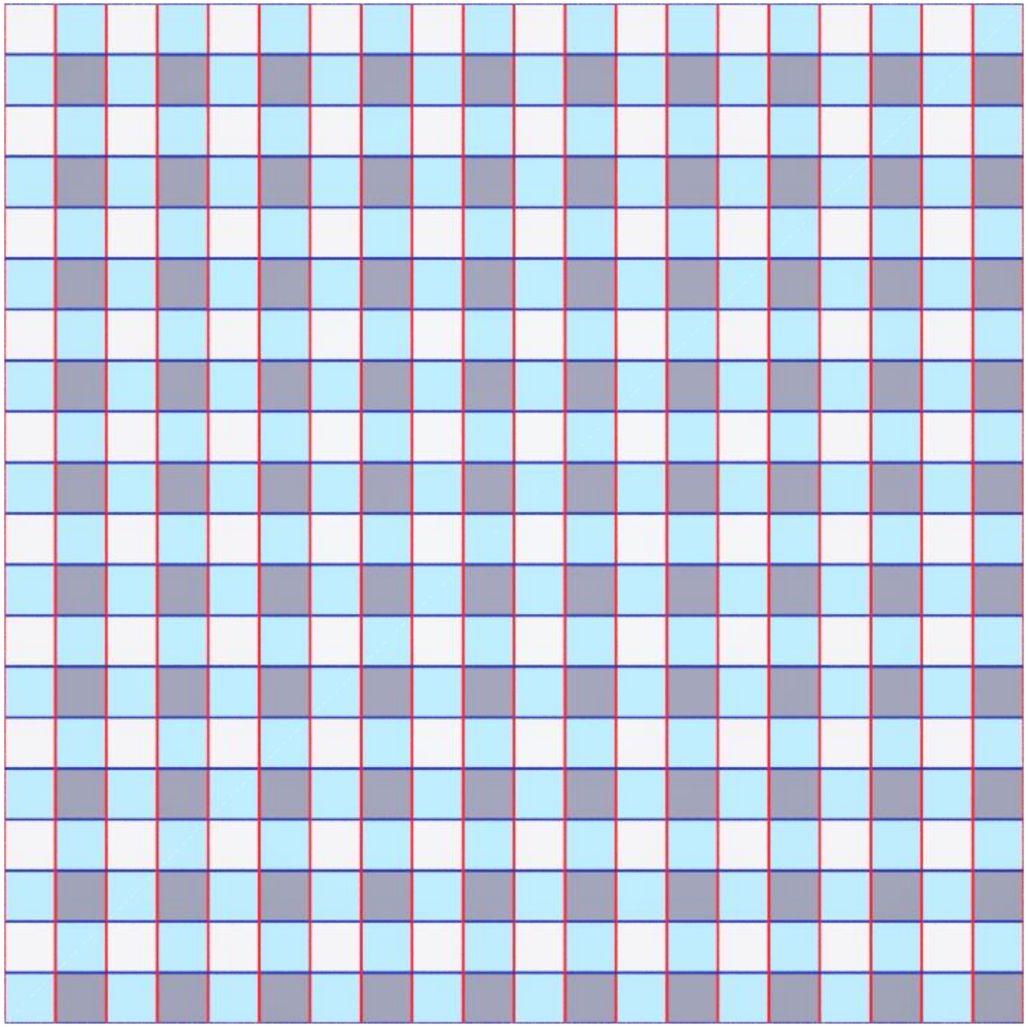


Only Locally Bijective!



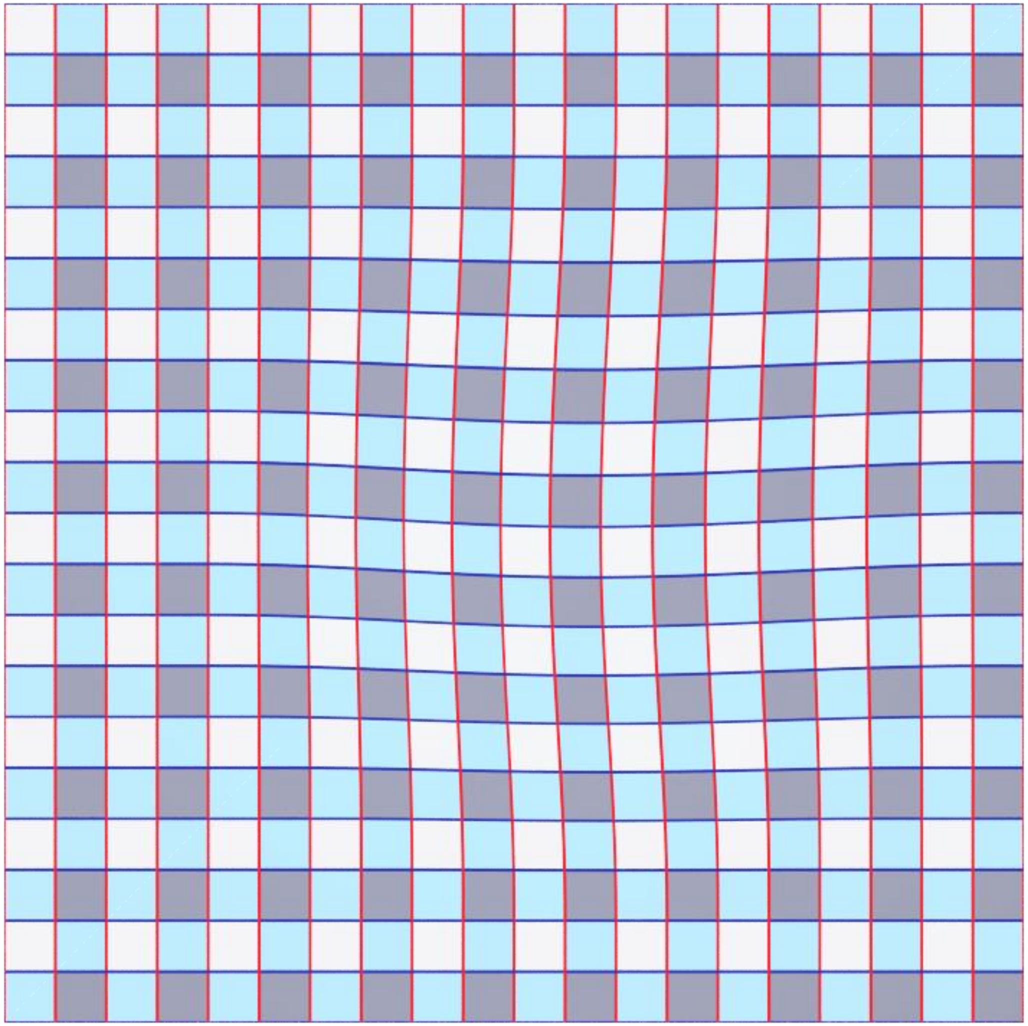
f(U)

Locally Bijection – Non-example

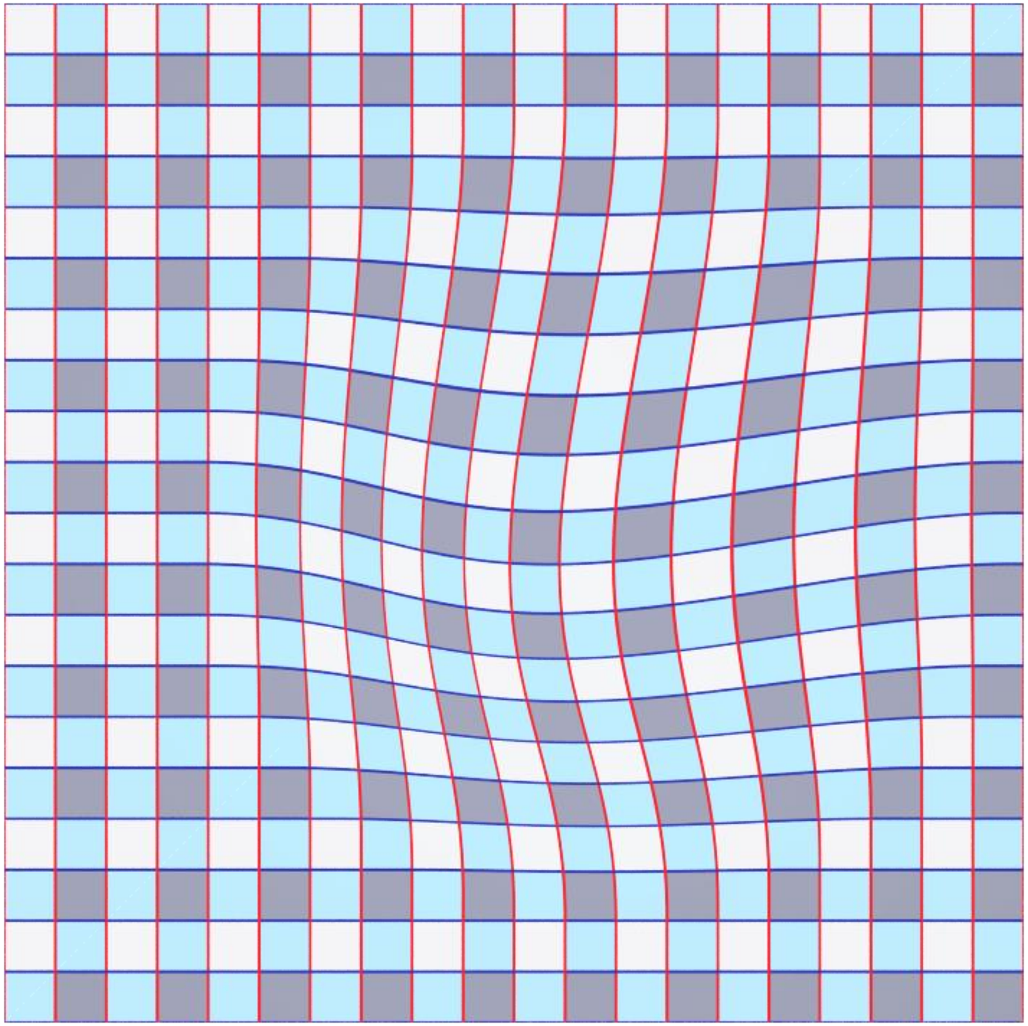


ETH zürich

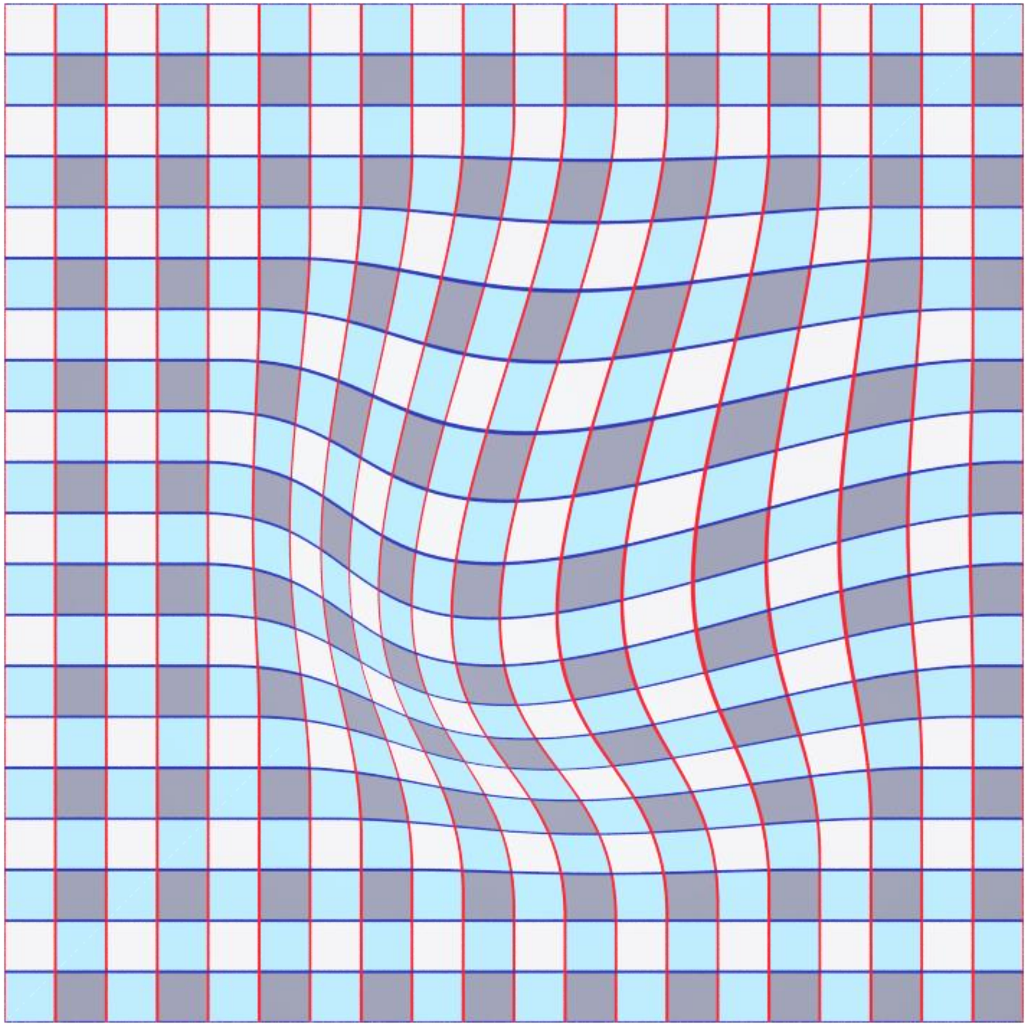
Locally Bijection – Non-example



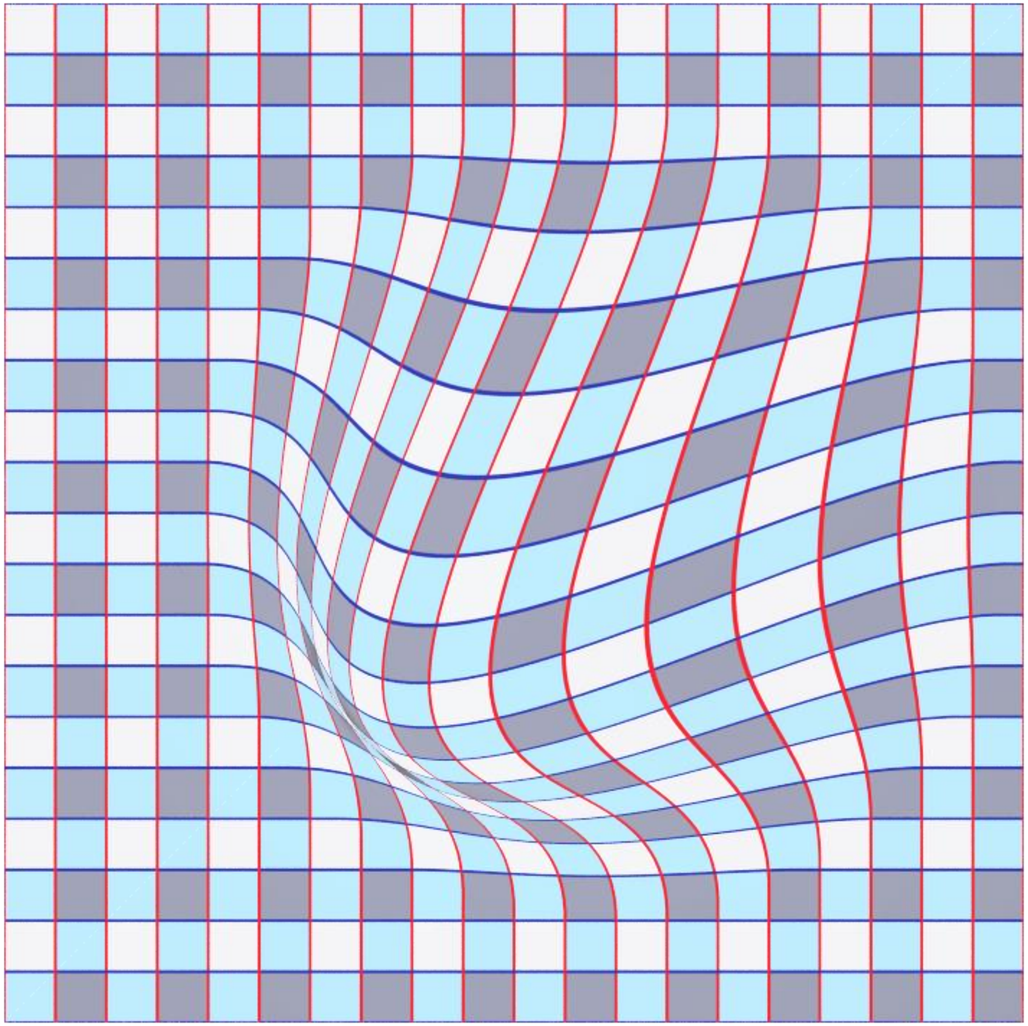
Locally Bijection – Non-example



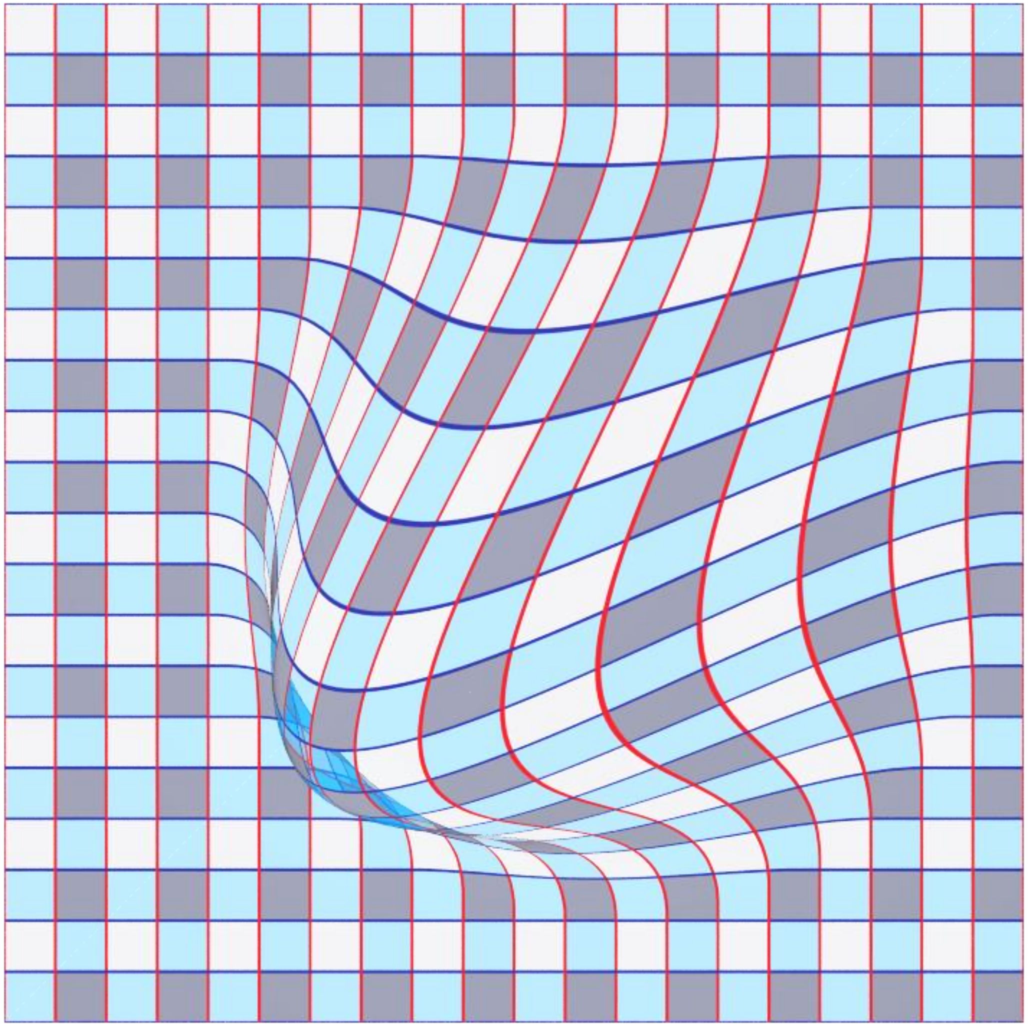
Locally Bijection – Non-example



Locally Bijection – Non-example

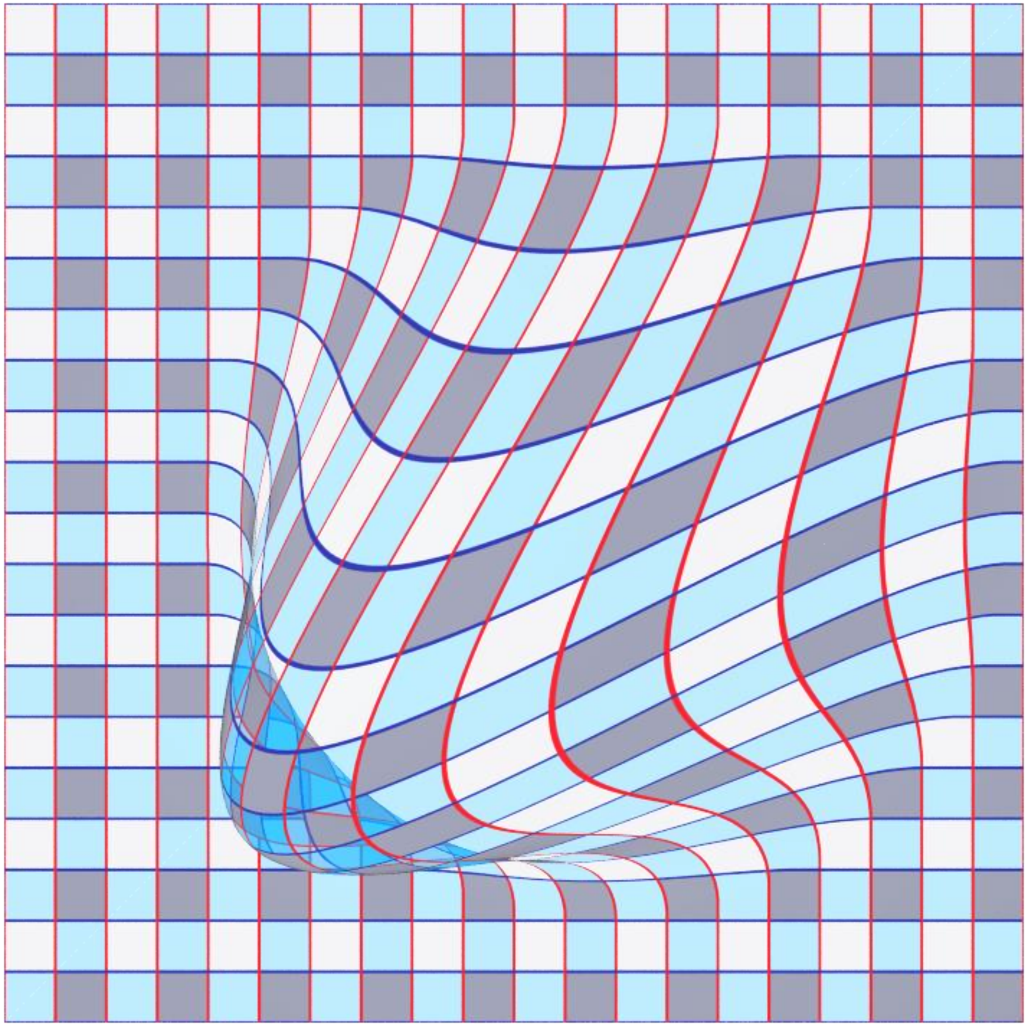


Locally Bijection – Non-example

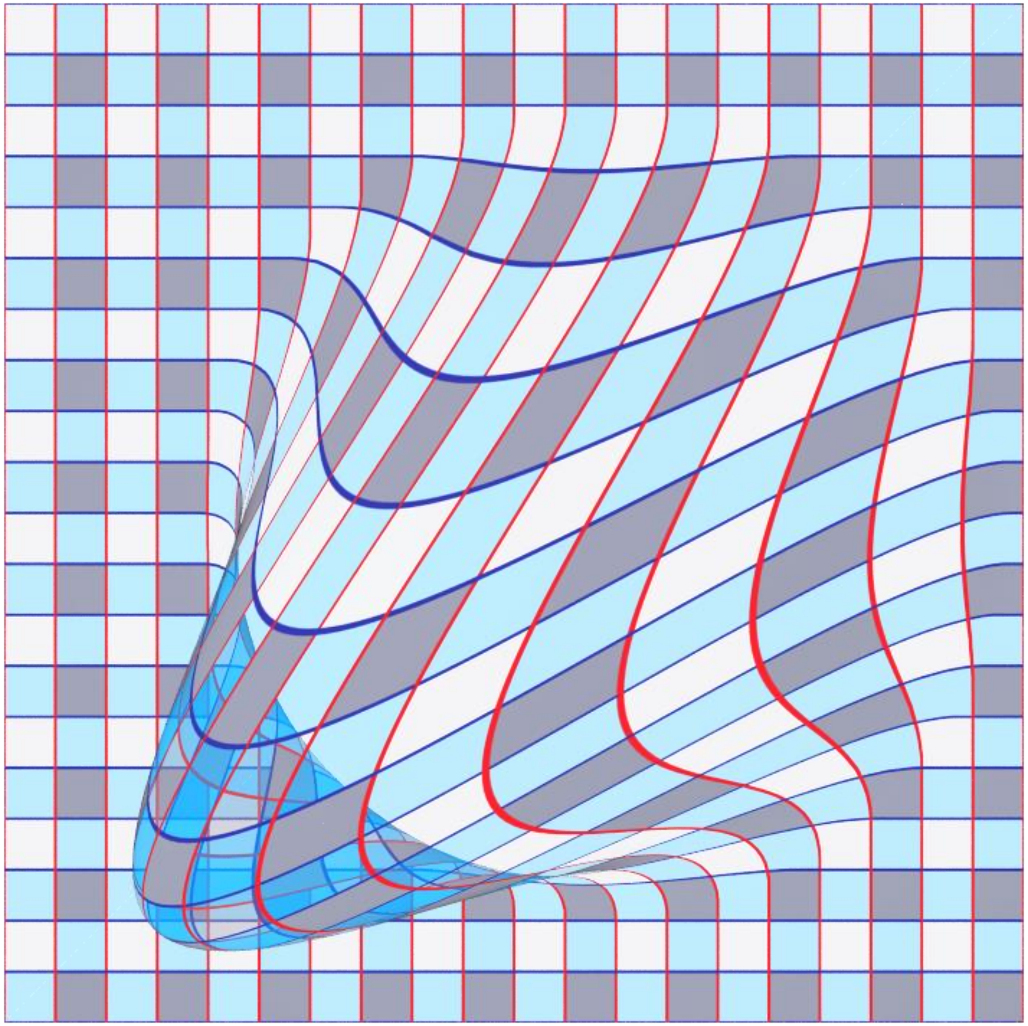


ETH zürich

Locally Bijection – Non-example

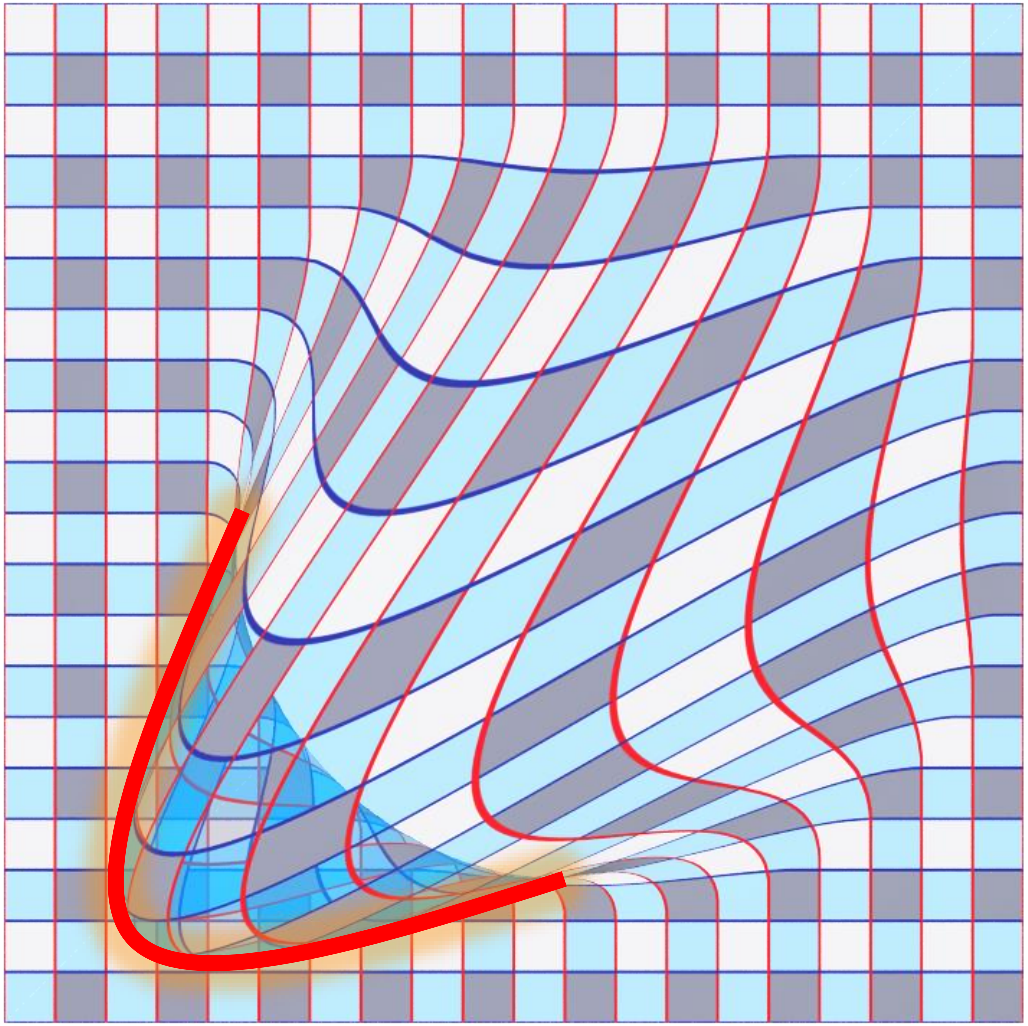


Locally Bijection – Non-example



ETH zürich

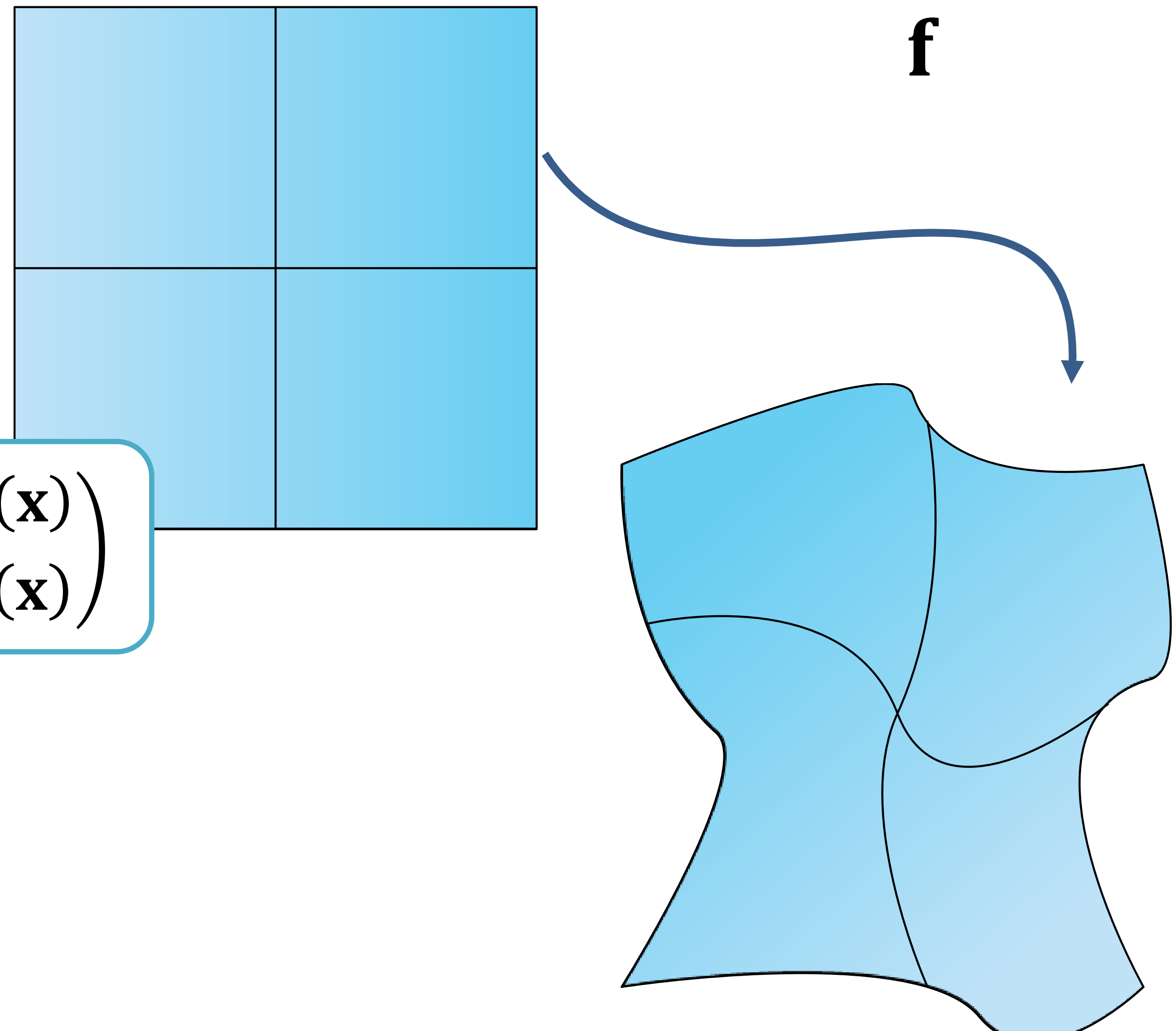
Locally Bijection – Non-example



ETH zürich

Local Bijection Sufficient condition

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} u(\mathbf{x}) \\ v(\mathbf{x}) \end{pmatrix}$$

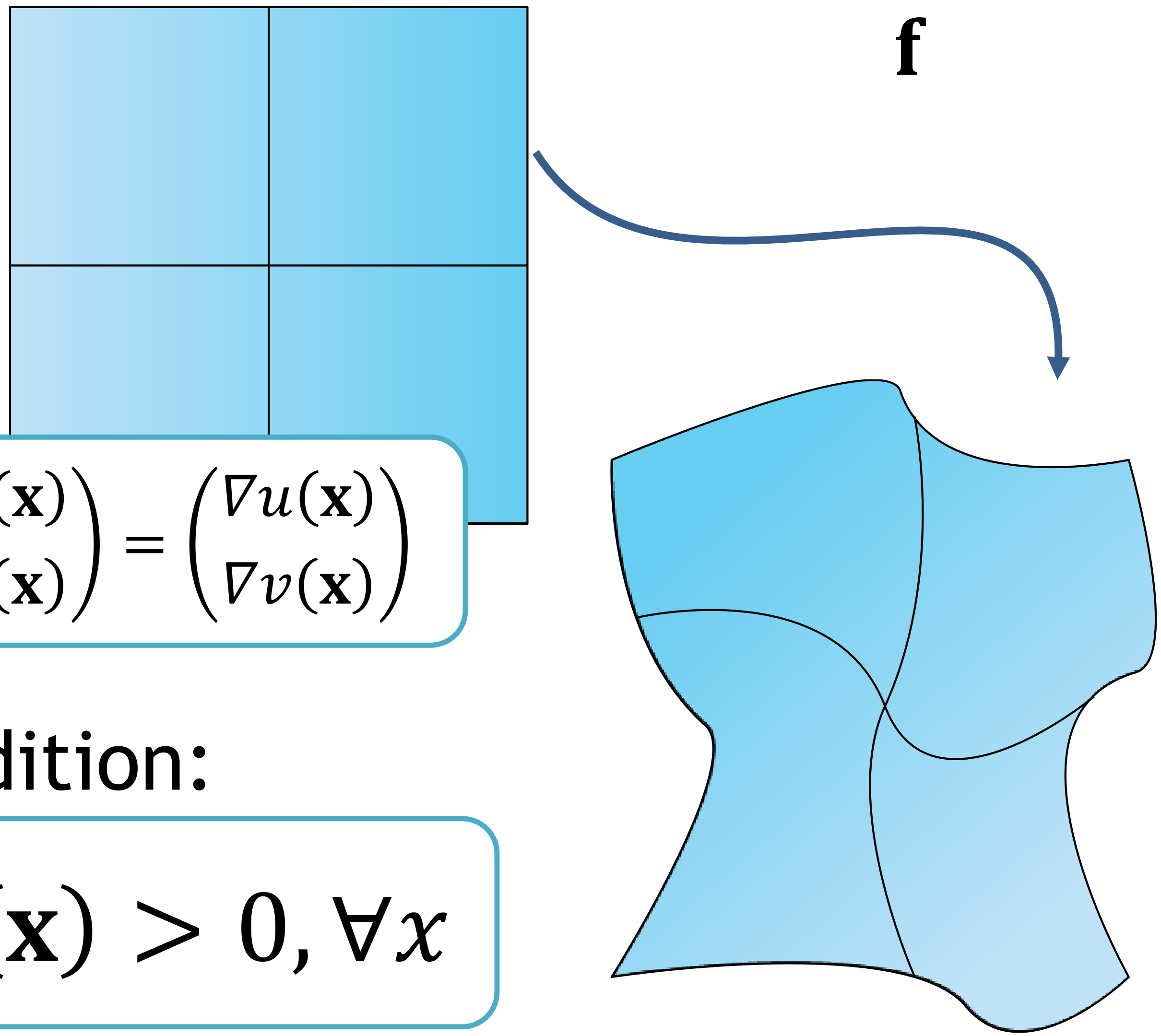


The Jacobian:

$$\mathcal{J}\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \partial_x u(\mathbf{x}) & \partial_y u(\mathbf{x}) \\ \partial_x v(\mathbf{x}) & \partial_y v(\mathbf{x}) \end{pmatrix}$$

Local Bijection Sufficient condition

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} u(\mathbf{x}) \\ v(\mathbf{x}) \end{pmatrix}$$



The Jacobian:

$$\mathcal{J}\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \partial_x u(\mathbf{x}) & \partial_y u(\mathbf{x}) \\ \partial_x v(\mathbf{x}) & \partial_y v(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \nabla u(\mathbf{x}) \\ \nabla v(\mathbf{x}) \end{pmatrix}$$

The Condition:

$$\det \mathcal{J}\mathbf{f}(\mathbf{x}) > 0, \forall x$$

Globally VS. Locally Bijective

Globally Bijective $\Rightarrow$ Locally Bijective

$\mathbf{f}$ is bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective

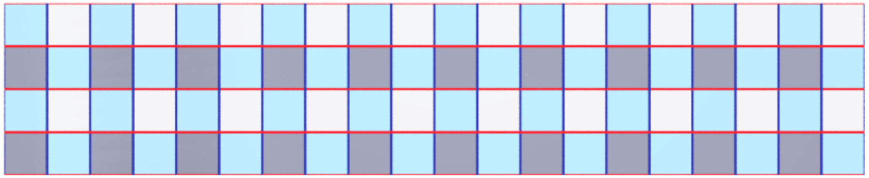
Globally VS. Locally Bijective

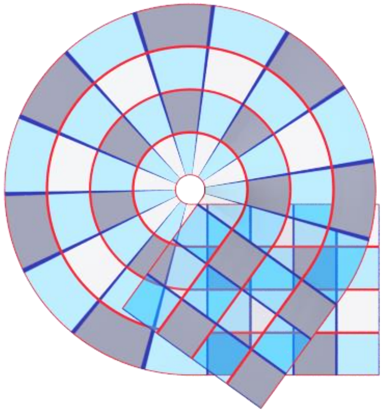
Globally Bijective

Locally Bijective

$\mathbf{f}$ is bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective





ETH zürich

Globally VS. Locally Bijective

Globally Bijective

Locally Bijective

$\mathbf{f}$ is bijective

$\mathbf{f}: \mathbf{U} \rightarrow \mathbf{f}(\mathbf{U})$ is bijective

“Global inversion theorems”

What are good maps?

Local

Bijectivity

Low distortion

Not Bijective

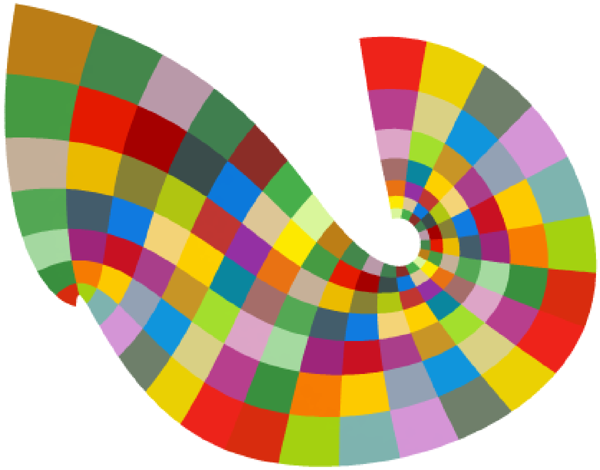
Bijective

Lower distortion

Distortion - Types

Conformal distortion

Isometric distortion

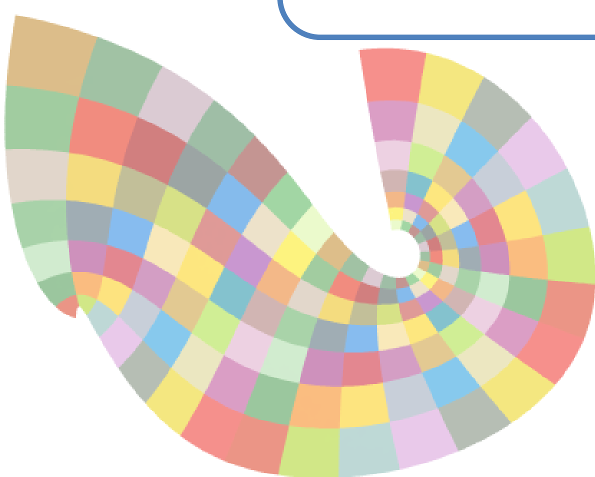


Distortion - Types

Conformal distortion

Isometric distortion

The distortion is a function of the Jacobian at a point



Distortion - LSCM [Lévy et al. 2002]

LSCM – **L**east **S**quares **C**onformal **M**ap

We want the Jacobian to be a similarity matrix

$$\begin{pmatrix} \partial_x u & \partial_y u \\ \partial_x v & \partial_y v \end{pmatrix} \qquad \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix}$$

$$\begin{aligned} \partial_x u &= \partial_y v \\ \partial_y u &= -\partial_x v \end{aligned}$$

Cauchy-Riemann Equations

Distortion - LSCM

LSCM – **L**east **S**quares **C**onformal **M**ap

We want the Jacobian to be a similarity matrix

$$\begin{pmatrix} \partial_x u & \partial_y u \\ \partial_x v & \partial_y v \end{pmatrix} \qquad \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix}$$

$$\mathcal{D}_{\mathrm{LSCM}} = (\partial_x u - \partial_y v)^2 + (\partial_y u + \partial_x v)^2$$

Quick Notation Change

$$\mathcal{J}\mathbf{f} \Longrightarrow \mathbf{A}$$

$$\begin{pmatrix} \partial_x u & \partial_y u \\ \partial_x v & \partial_y v \end{pmatrix} \Longrightarrow \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$\mathcal{D}_{\mathrm{LSCM}} = (\partial_x u - \partial_y v)^2 + (\partial_y u + \partial_x v)^2$$

$$(a - d)^2 + (b + c)^2$$

Distortion – ASAP

[Liu et al. 2008]

ASAP– **As** Similar **As** Possible

$$\mathcal{D}_{\text{ASAP}} = \|\underbrace{\mathbf{A}}_{\text{Jacobian}} - \underbrace{\mathcal{S}_{\mathbf{A}}}_{\text{Closest Similarity}}\|_F^2$$

How to compute closest similarity?

ETH zürich

Distortion – ASAP

ASAP– **A**s **S**imilar **A**s **P**ossible

How to compute closest similarity?

Distortion – ASAP

ASAP– **A**s **S**imilar **A**s **P**ossible

How to compute closest similarity?

In 2D:

$$\min_{\mathcal{S}} \|\mathbf{A} - \mathcal{S}\|_F^2$$
$$\text{s.t.} \quad \mathcal{S} \text{ is similarity}$$

$$\min_{\alpha,\beta} \left\| \begin{pmatrix} a & b \\ c & d \end{pmatrix} - \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix} \right\|_F^2$$

Distortion – ASAP

ASAP– **A**s **S**imilar **A**s **P**ossible

How to compute closest similarity?

In 2D:

$$\min_{\mathcal{S}} \|\mathbf{A} - \mathcal{S}\|_F^2$$
$$\text{s.t.} \quad \mathcal{S} \text{ is similarity}$$

$$\mathcal{S} = \frac{1}{2}\begin{pmatrix} a+d & c-b \\ b-c & a+d \end{pmatrix}$$

Distortion – ASAP

ASAP– **A**s **S**imilar **A**s **P**ossible

How to compute closest similarity?

In 2D:

$$\mathbf{A} = \frac{1}{2}\begin{pmatrix} a+d & c-b \\ b-c & a+d \end{pmatrix} + \frac{1}{2}\begin{pmatrix} a-d & c+b \\ b+c & d-a \end{pmatrix}$$

Distortion – ASAP

ASAP– **A**s **S**imilar **A**s **P**ossible

How to compute closest similarity?

In 2D:

$$\mathbf{A} = \underbrace{\overbrace{\frac{1}{2}\begin{pmatrix} a+d & c-b \\ b-c & a+d \end{pmatrix}}^{\mathcal{S}_A}}_{\text{Similarity}} + \underbrace{\overbrace{\frac{1}{2}\begin{pmatrix} a-d & c+b \\ b+c & d-a \end{pmatrix}}^{\mathcal{S}_A^{\perp}}}_{\text{Anti-Similarity}}$$

Distortion – ASAP

ASAP– **A**s **S**imilar **A**s **P**ossible

$$\| \quad \mathcal{S}_A^{\perp} \quad \|_F^2$$

Measure of antisimilarity

Jacobian

Closest Similarity

$$\left\| \begin{pmatrix} a-d & c+b \\ b+c & d-a \end{pmatrix} \right\|_F^2$$

$$(a-d)^2 + (b+c)^2$$

Distortion – ASAP

ASAP- As Similar As Possible

$$\| \quad \mathcal{S}_A^{\perp} \quad \|_F^2$$

y

LSCM = ASAP

$$\left\| \begin{pmatrix} a-d & c+b \\ b+c & d-a \end{pmatrix} \right\|_F^2$$

$$(a-d)^2+(b+c)^2$$

Distortion - ARAP

ARAP– **As Rigid As Possible**

$$\mathcal{D}_{\text{ARAP}} = \|\mathbf{A} - \mathcal{R}_{\mathbf{A}}\|_F^2$$

Jacobian

Closest Rotation

How to compute closest rotation?

Singular Value Decomposition

Every Matrix M has a factorization of the form

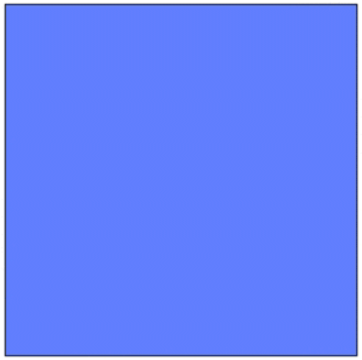
$$M \quad = \quad U \qquad\qquad S \qquad\qquad V^T$$

$$\begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} \quad \sigma_1 > \sigma_2$$

Singular Value Decomposition

Every Matrix M has a factorization of the form

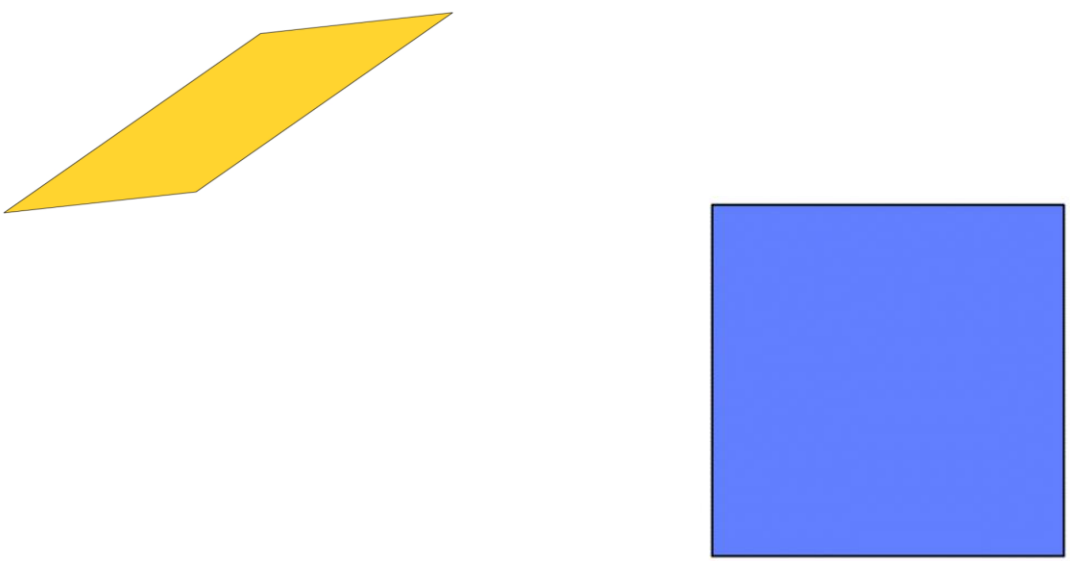
$$M \quad = \quad U \qquad\qquad S \qquad\qquad V^T$$



Singular Value Decomposition

Every Matrix M has a factorization of the form

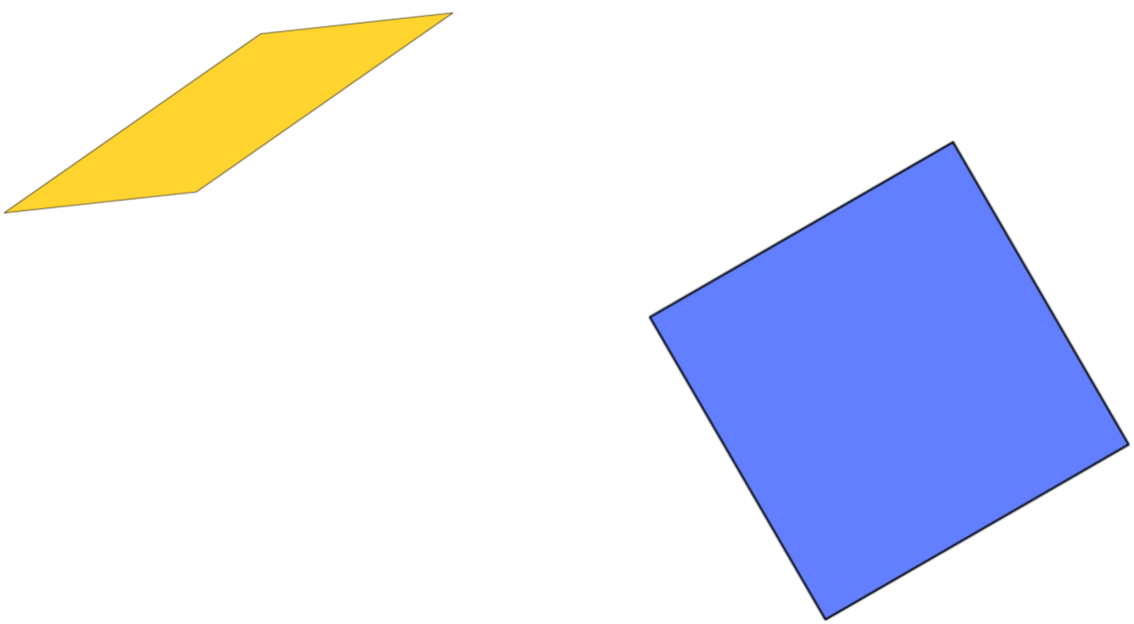
$$M \quad = \quad U \qquad\qquad S \qquad\qquad V^T$$



Singular Value Decomposition

Every Matrix M has a factorization of the form

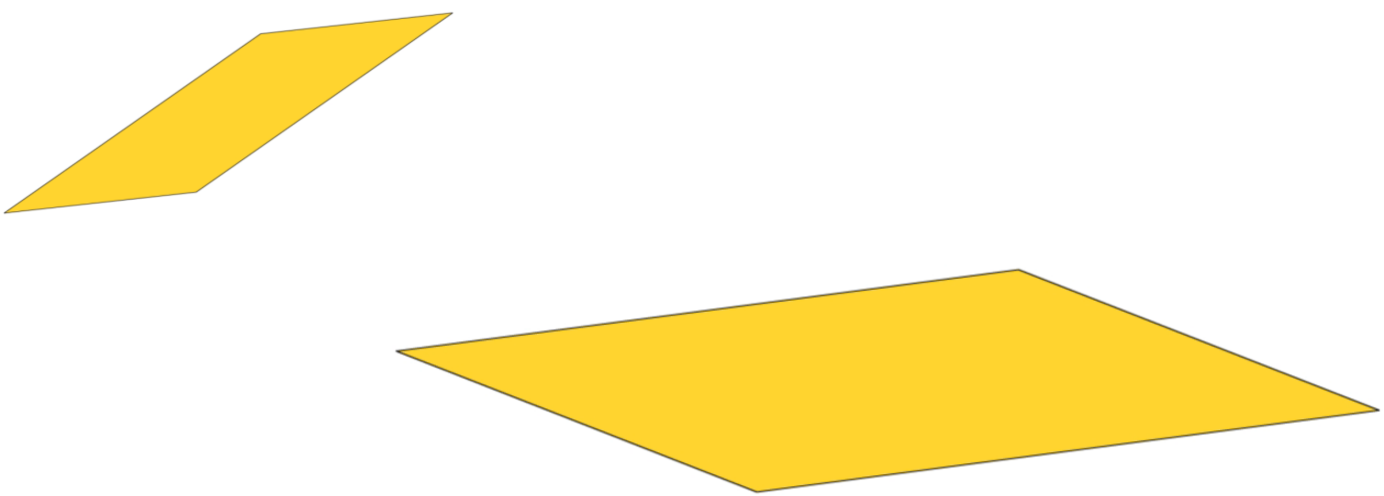
$$M \quad = \quad U \qquad\qquad S \qquad\qquad V^T$$



Singular Value Decomposition

Every Matrix M has a factorization of the form

$$M \quad = \quad U \qquad\qquad S \qquad\qquad V^T$$



Singular Value Decomposition

Every Matrix M has a factorization of the form

$$M \quad = \quad U \qquad\qquad S \qquad\qquad V^T$$

U and V are not rotations!

Singular Value Decomposition

Every Matrix M has a factorization of the form

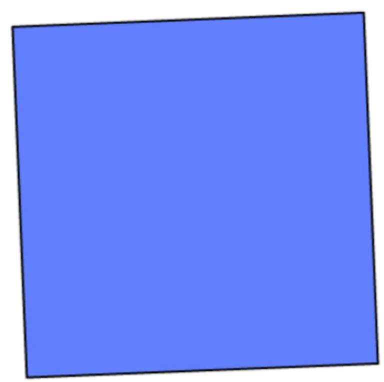
$$M \quad = \quad U \qquad\qquad S \qquad\qquad V^T$$

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Singular Value Decomposition

Every Matrix M has a factorization of the form

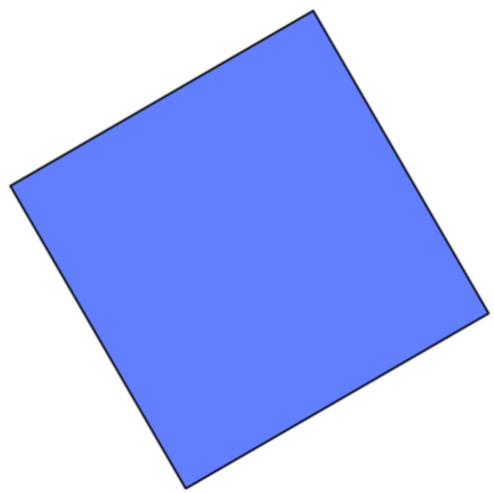
$$M \quad = \quad U \qquad\qquad S \qquad\qquad RV^T$$



Singular Value Decomposition

Every Matrix M has a factorization of the form

$$M \quad = \quad U \qquad\qquad S \qquad\qquad RV^T$$

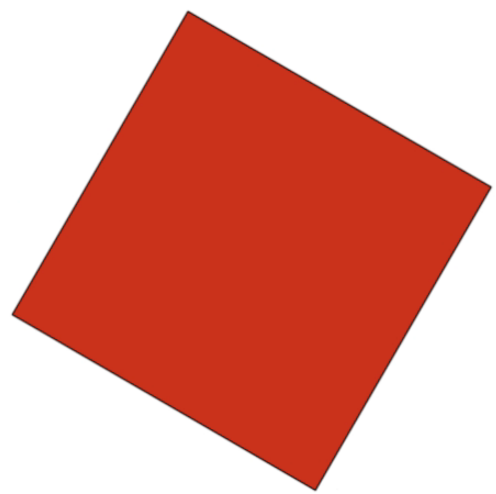


ETH zürich

Singular Value Decomposition

Every Matrix M has a factorization of the form

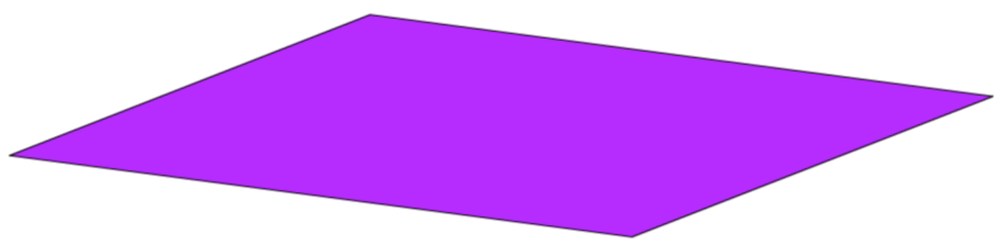
$$M \quad = \quad U \qquad\qquad S \qquad\quad RV^T$$



Singular Value Decomposition

Every Matrix M has a factorization of the form

$$M \quad = \quad U \qquad\qquad S \qquad\qquad RV^T$$



Signed

Singular Value Decomposition

Every Matrix M has a factorization of the form

$$M = U \quad S \quad RV^T$$

Signed Singular Value Decomposition

Every Matrix M has a factorization of the form

$$M = U \quad SR \quad V^T$$

$$\begin{pmatrix} \sigma_1 & 0 \\ 0 & -\sigma_2 \end{pmatrix} \sigma_1 > \sigma_2$$

Now U and V are rotations!

Signed

Singular Value Decomposition

Every Matrix M has a factorization of the form

$$M = U \quad SR \quad V^T$$

$$\begin{pmatrix} \sigma_1 & 0 \\ 0 & -\sigma_2 \end{pmatrix} \sigma_1 > \sigma_2$$

Now U and V are rotations!

What if U and V both had reflections?

Signed Singular Value Decomposition

Every Matrix M has a factorization of the form

$$M \quad = \quad U \qquad\qquad SR \qquad\qquad V^T$$

$$\begin{pmatrix} \sigma_1 & 0 \\ 0 & -\sigma_2 \end{pmatrix} \sigma_1 > \sigma_2$$

Now U and V are rotations!

What if U and V both had reflections?

$$\operatorname{sign}\det M = \operatorname{sign}(\sigma_2)$$

Distortion - ARAP

ARAP- **As Rigid As Possible**

$$\mathcal{D}_{\text{ARAP}} = \|\mathbf{A} - \mathcal{R}_{\mathbf{A}}\|_F^2$$

Jacobian

Closest Rotation

How to compute closest Rotation?

Distortion - ARAP

ARAP– **As Rigid As Possible**

$$\mathcal{D}_{\text{ARAP}} = \|\mathbf{A} - \mathcal{R}_{\mathbf{A}}\|_F^2$$

Jacobian

Closest Rotation

$$\mathbf{A} = USV^T \qquad \mathcal{R}_{\mathbf{A}} = UV^T$$

Proof: Using Lagrange multipliers

Distortion – ASAP

ASAP– **As** Similar **As** Possible

$$\mathcal{D}_{\text{ASAP}} = \|\mathbf{A} - \mathcal{S}_{\mathbf{A}}\|_F^2$$

Jacobian

Closest Similarity

$$\mathbf{A} = USV^T$$

$$\mathcal{S}_{\mathbf{A}} = \bar{\sigma}UV^T$$

$$\bar{\sigma} = \frac{\sigma_1 + \sigma_2}{2}$$

Distortion - ARAP

ARAP– **As Rigid As Possible**

$$\mathcal{D}_{\text{ARAP}} = \|\underbrace{\mathbf{A}}_{\text{Jacobian}} - \underbrace{\mathcal{R}_{\mathbf{A}}}_{\text{Closest Rotation}}\|_F^2$$

$$\mathbf{A} = USV^T \qquad \mathcal{R}_{\mathbf{A}} = UV^T$$

Proof: Using Lagrange multipliers

ETH zürich

Distortion - ARAP

ARAP– **A**s **R**igid **A**s **P**ossible

$$\mathcal{D}_{\text{ARAP}} = \|\mathbf{A} - \mathcal{R}_{\mathbf{A}}\|_F^2 = \|\mathbf{A} - UV^T\|_F^2$$

$$= \|USV^T - UV^T\|_F^2$$

$$= \|U(S - I)V^T\|_F^2$$

$$= \|(S - I)\|_F^2$$

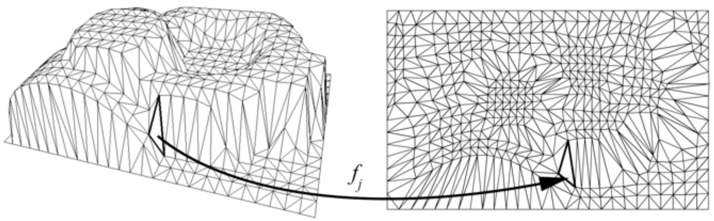
$$= (\sigma_1 - 1)^2 + (\sigma_2 - 1)^2$$

Distortion - ASAP

ASAP– **A**s **S**imilar **A**s **P**ossible

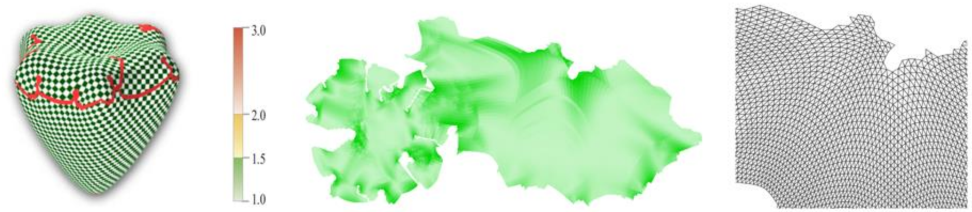
$$
\begin{aligned}
\mathcal{D}_{\mathrm{ASAP}} &= \|\mathbf{A} - \mathcal{S}_{\mathbf{A}}\|_F^2 \\
&= \|USV^T - \bar{\sigma}UV^T\|_F^2 \\
&= \|U(S - \bar{\sigma}I)V^T\|_F^2 \\
&= (\sigma_1 - \bar{\sigma})^2 + (\sigma_2 - \bar{\sigma})^2 \\
&= (\sigma_1 - \sigma_2)^2
\end{aligned}
$$

$$\frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1}$$



MIPS [Hormann & Greiner 2000]

$$\max\left\{\sigma_1, \frac{1}{\sigma_2}\right\}$$

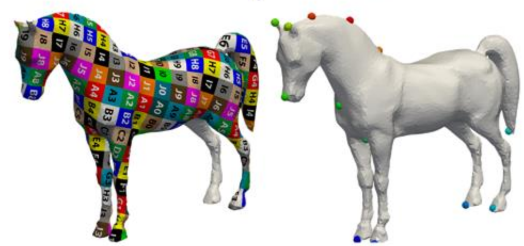


[Sorkine et al. 2000]

$$\sigma_1^2 + \sigma_2^2 + \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}$$

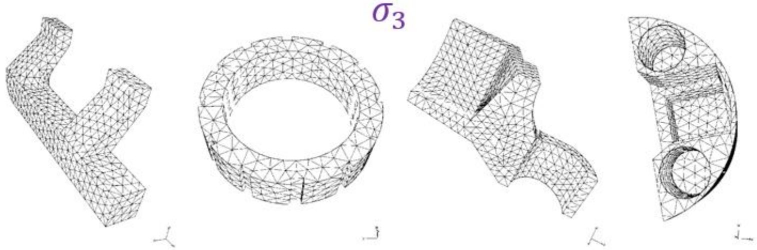
[Schreiner et al. 2014]

$$\sqrt{\sigma_1^2 + \frac{1}{\sigma_2^2}}$$



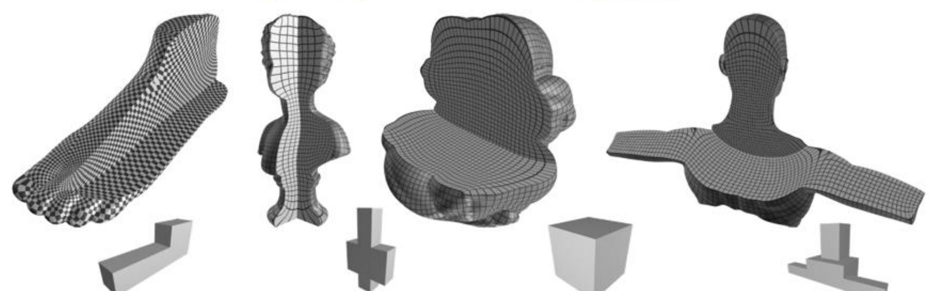
[Aigerman et al. 2014]

$$\frac{\sigma_1}{\sigma_3}$$



[Freitag & Knupp 2002]

$$\frac{\sigma_1}{\sigma_3} + \frac{\sigma_3}{\sigma_1}; \quad \sigma_1\sigma_2\sigma_3 + \frac{1}{\sigma_1\sigma_2\sigma_3}$$



[Paillé & Poulin 2012]

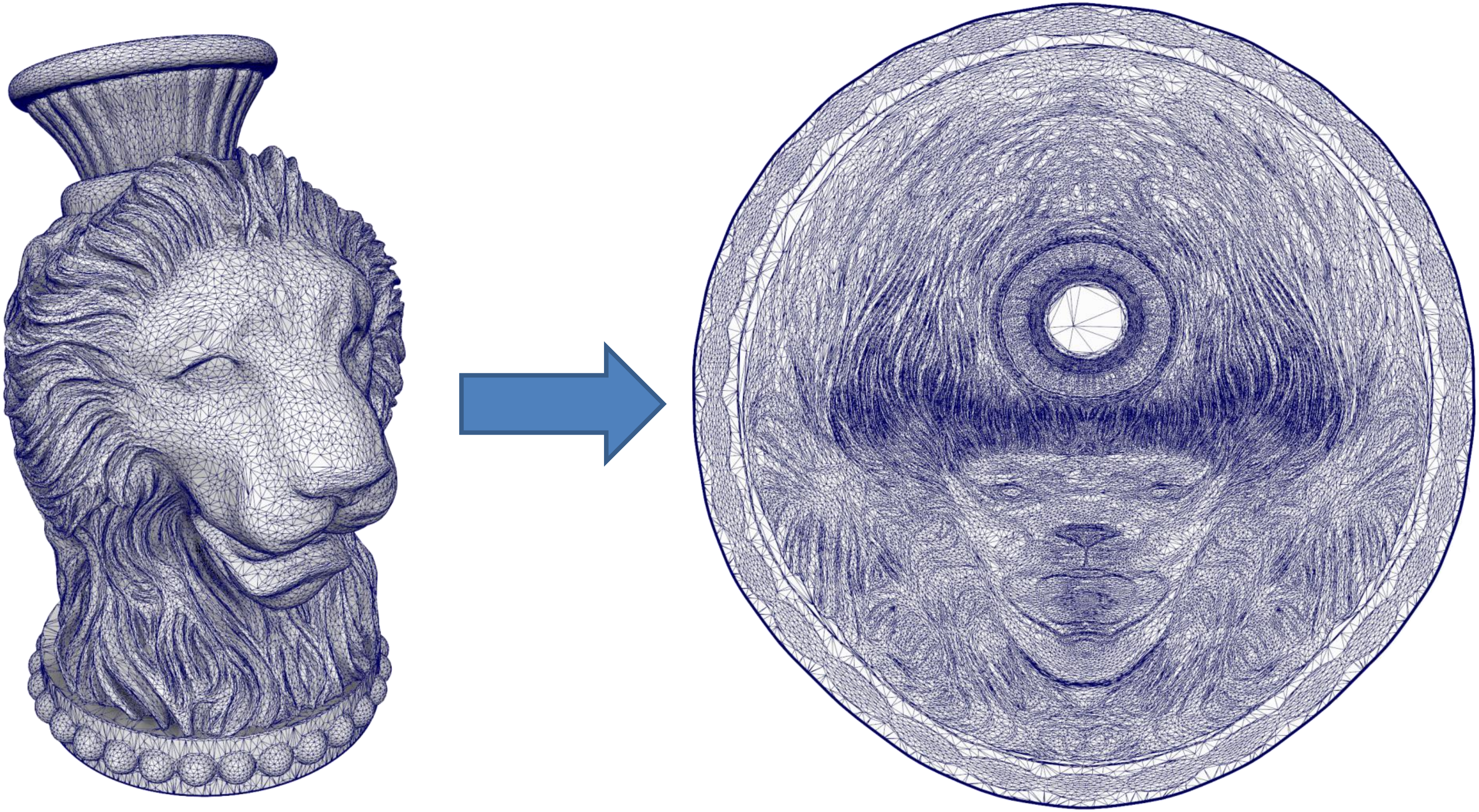
252-0538-00L, Spring 2018

Shape Modeling and Geometry Processing

Mesh Parameterization

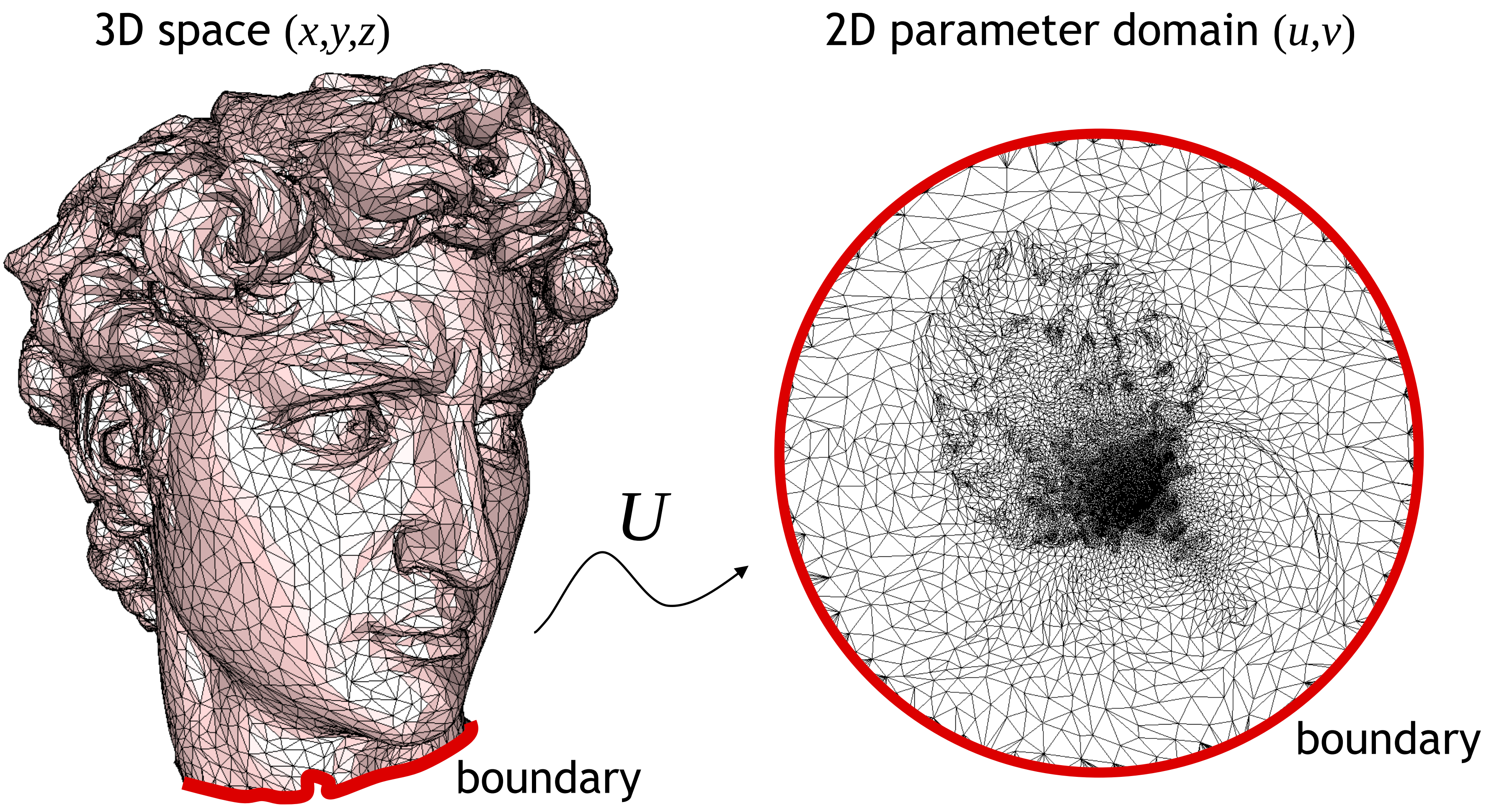
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

What Is a Parameterization?



What Is a Parameterization?

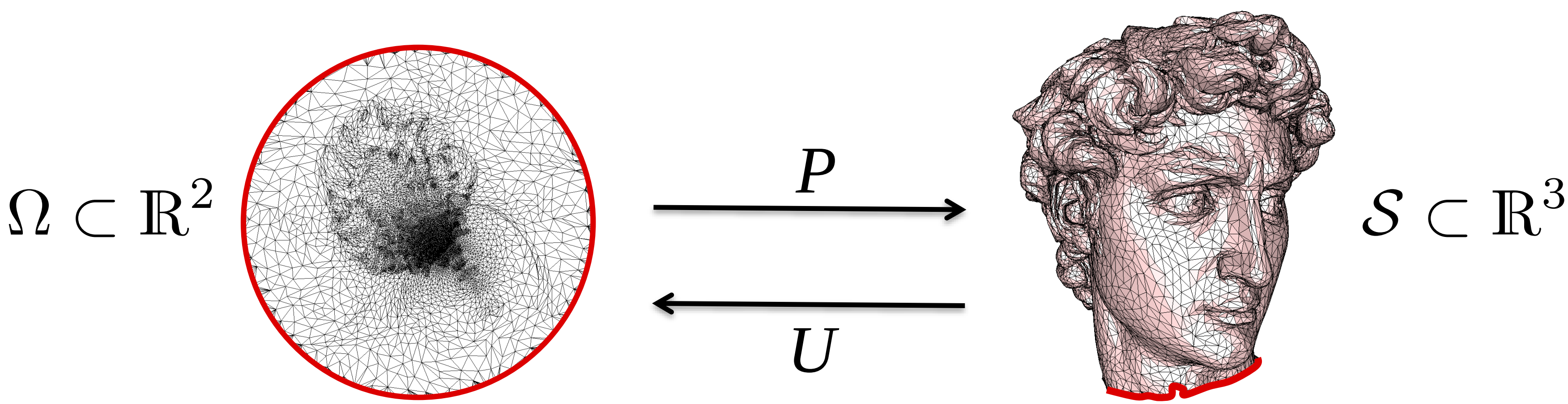
What Is a Parameterization?



ETH zürich

Parameterization – Definition

- Mapping P between a 2D domain Ω and the mesh S embedded in 3D (the inverse = flattening)
- Each mesh vertex has a corresponding 2D position:
$$U(\mathbf{v}_i) = (u_i, v_i)$$
- Inside each triangle, the mapping is affine (barycentric coordinates)

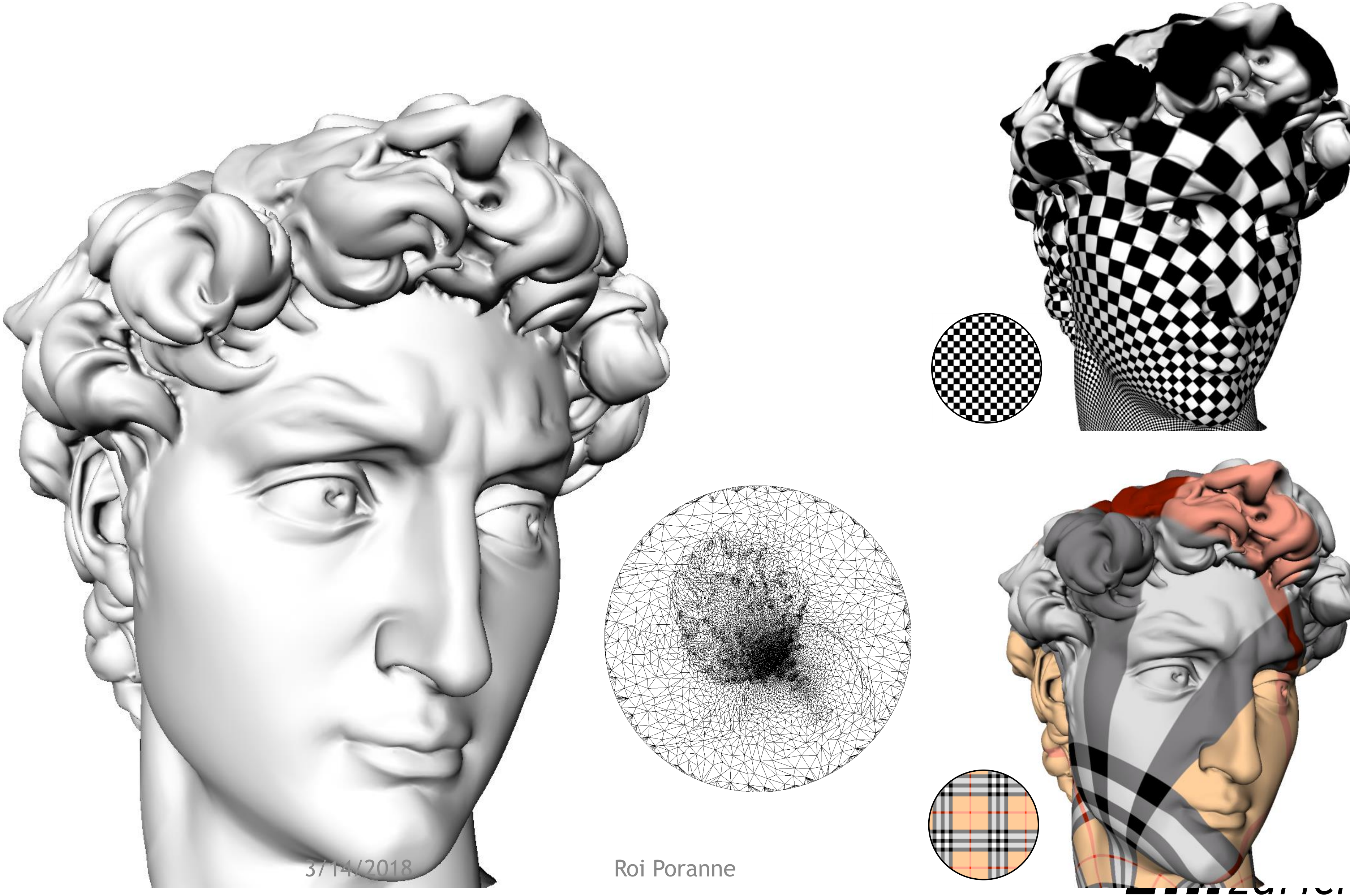


Why Parameterization?

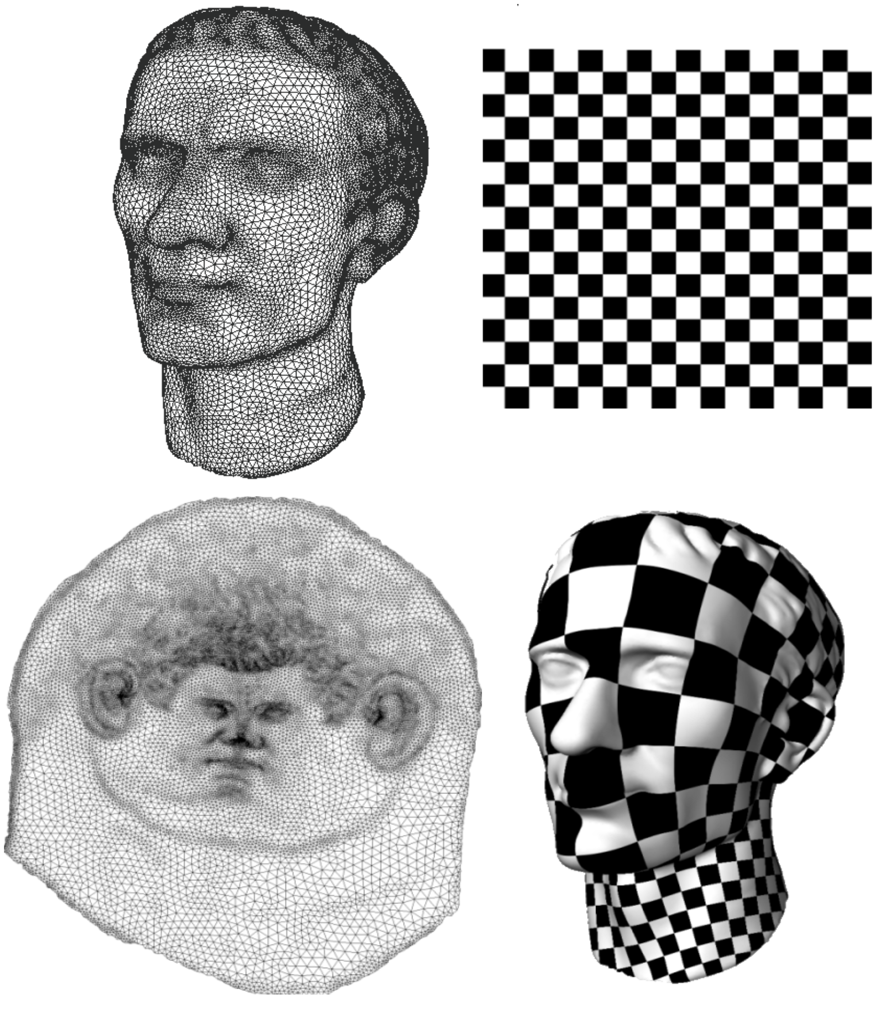
- Allows us to do many things in 2D and then map those actions onto the 3D surface
- It is often easier to operate in the 2D domain

- Mesh parameterization allows to use some notions from continuous surface theory

Main Application: Texture Mapping



Main Application: Texture Mapping



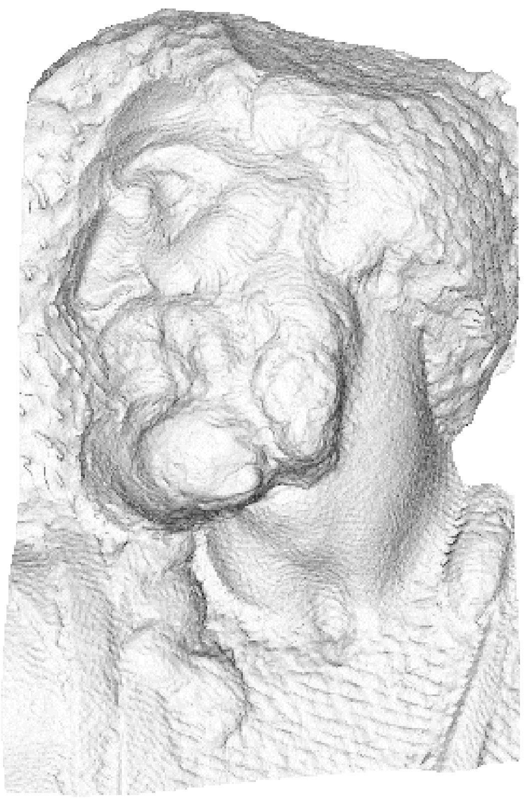
Texture Mapping

ETH zürich

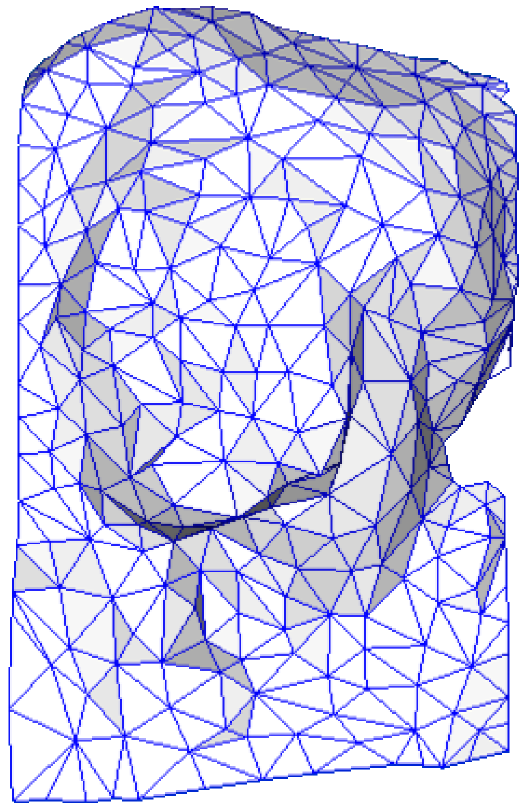
Texture Mapping

Image from Vallet and Levy, techreport INRIA

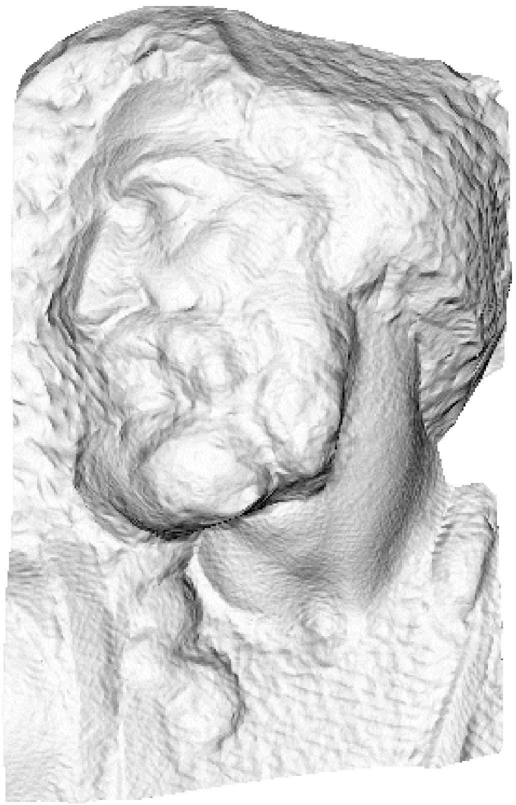
Normal/Bump Mapping



original mesh
4M triangles

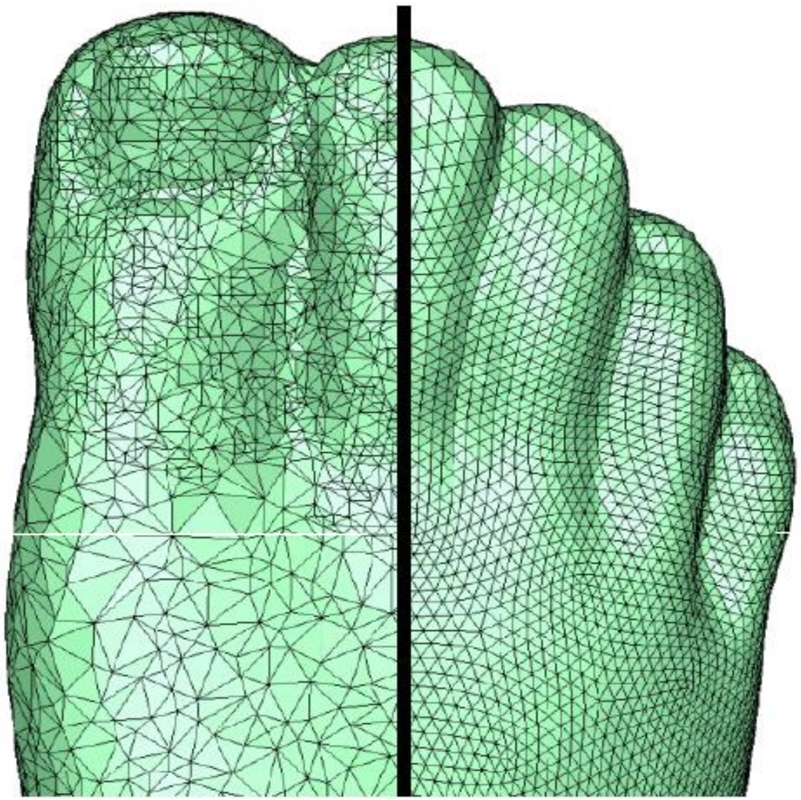


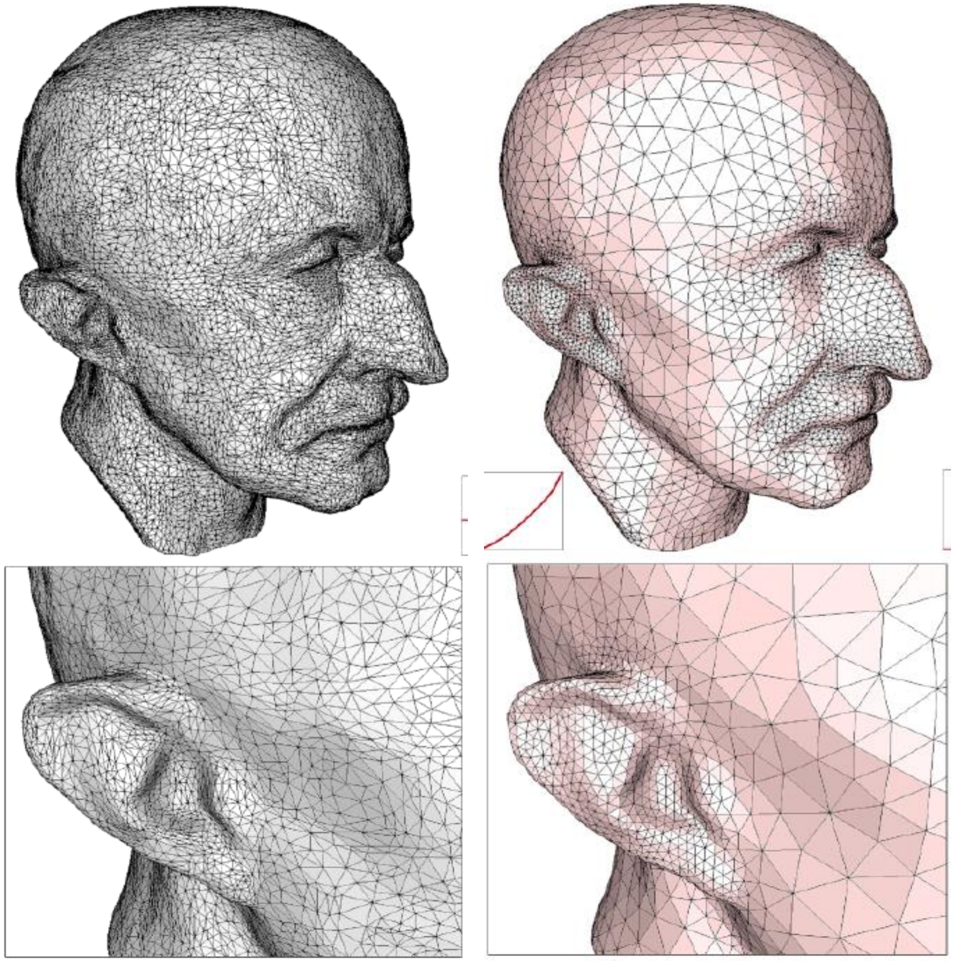
simplified mesh
500 triangles



simplified mesh
and normal mapping
500 triangles

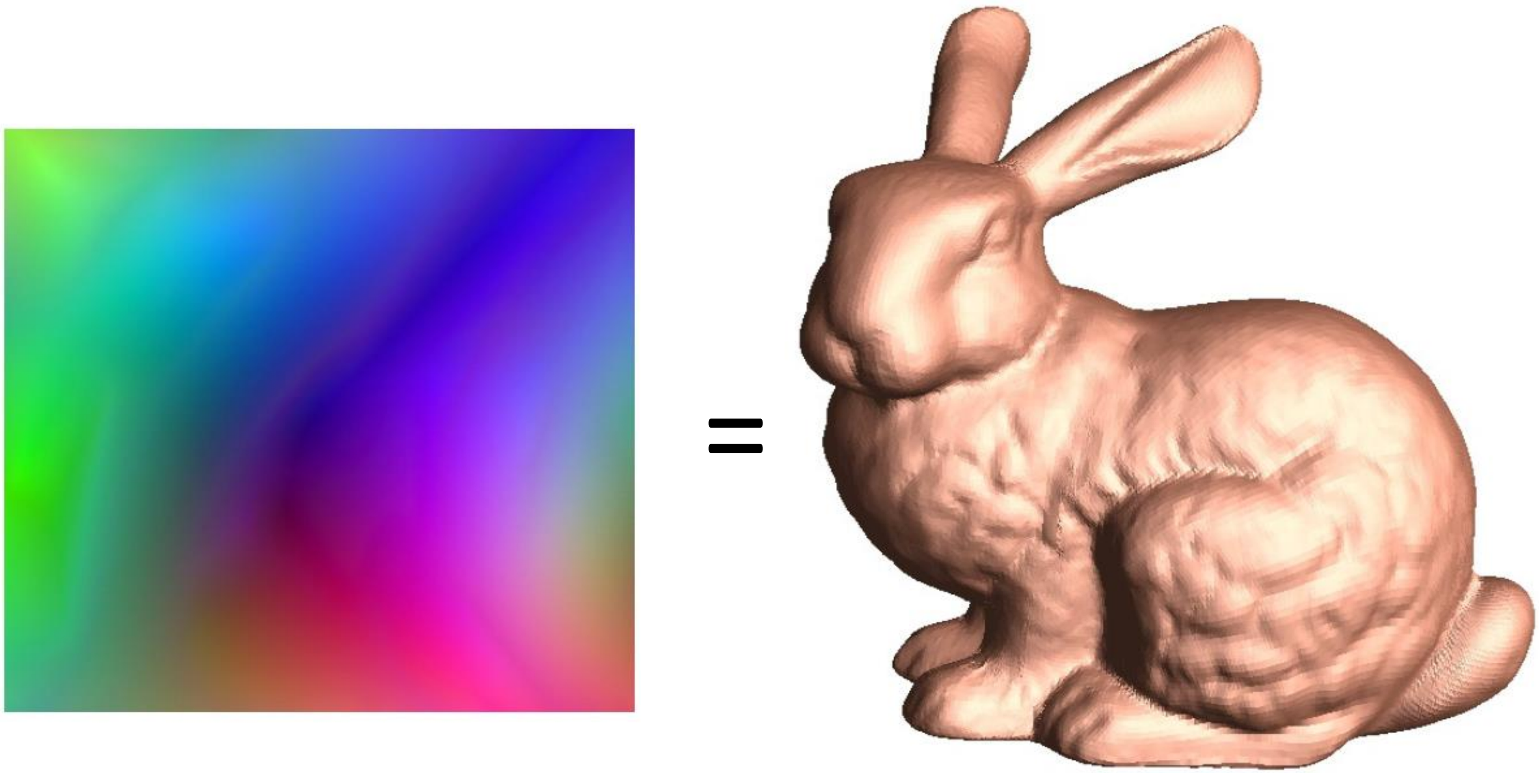
Remeshing





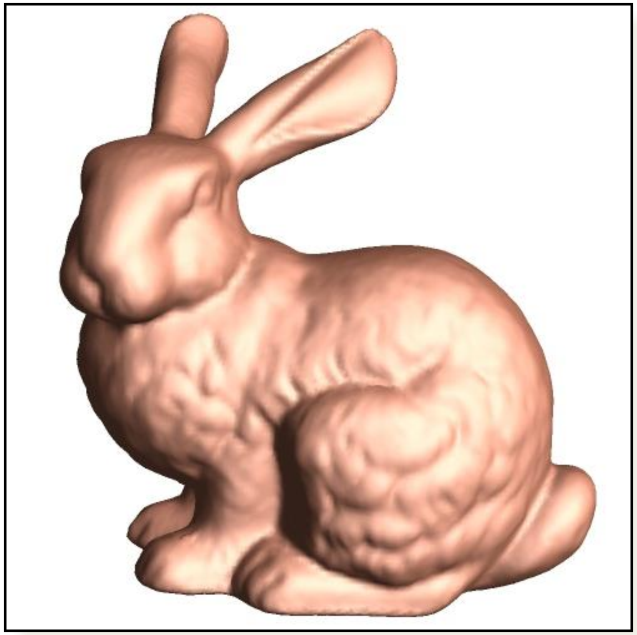
“Interactive Geometry Remeshing”, Alliez et al., SIGGRAPH 2002

Compression

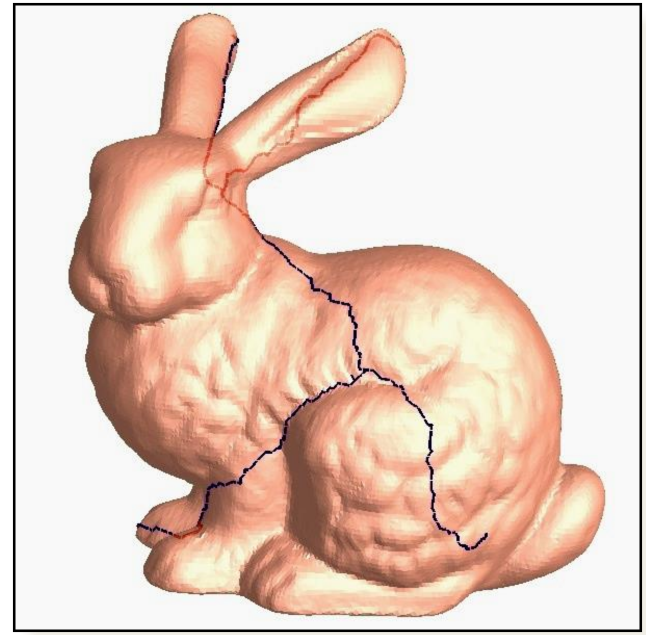


“Geometry images”, Gu et al., SIGGRAPH 2002
http://research.microsoft.com/en-us/um/people/hoppe/proj/gim/

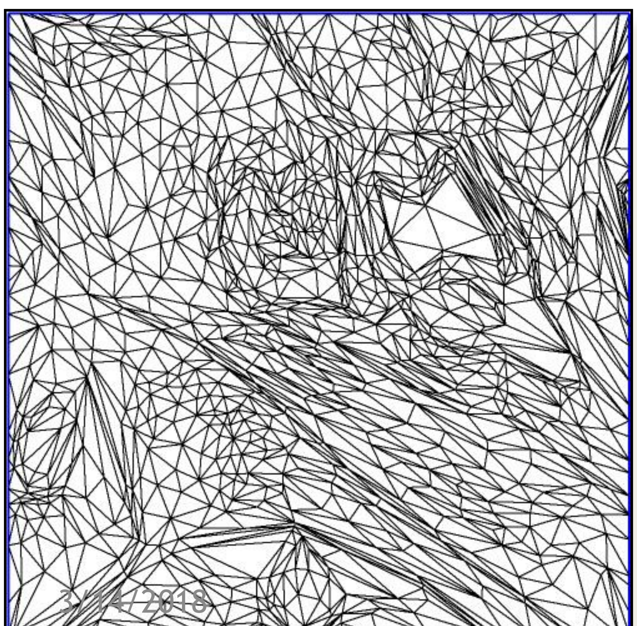
Geometry Images



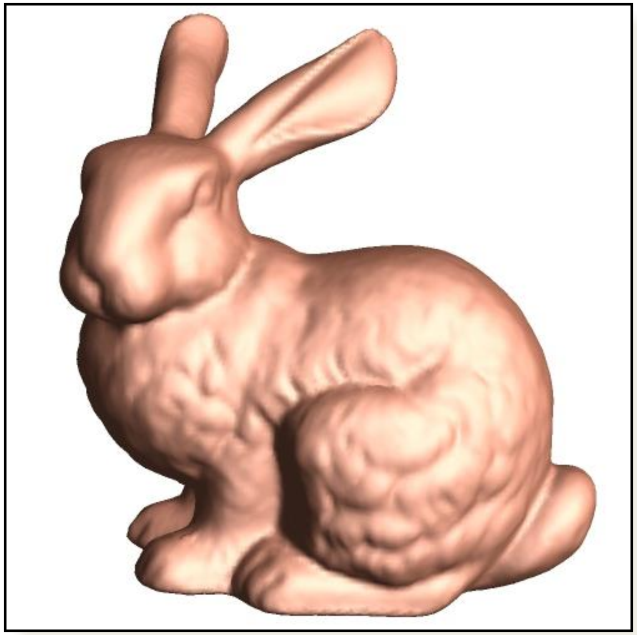
cut

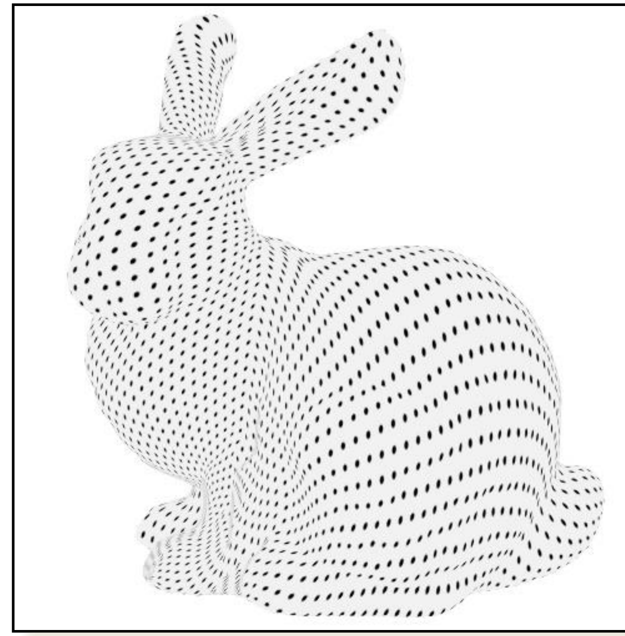


parametrize

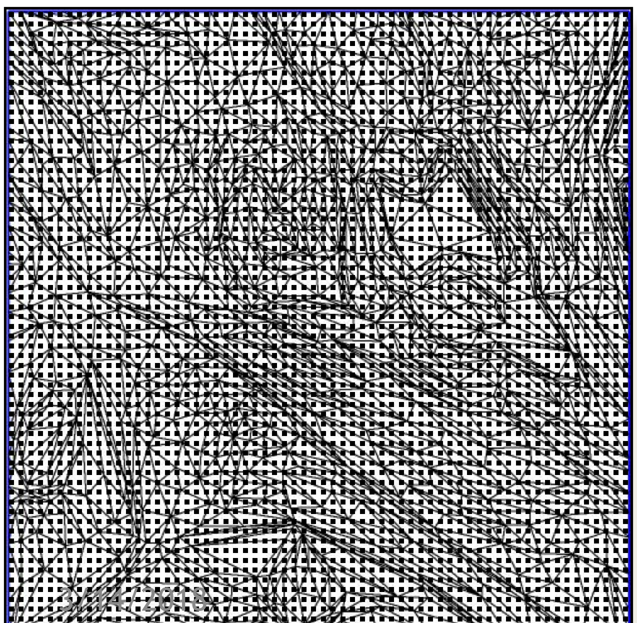


Geometry Images



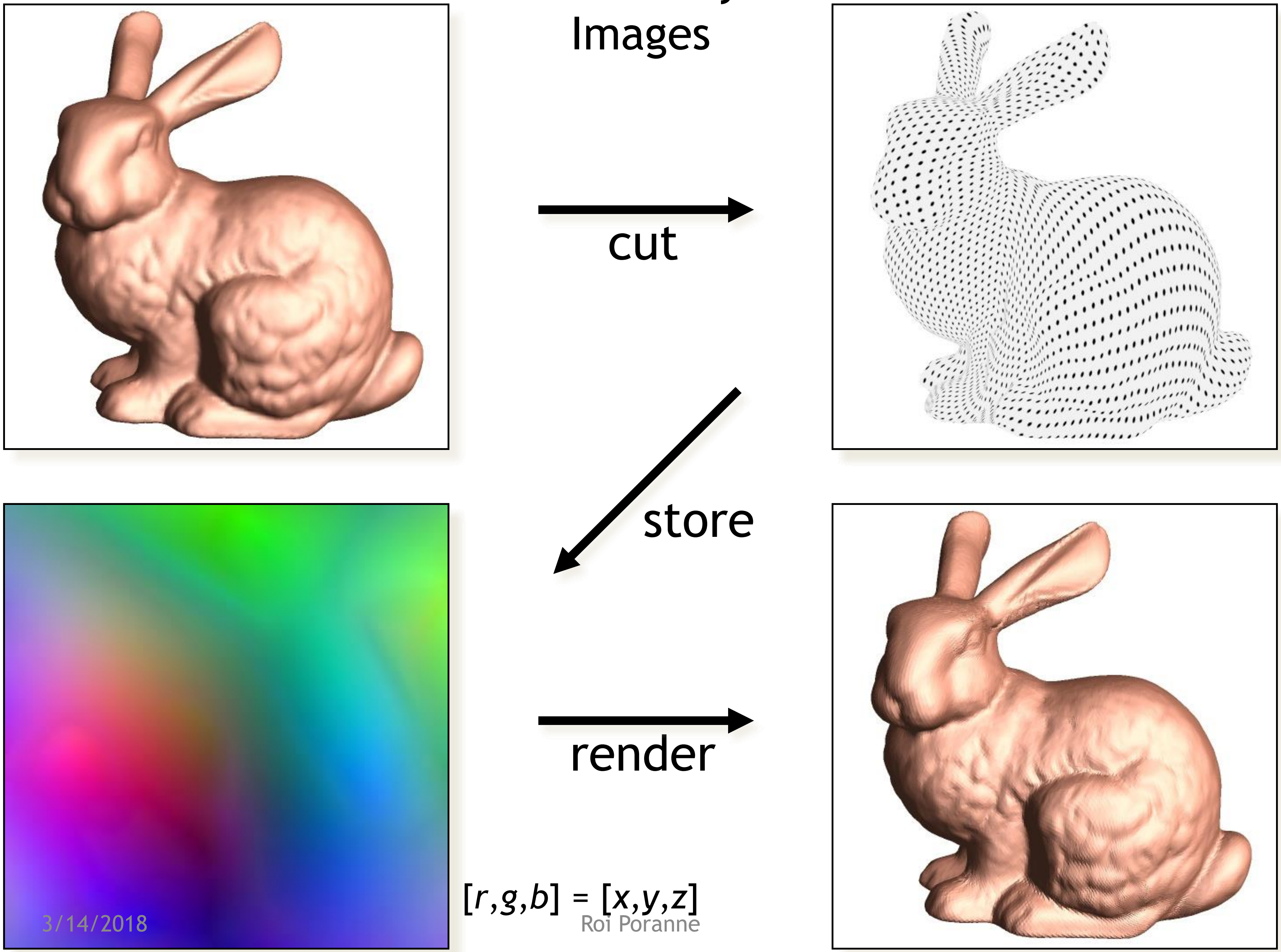


cut

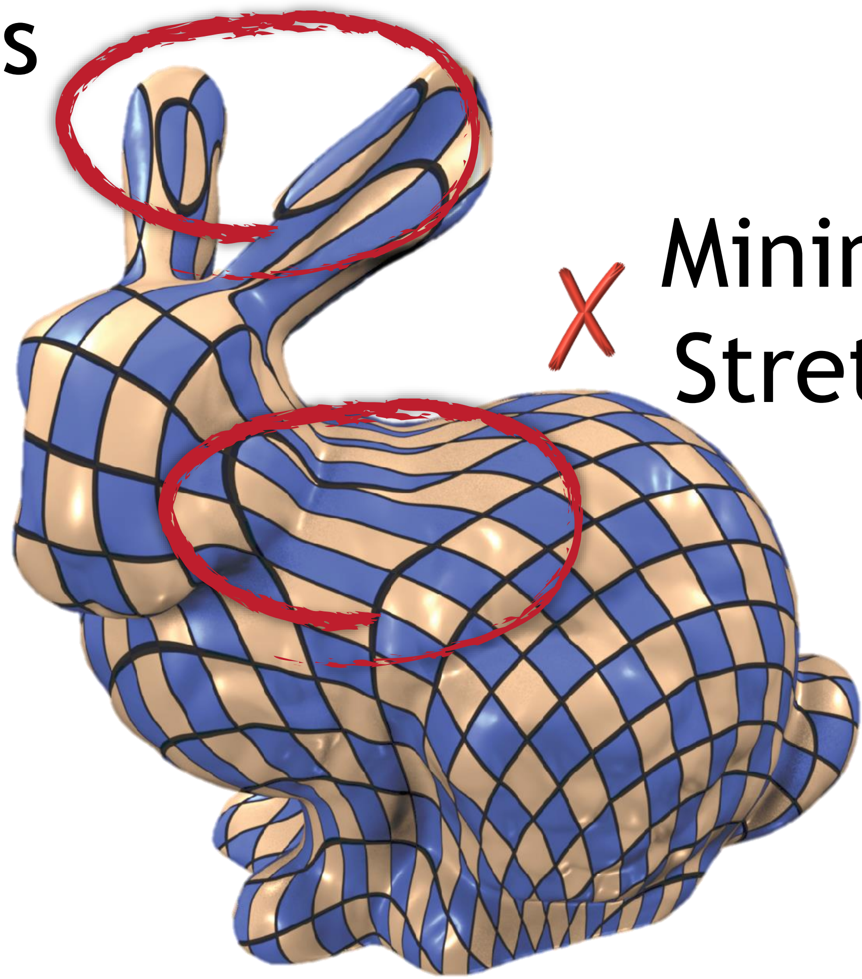


sample

Geometry Images

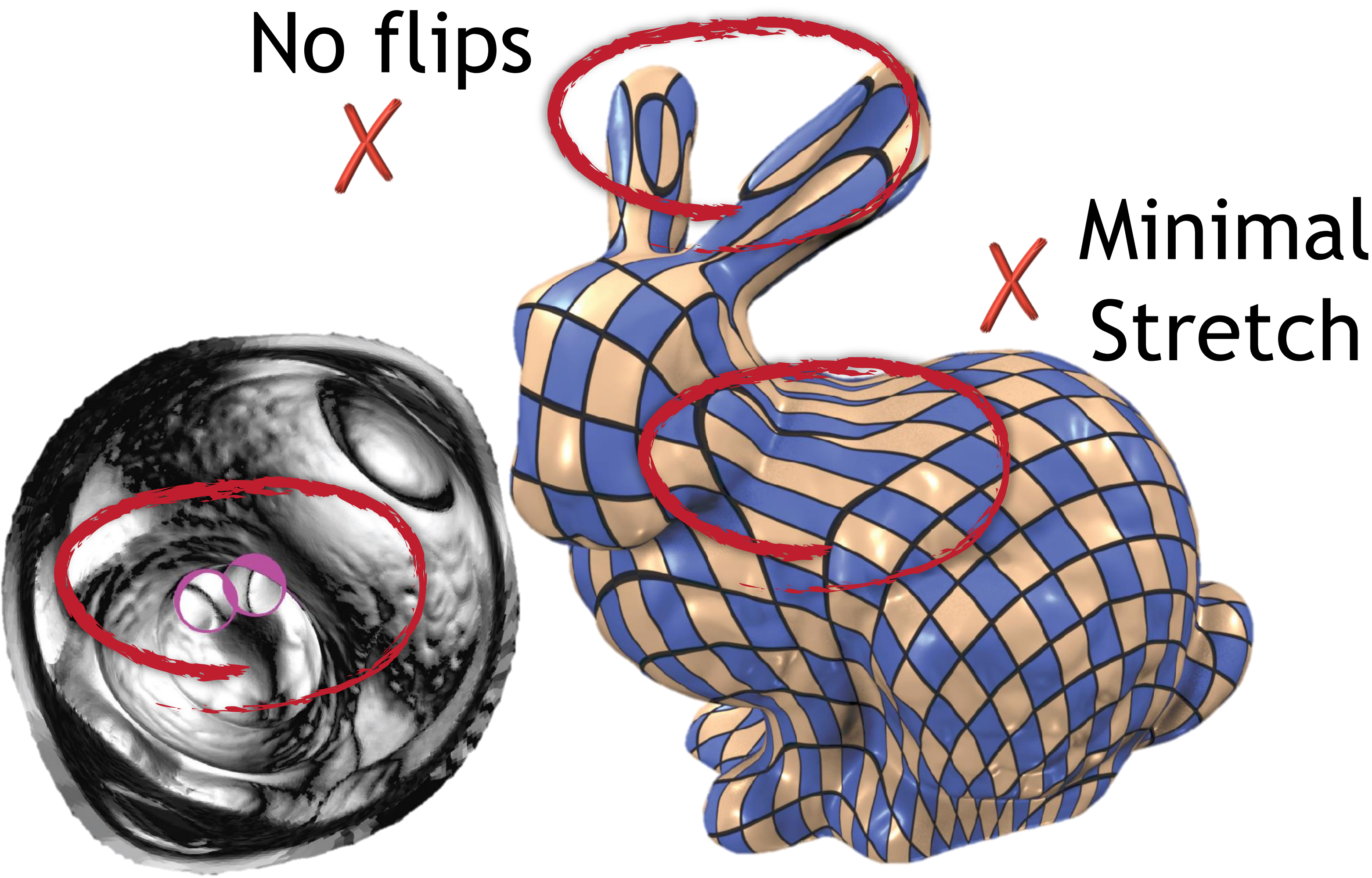


Good Parameterization



ETH zürich

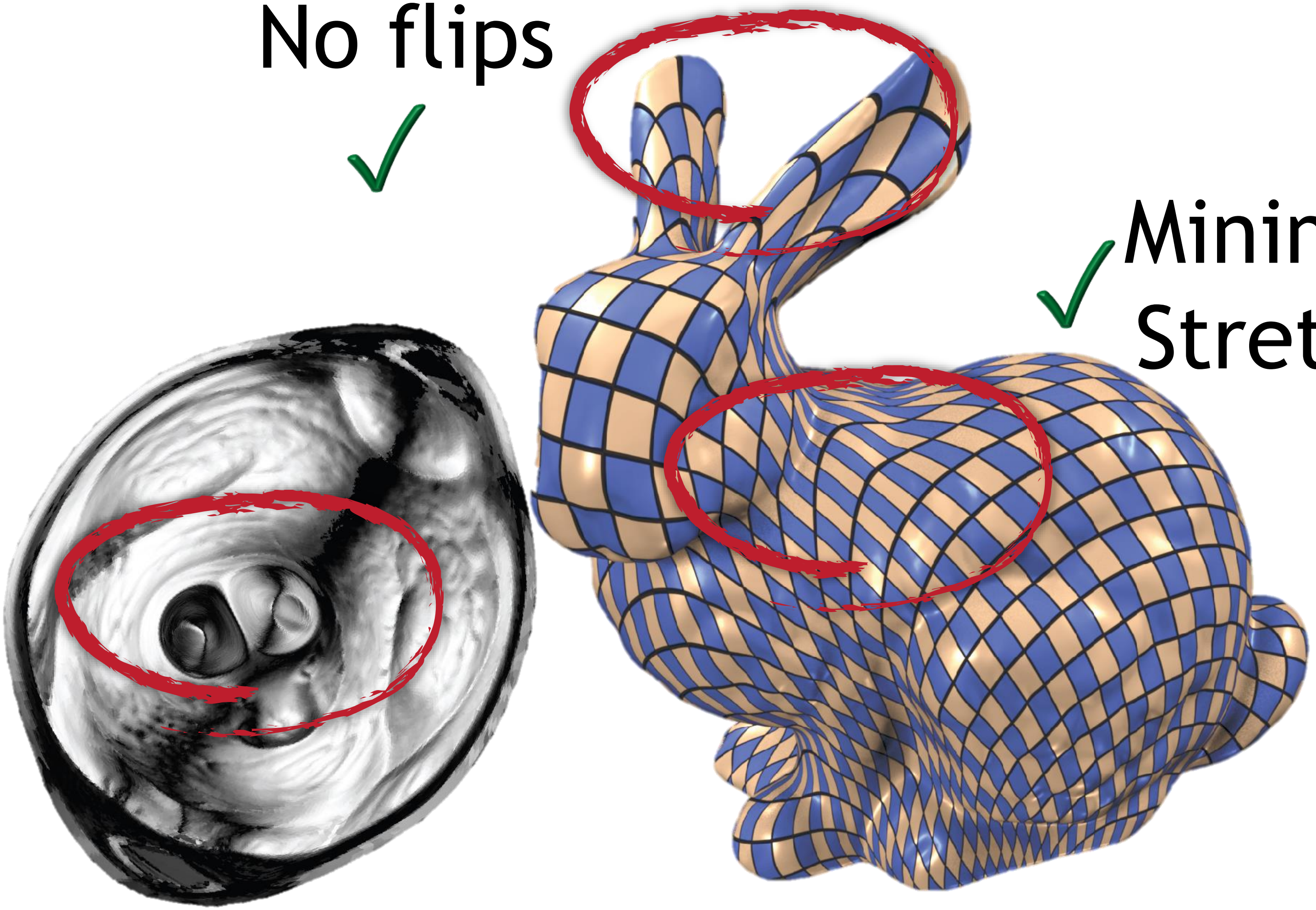
Good Parameterization



Good Parameterization

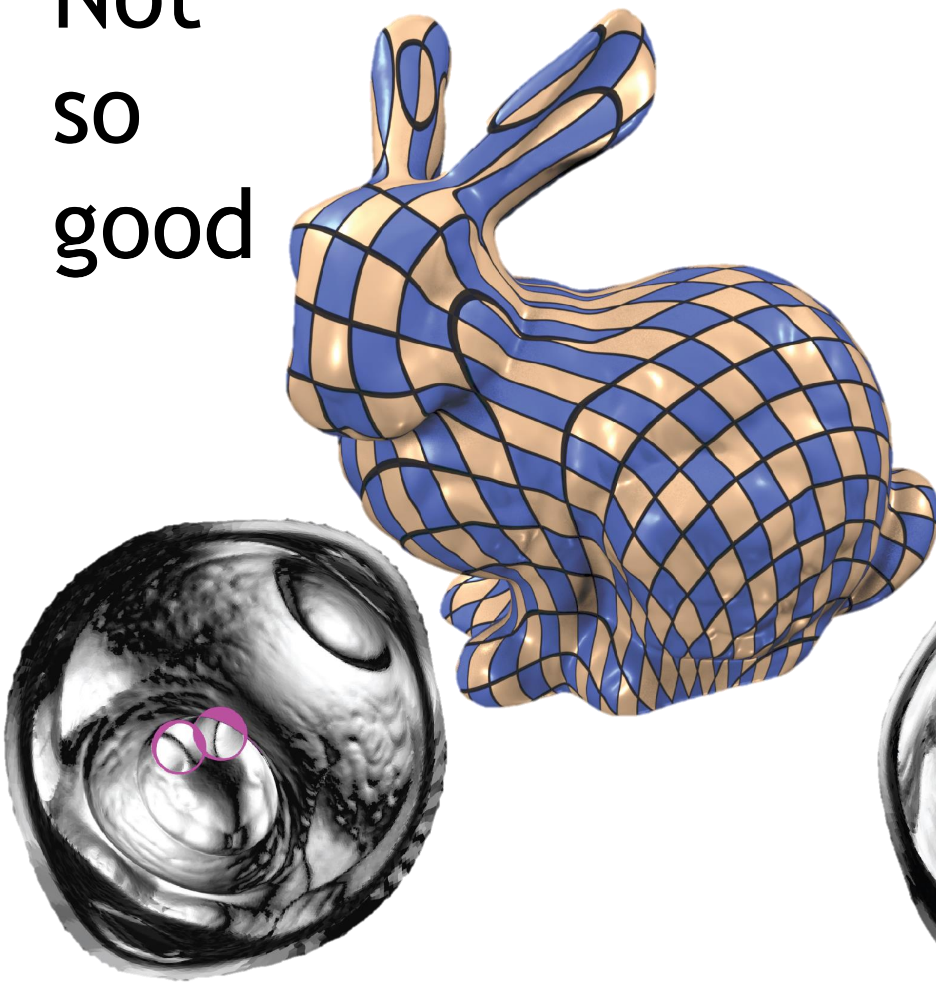
No flips ✓

✓ Minimal Stretch

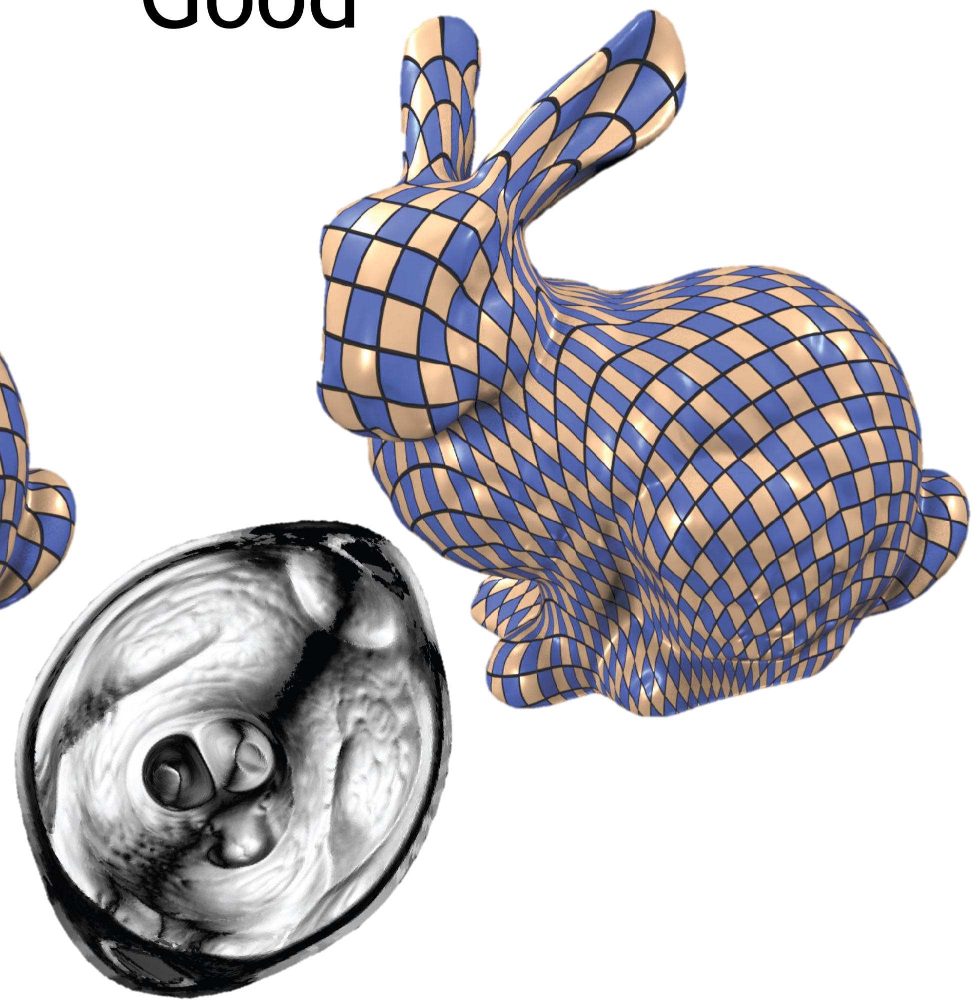


Good Parameterization

Not so good

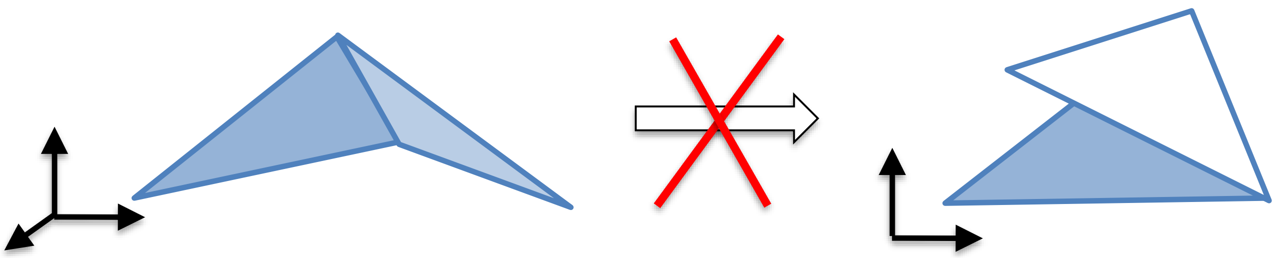


Good



Bijectivity

- Locally bijective (1-1 and onto): No triangles fold over.



- Globally bijective: locally bijective + no “distant” areas overlap

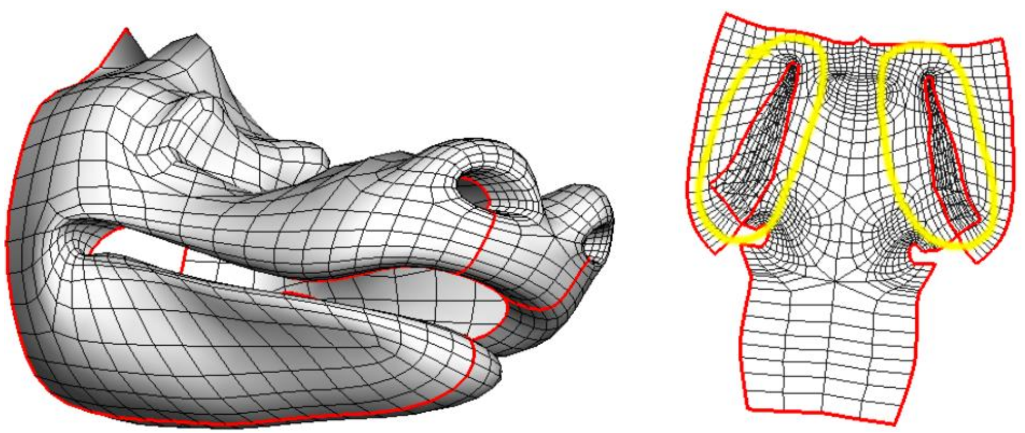
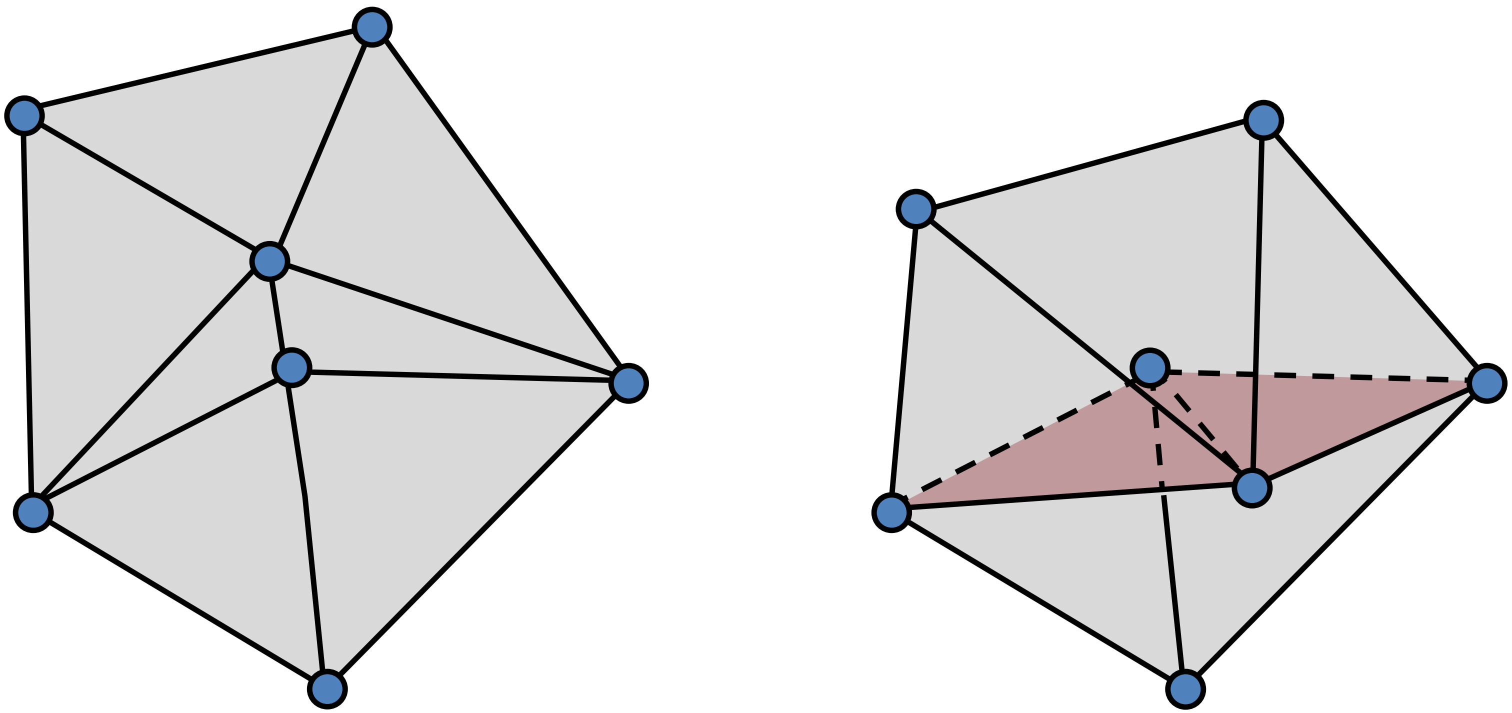
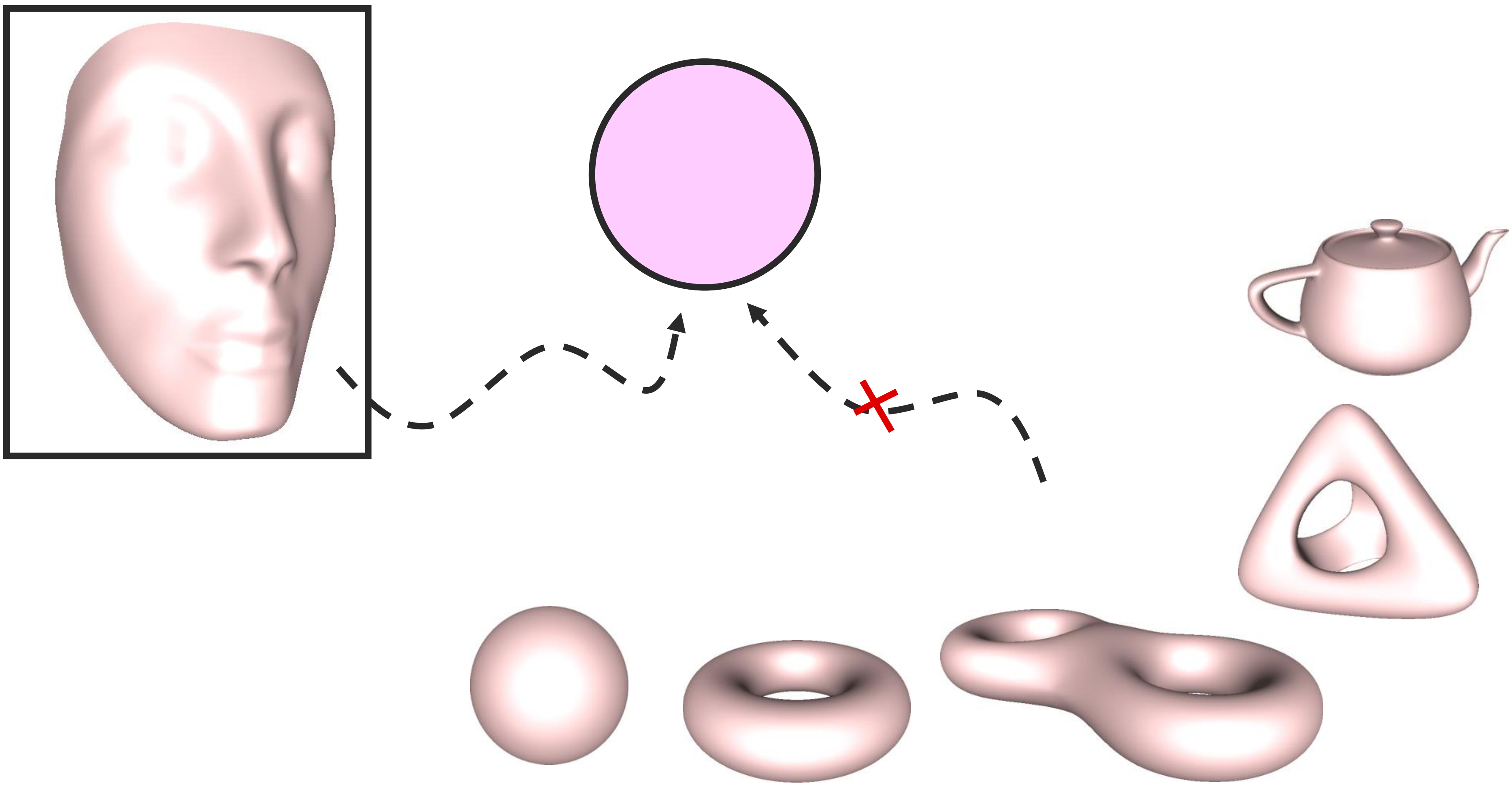


image from “Least Squares Conformal Maps”, Lévy et al., SIGGRAPH 2002

Local Foldovers

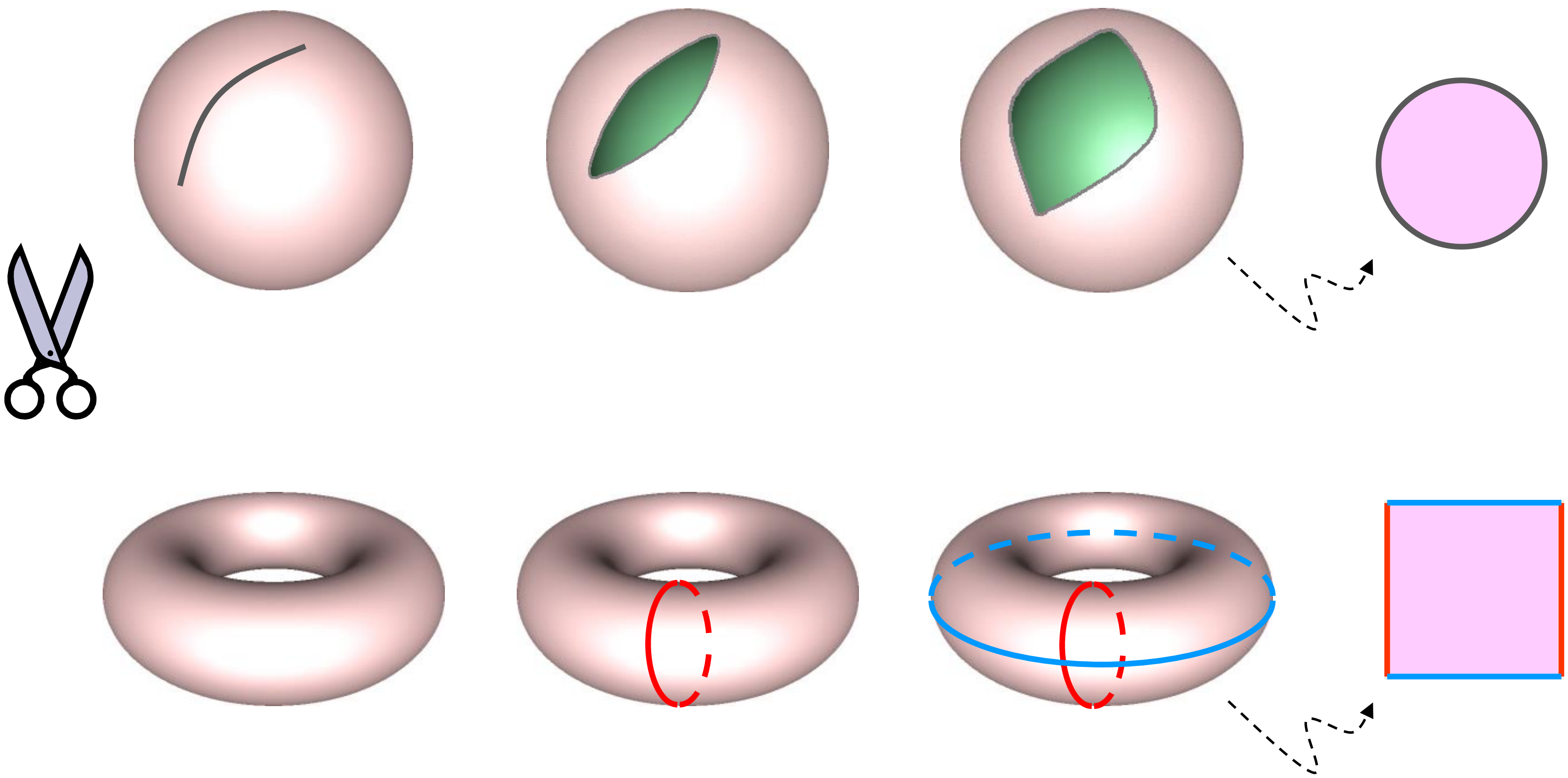


Bijectivity: Non-Disk Domains

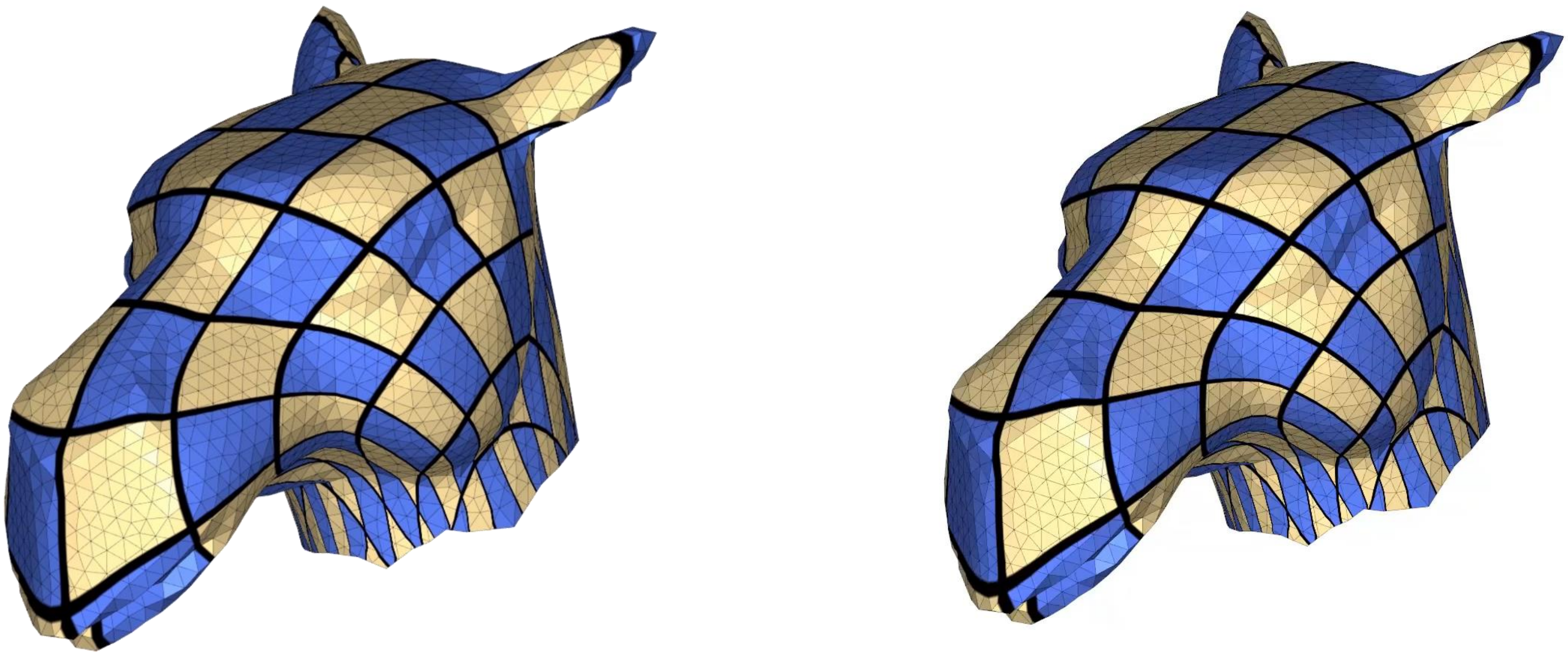


ETH zürich

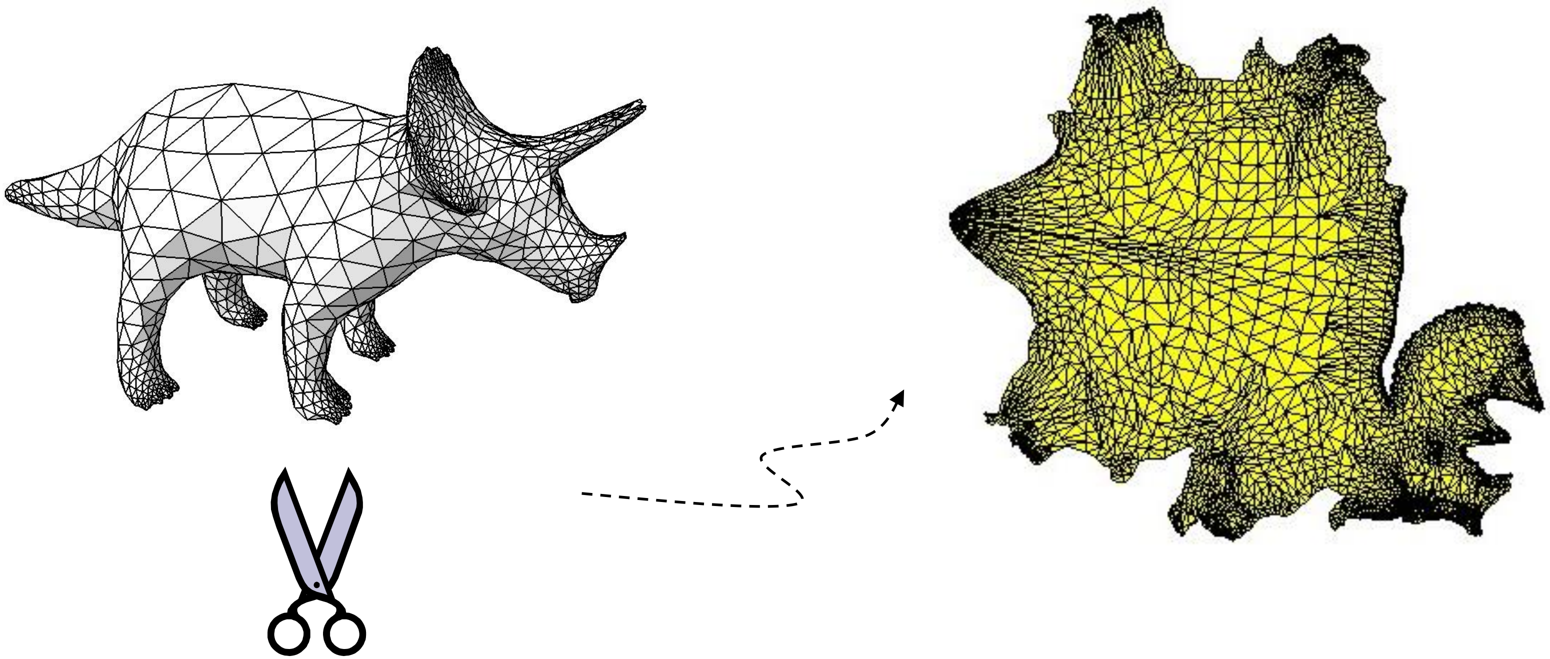
Mesh Cutting



ETH zürich

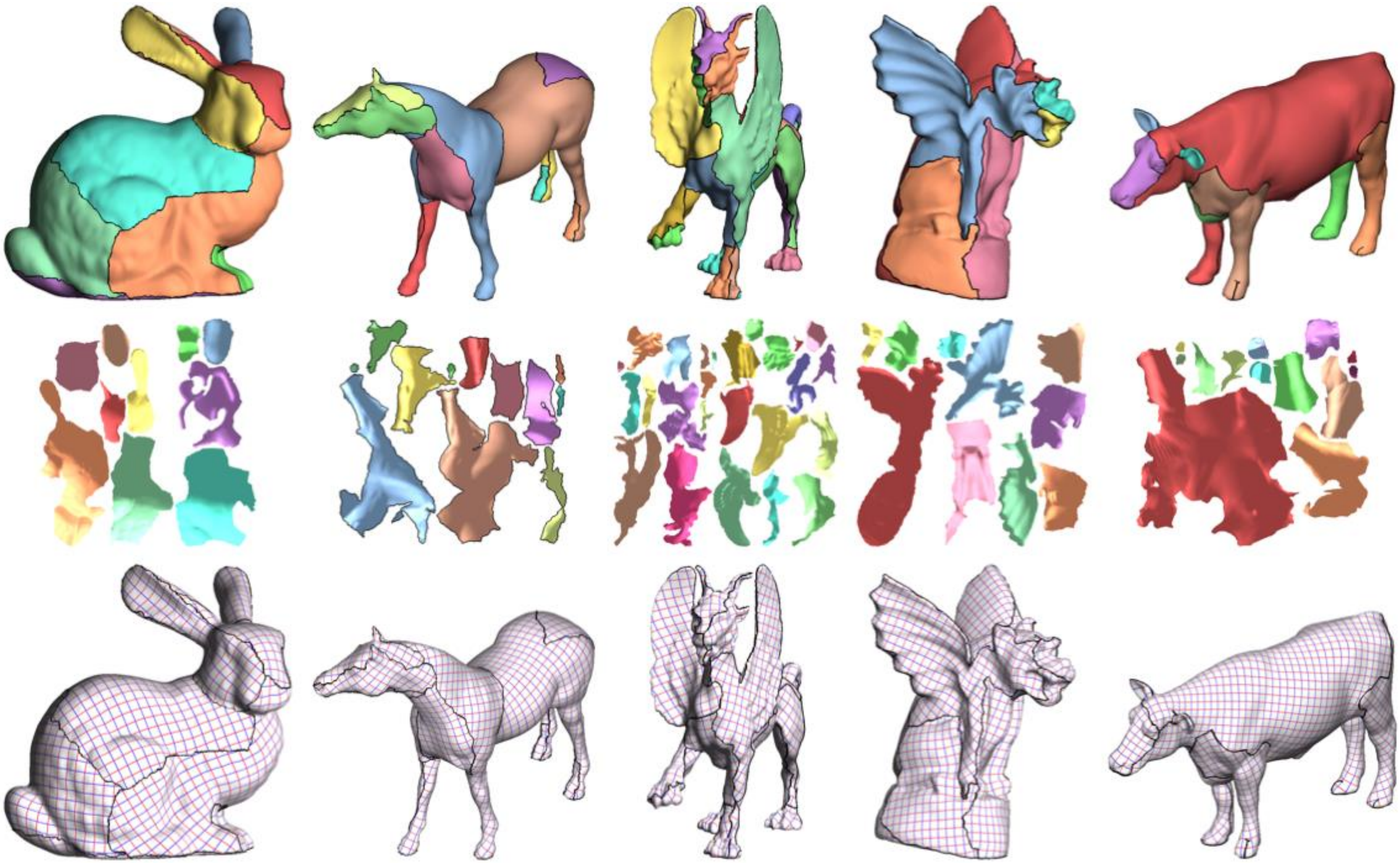


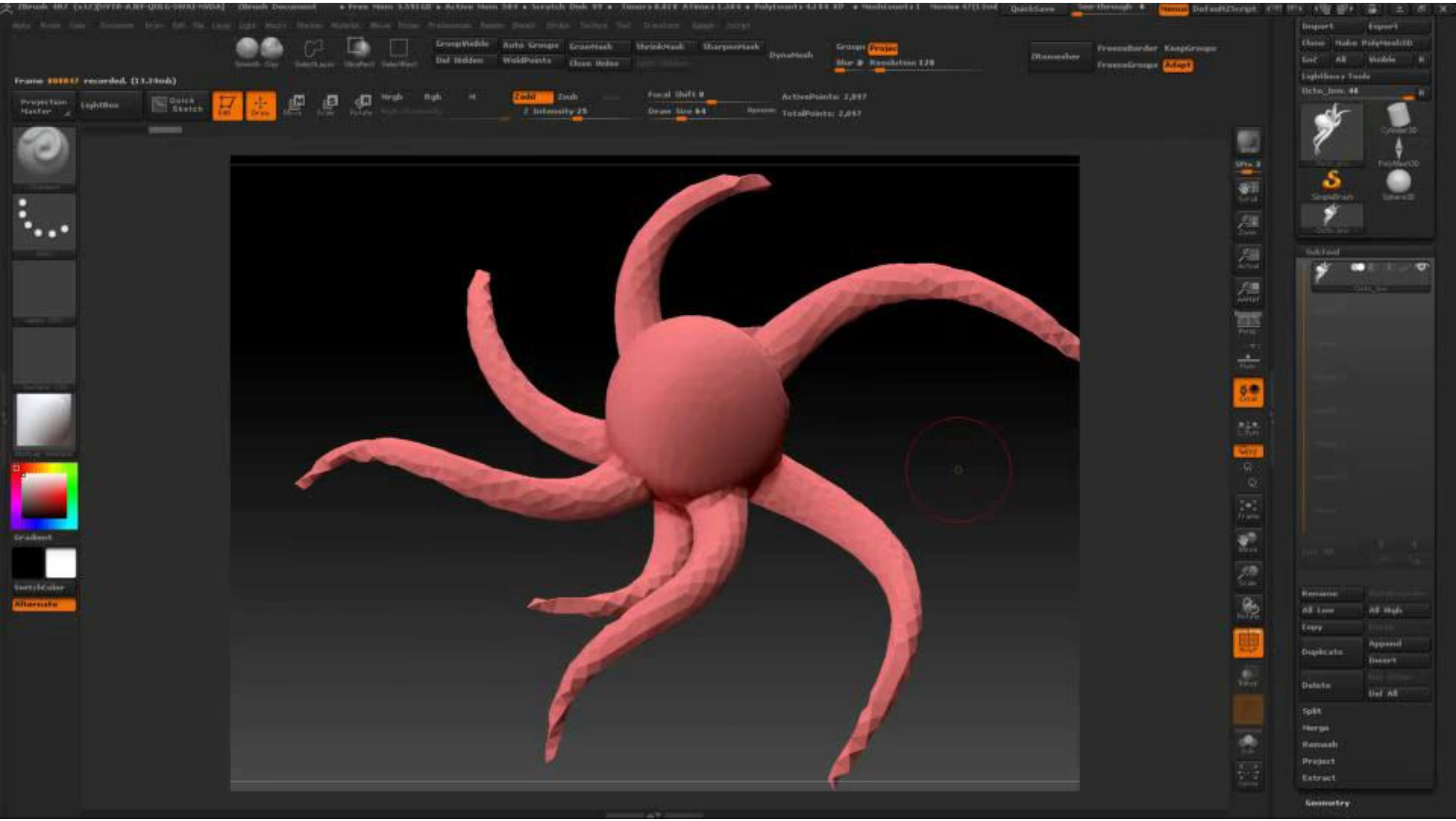
Mesh Cutting



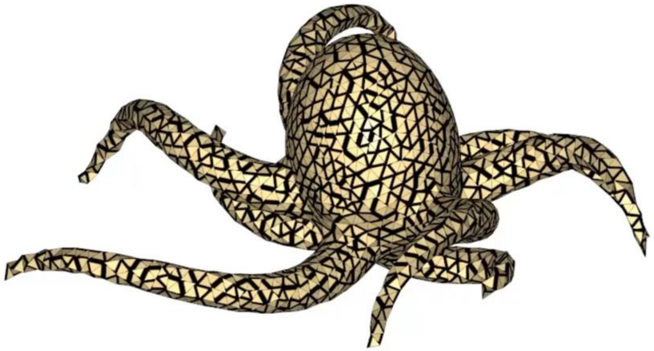
A. Sheffer, J. Hart:
Seamster: Inconspicuous Low-Distortion Texture Seam Layout, IEEE Vis 2002
http://www.cs.ubc.ca/~sheffa/papers/VIS02.pdf

Segmentation

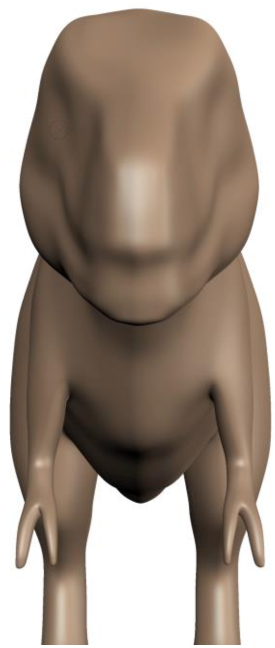



Projection Master
LightBox
Quick Sketch
Gradient
SwitchColor
Alternate
DynaMesh
Rename
All Low
All High
Copy
Duplicate
Append
Insert
Delete
Del All
Split
Merge
Remesh
Project
Extract
Geometry

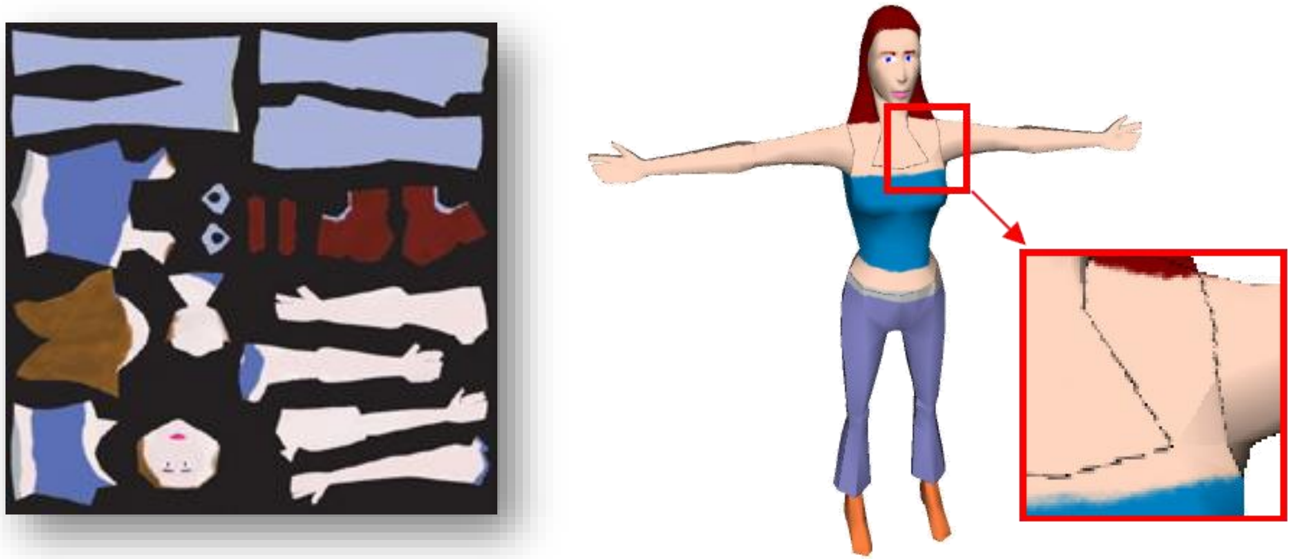

Lambda
Delta



3D painting



Texture artifacts

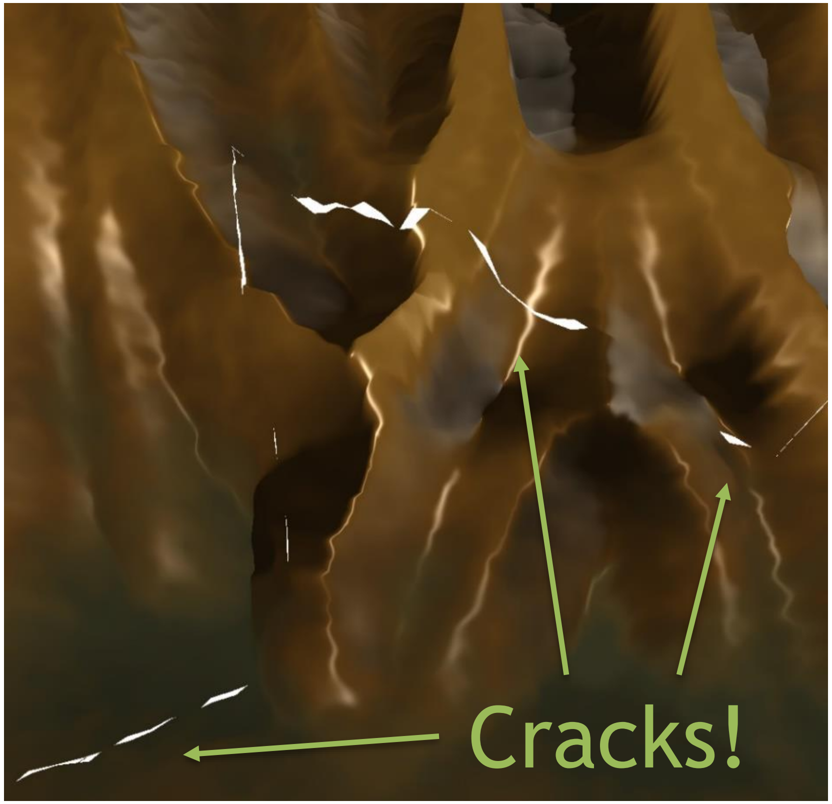


Texture artifacts

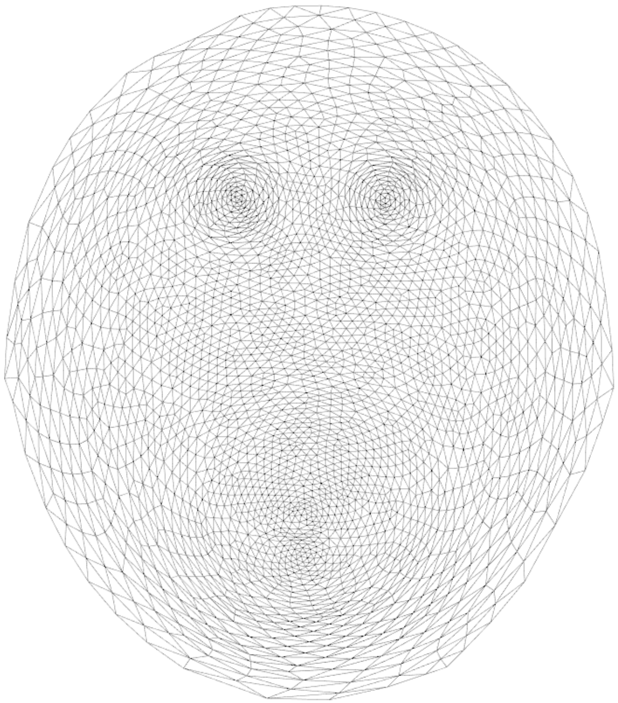
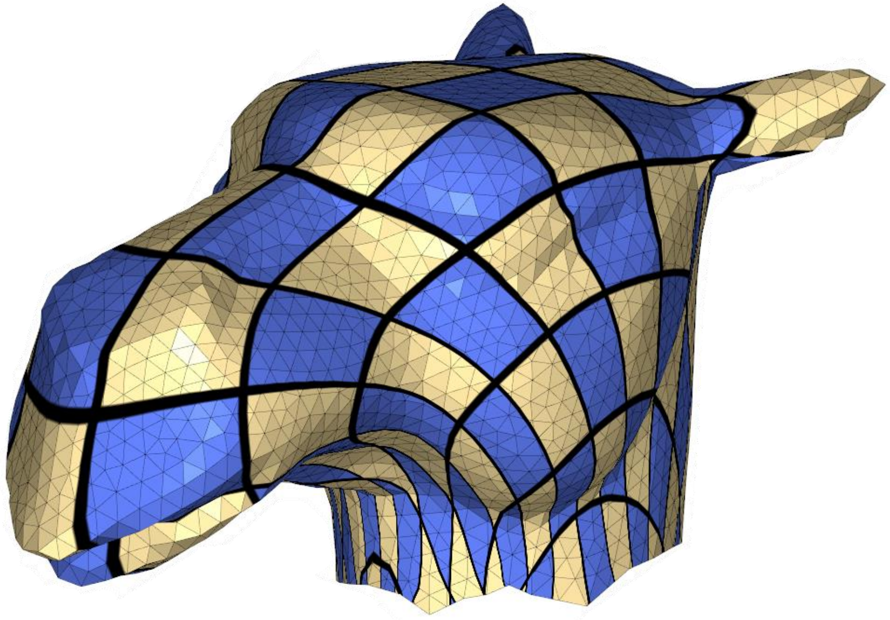
ETH zürich

Texture artifacts

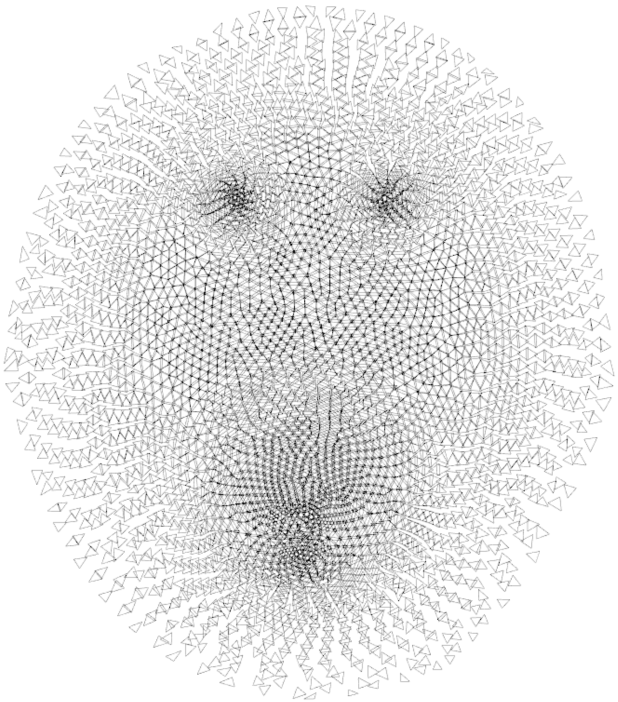
Texture artifacts

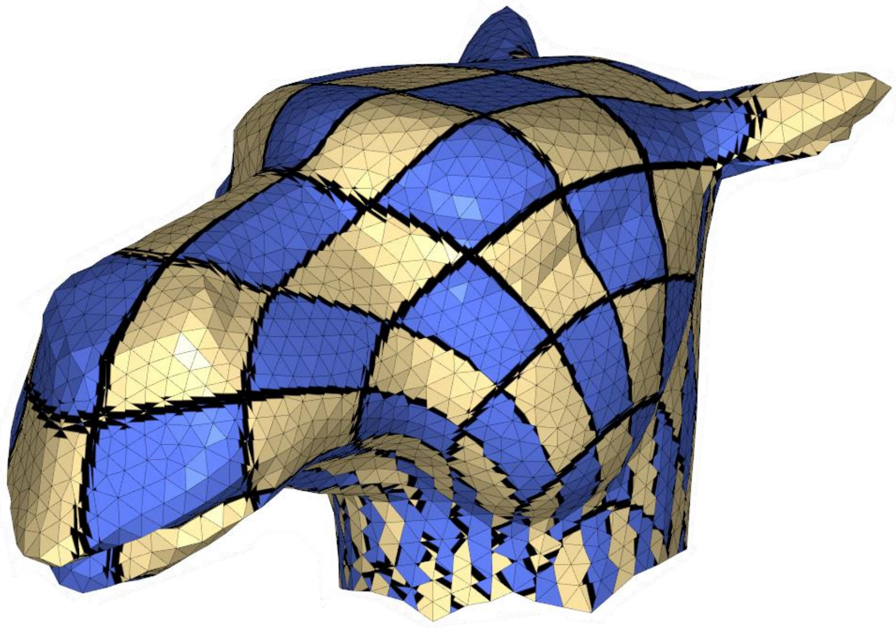


The balance

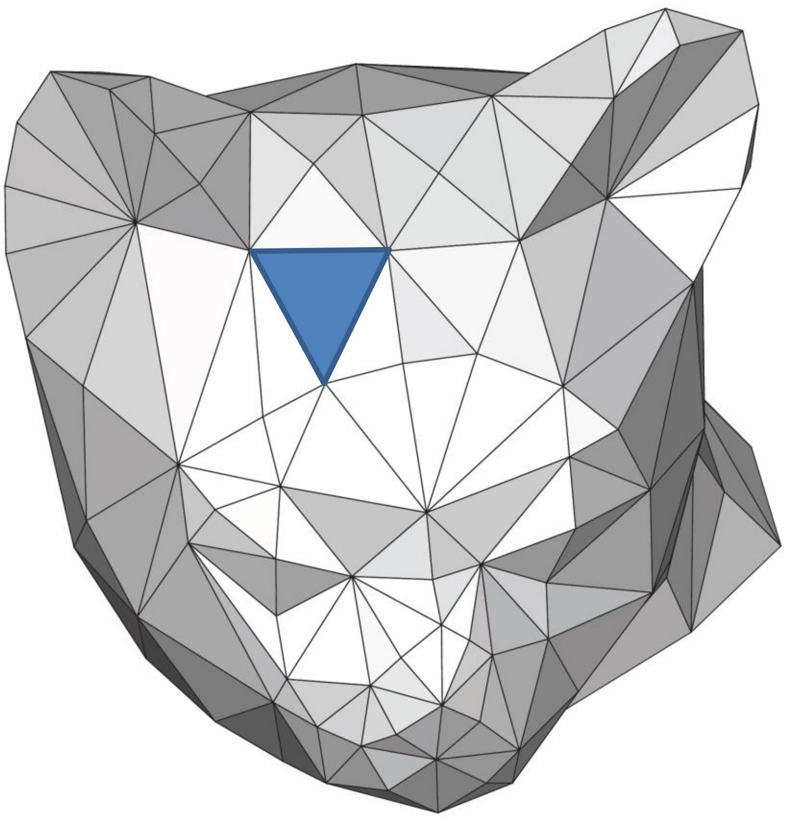
The balance

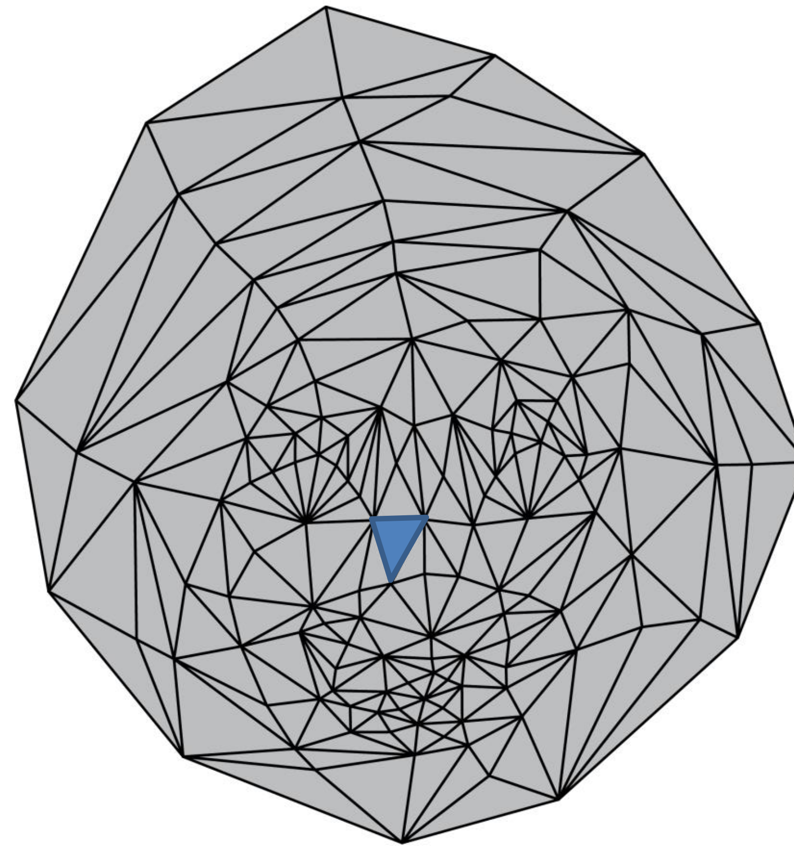




ETH zürich

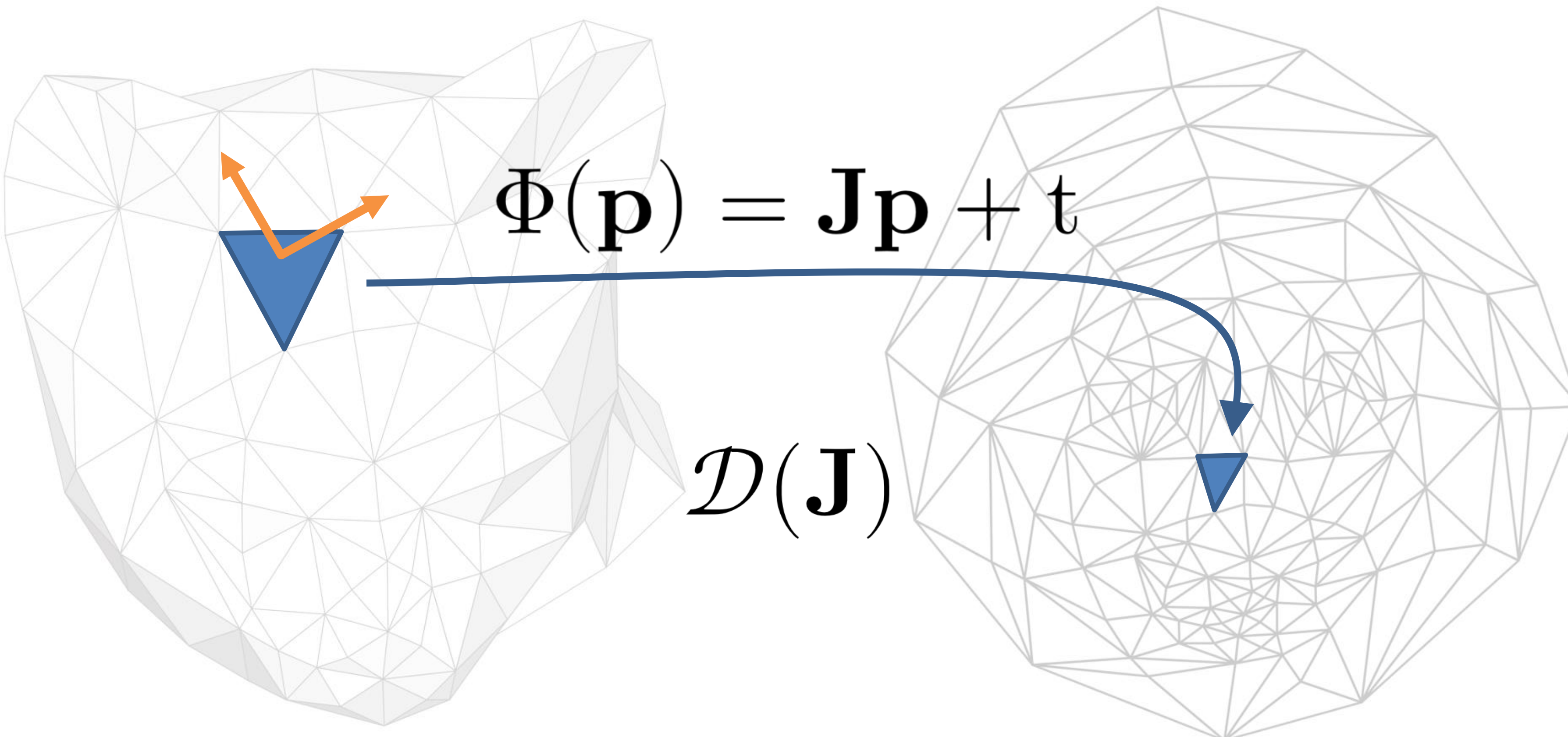
Distortion on Triangle Meshes



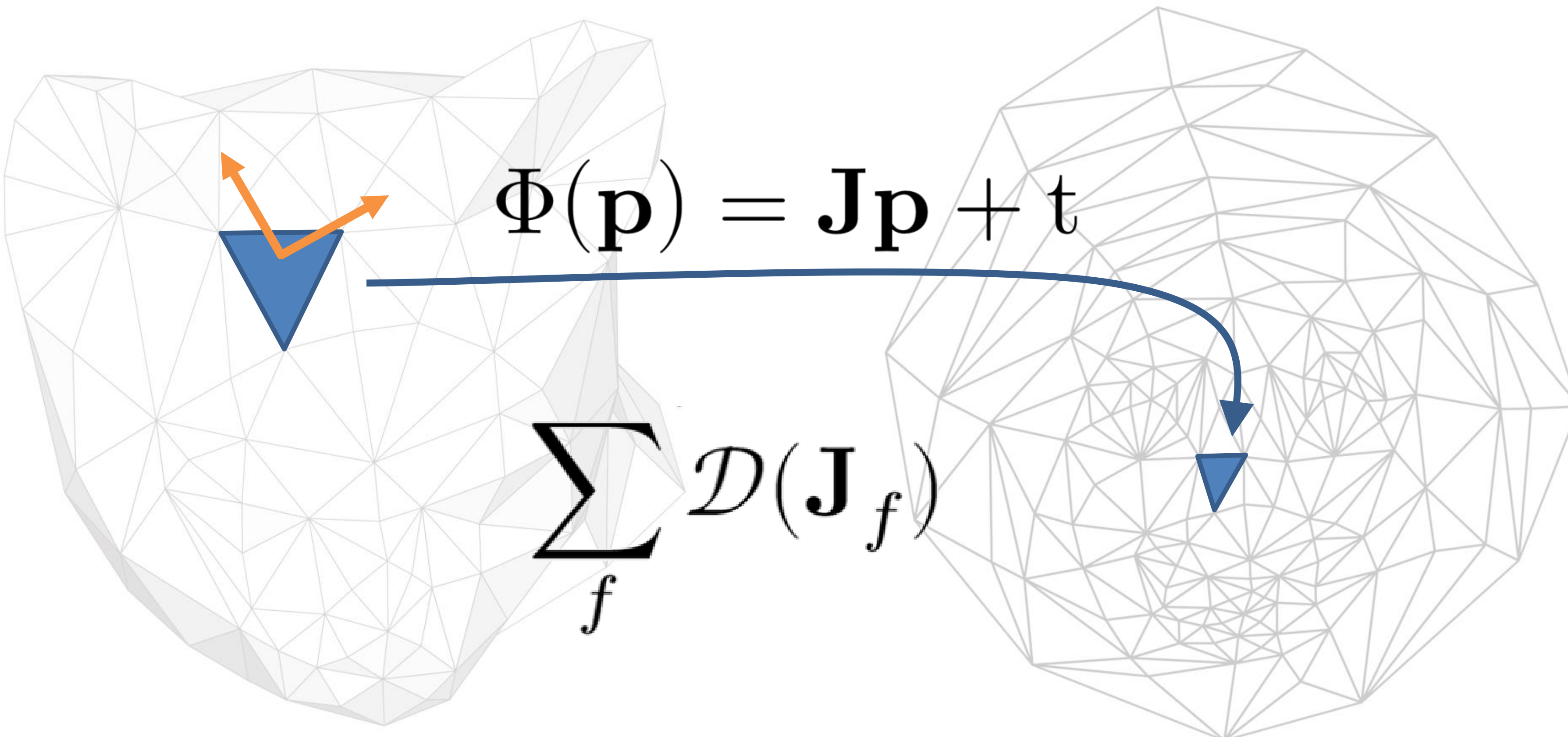


ETH zürich

Distortion on Triangle Meshes



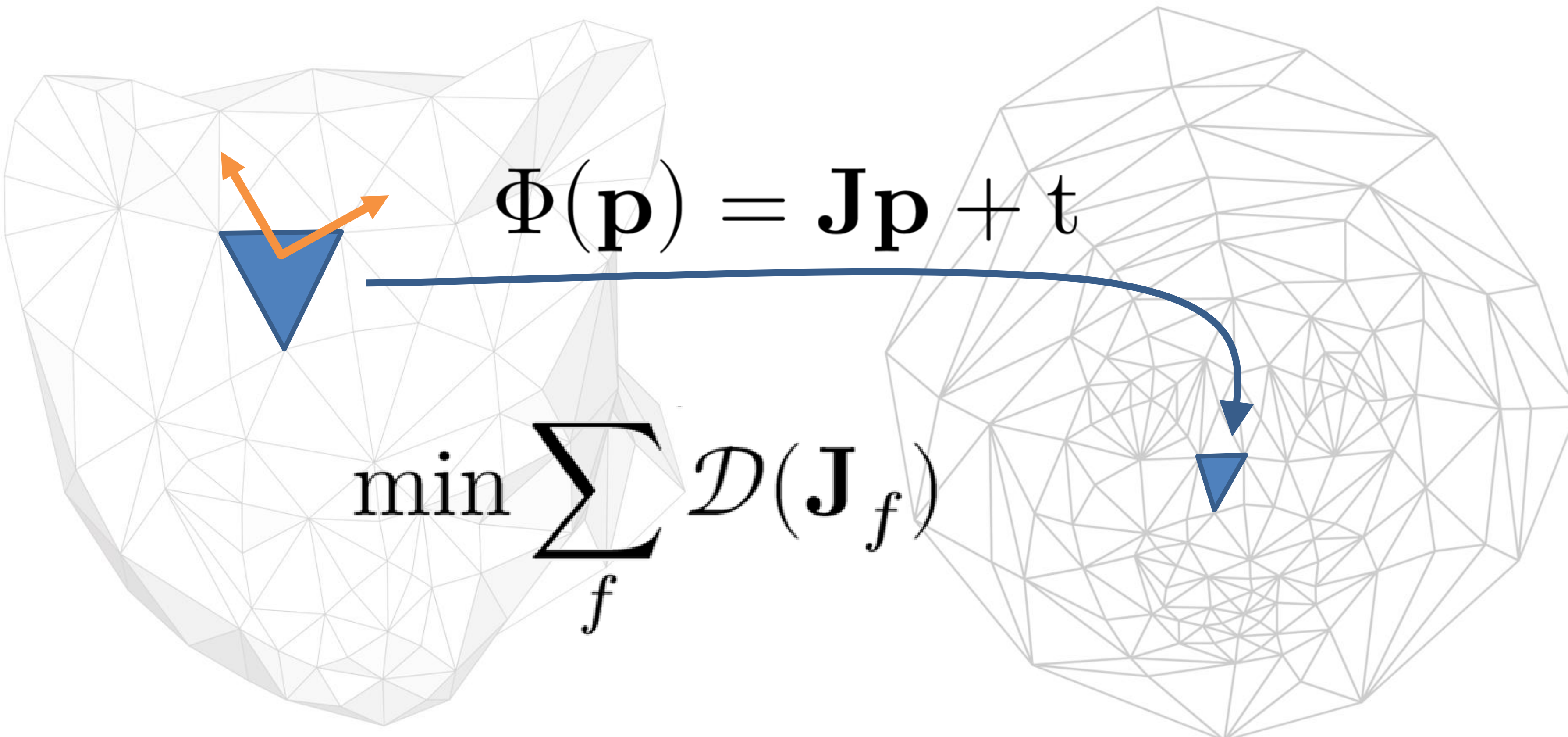
Distortion on Triangle Meshes



$$\Phi(\mathbf{p}) = \mathbf{J}\mathbf{p} + \mathrm{t}$$

$$\sum_f \mathcal{D}(\mathbf{J}_f)$$

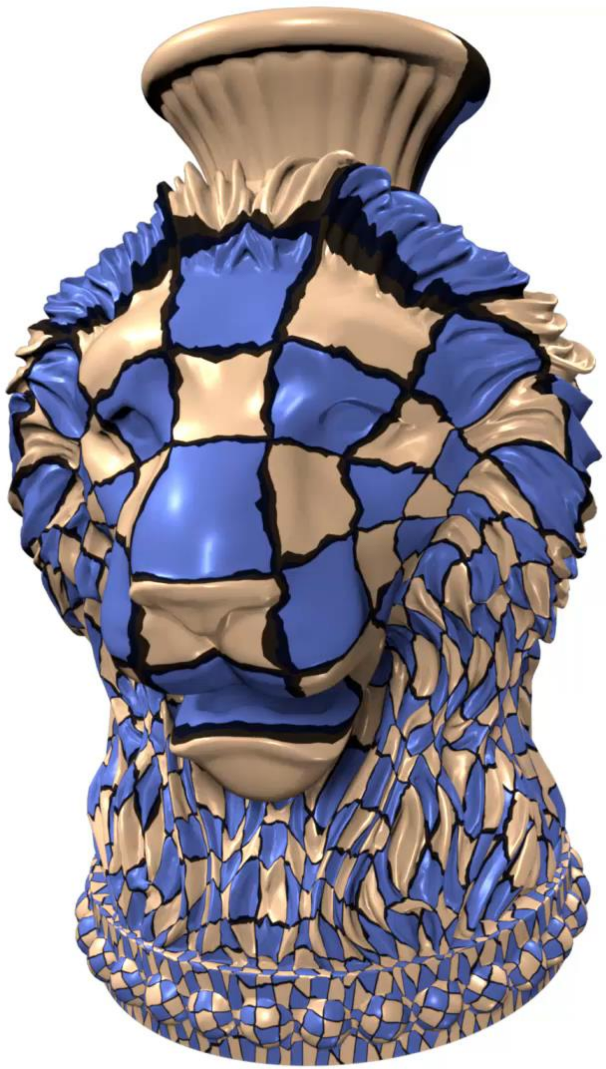
Distortion on Triangle Meshes

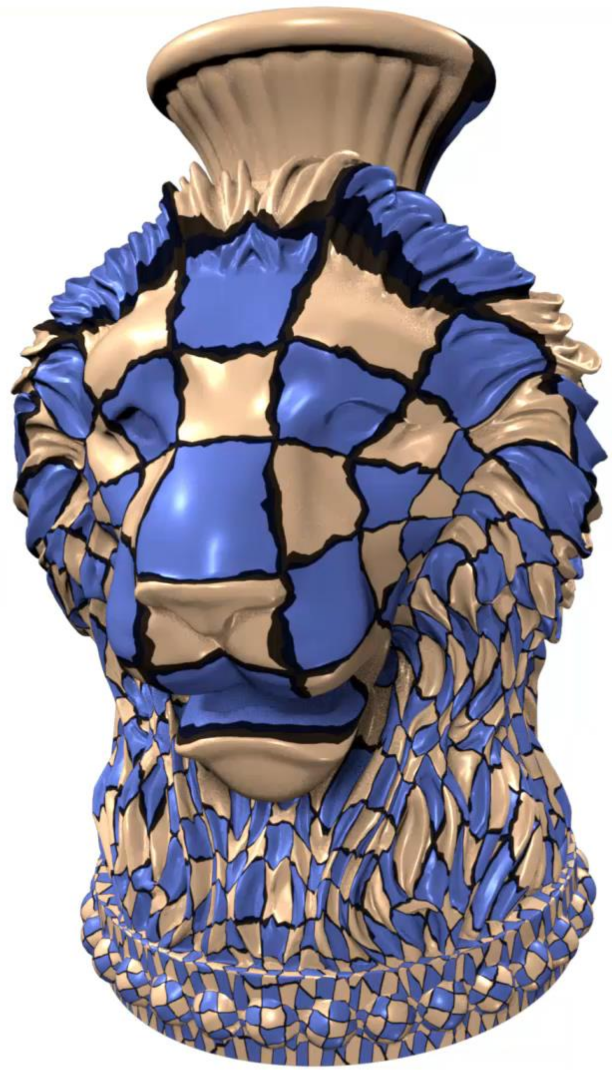


$$\Phi(\mathbf{p}) = \mathbf{J}\mathbf{p} + \mathrm{t}$$

$$\min \sum_{f} \mathcal{D}(\mathbf{J}_f)$$

100K Faces: 3s



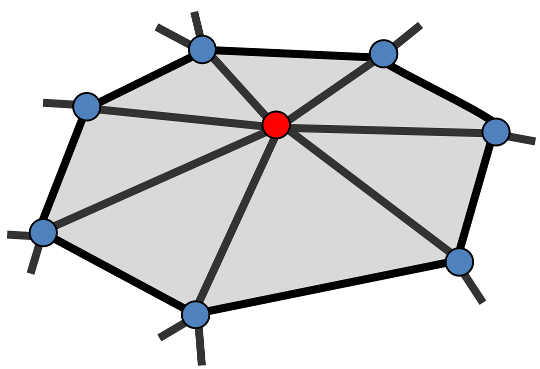


Conformal

ETH zürich

Harmonic Mapping – Idea

- Want to flatten the mesh $\rightarrow$ no curvature $\rightarrow$ Laplace operator gives zero.



$\mathbf{u} = (u,v)$ domain

$$\Delta(\mathbf{u}) = 0$$

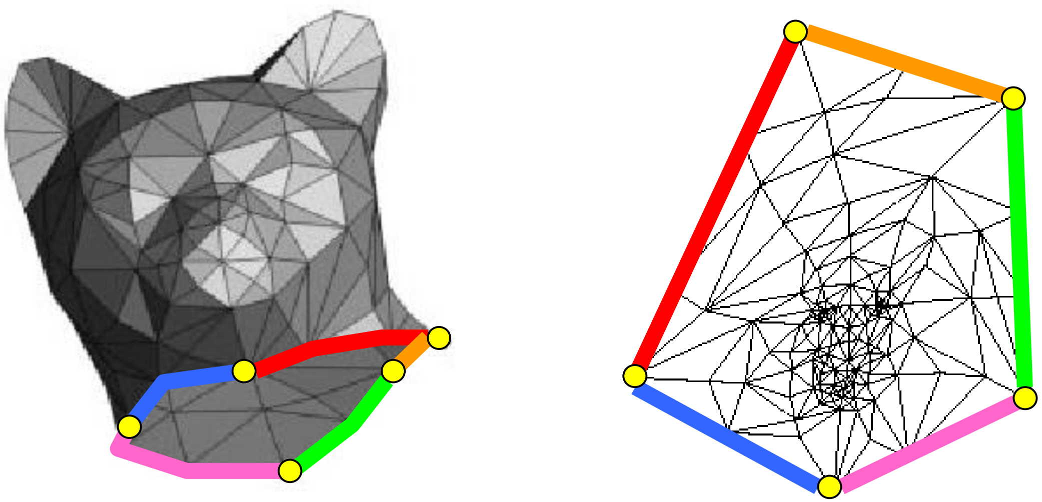
need boundary constraints to prevent trivial solution;

which Laplacian operator? (which weights?)

ETH zürich

Convex Mapping (Tutte, Floater)

- Boundary vertices are fixed
- Convex weights

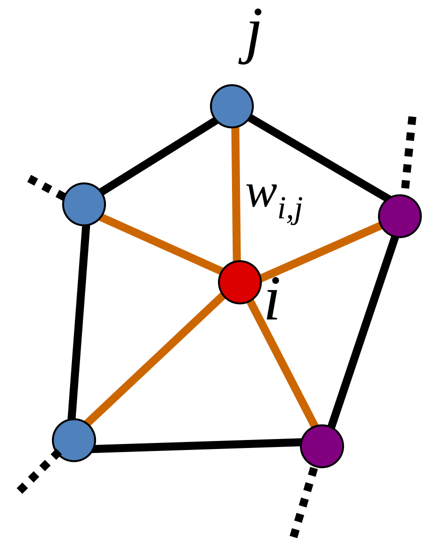


$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ - inner vertices

$\mathbf{v}_{n+1}, \ldots, \mathbf{v}_N$ - boundary vertices

Convex Mapping (Tutte, Floater)

- Boundary vertices are fixed
- Convex weights



$$\Delta(\mathbf{u}_i) = 0, \quad i = 1, \ldots, n$$

$$L(\mathbf{u}_i) = \frac{1}{W_i} \sum_{j \in \mathcal{N}(i)} w_{ij}(\mathbf{u}_j - \mathbf{u}_i) = 0, \quad i = 1, \ldots, n$$

$$\boxed{w_{ij} > 0}$$

ETH zürich

Convex Mapping (Tutte, Floater)

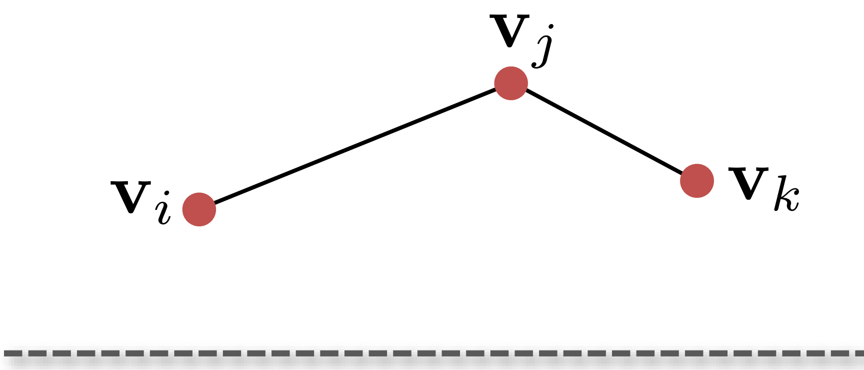
- Solve the linear system

$$L\mathbf{u} = 0 \qquad \mathbf{u} \in \mathbb{R}^{n \times 2}$$

- The values of the boundary vertices are known and thus substituted (transfer to right-hand side)

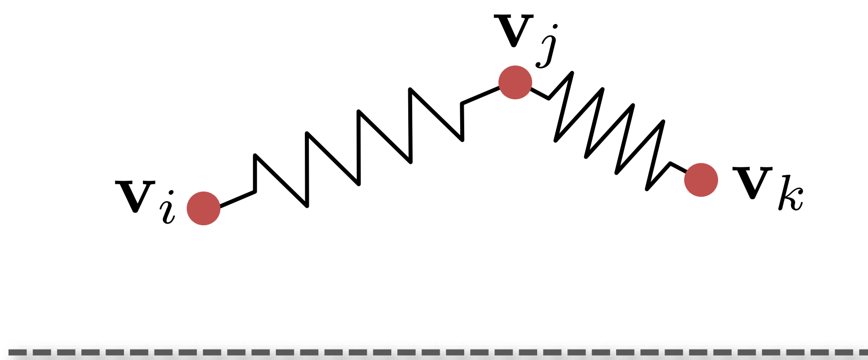
Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane



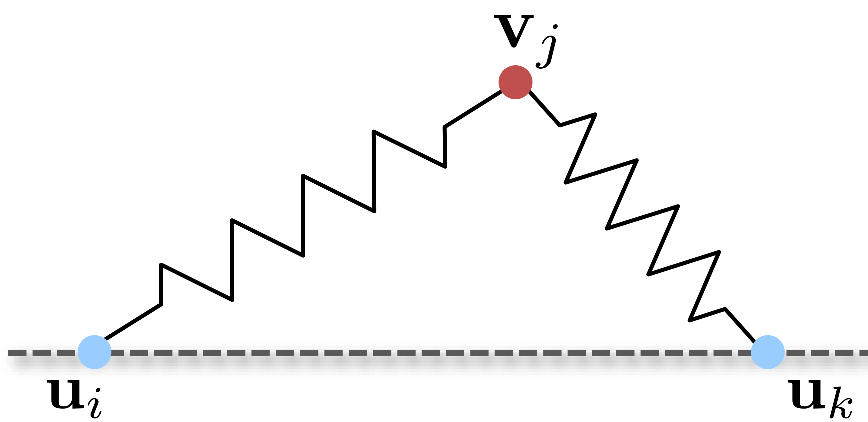
Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane



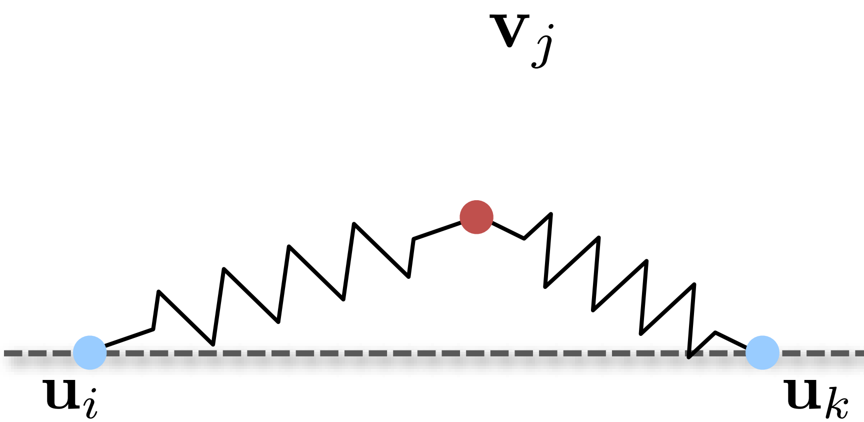
Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane



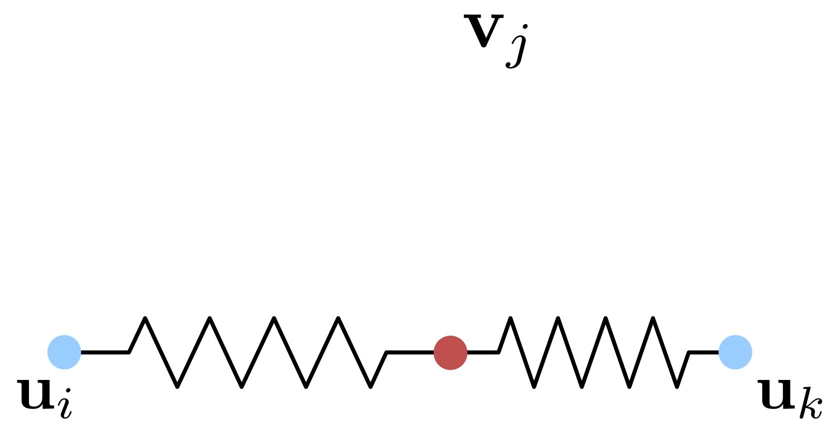
Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane



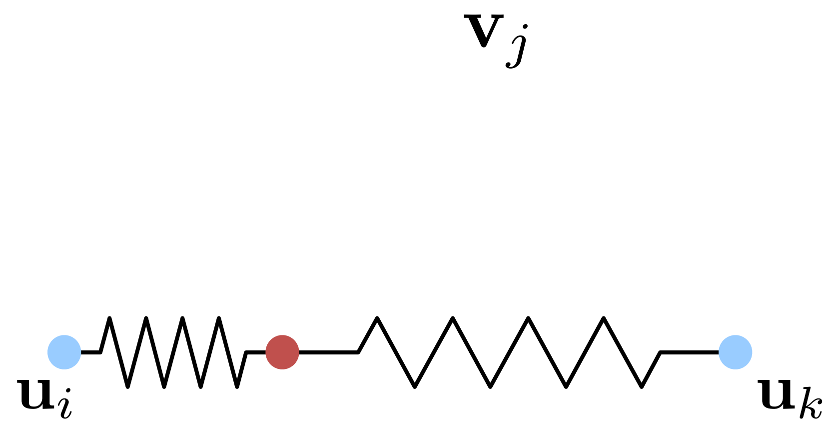
Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane



Harmonic Mapping

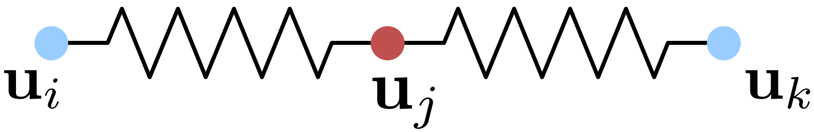
- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane



ETH zürich

Harmonic Mapping

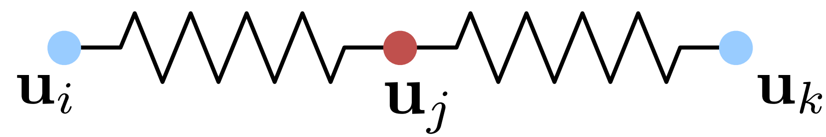
- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane



ETH zürich

Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane
- Spring energy:

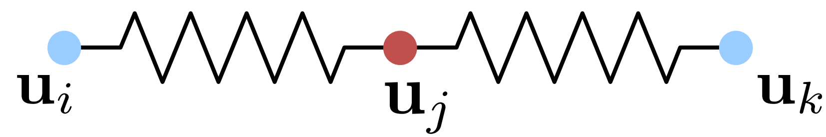


$$\frac{1}{2} k_{i,j} \|\mathbf{u}_i - \mathbf{u}_j\|^2$$

$$\mathbf{u}_i, \mathbf{u}_j \in \mathbb{R}^2$$

Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane



- Total spring energy of the flattened mesh:

$$E(\mathbf{u}_1, \ldots, \mathbf{u}_n) = \sum_{(i,j)\in\mathcal{E}} \frac{1}{2} k_{i,j} \|\mathbf{u}_i - \mathbf{u}_j\|^2$$

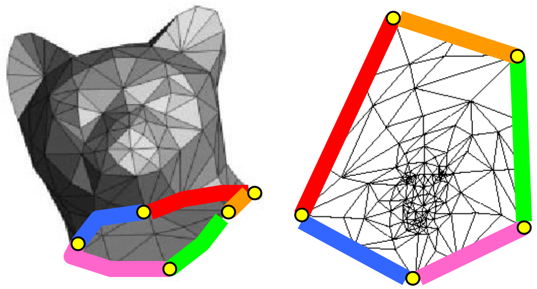
ETH zürich

Demo

- Libigl Tutorial 501

Minimizing Spring Energy

$$E(\mathbf{u}_1, \ldots, \mathbf{u}_n) = \sum_{(i,j)\in\mathcal{E}} \frac{1}{2} k_{i,j} \|\mathbf{u}_i - \mathbf{u}_j\|^2$$



$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ - inner vertices

$\mathbf{v}_{n+1}, \ldots, \mathbf{v}_N$ - boundary vertices

$$\frac{\partial E(\mathbf{u}_1, \ldots, \mathbf{u}_n)}{\partial \mathbf{u}_i} = \sum_{j\in\mathcal{N}(i)} k_{i,j}(\mathbf{u}_i - \mathbf{u}_j) = 0$$

$$\sum_{j\in\mathcal{N}(i)\cap\mathcal{B}} k_{i,j}\mathbf{u}_i + \underbrace{\sum_{j\in\mathcal{N}(i)\setminus\mathcal{B}} k_{i,j}(\mathbf{u}_i - \mathbf{u}_j)}_{\text{unknown flat vertex positions}} = \sum_{j\in\mathcal{N}(i)\cap\mathcal{B}} k_{i,j}\underbrace{\mathbf{u}_j}_{\text{known fixed boundary positions}}$$

unknown
flat vertex
positions

known fixed
boundary
positions

ETH zürich

Minimizing Spring Energy

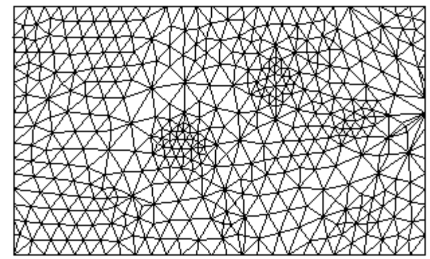
- Sparse linear system of n equations to solve!

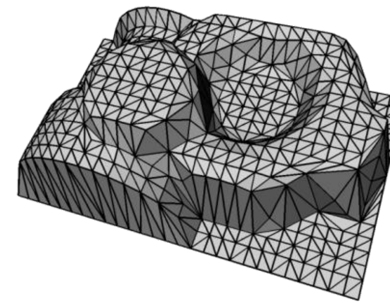
$$\boxed{\sum_{j\in\mathcal{N}(i)\cap\mathcal{B}} k_{i,j}\mathbf{u}_i + \sum_{j\in\mathcal{N}(i)\setminus\mathcal{B}} k_{i,j}(\mathbf{u}_i - \mathbf{u}_j) = \sum_{j\in\mathcal{N}(i)\cap\mathcal{B}} k_{i,j}\mathbf{u}_j}$$

$$\begin{pmatrix} \sum_j k_{i,j} & * & \cdots & -k_{i,j} \\ * & \sum_j k_{i,j} & * & \vdots \\ \vdots & * & \ddots & * \\ -k_{j,i} & \cdots & * & \sum_j k_{i,j} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_n \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{u}}_1 \\ \bar{\mathbf{u}}_2 \\ \vdots \\ \bar{\mathbf{u}}_n \end{pmatrix}$$

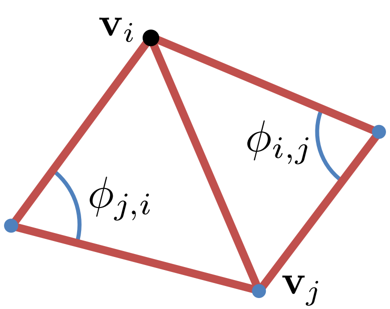
Choice of spring constants $k_{i,j}$

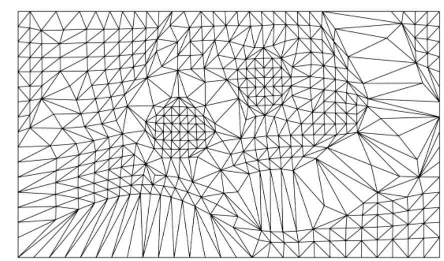
- Uniform $k_{i,j} = 1$





- Cotan $k_{i,j} = \cot \phi_{i,j} + \cot \phi_{j,i}$



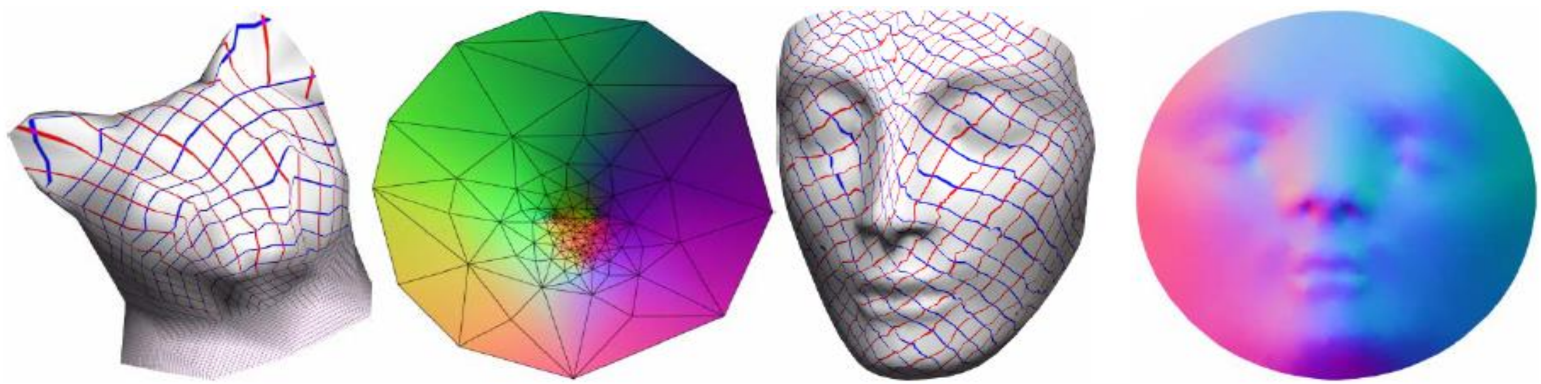


Tutte’s Theorem

- If the weights are **nonnegative**, and the boundary is fixed to a **convex** polygon, the parameterization is **bijective**
- (Tutte’63 proved for uniform weights, Floater’97 extended to arbitrary nonnegative weights)
- W.T. Tutte. “How to draw a graph”. Proceedings of the London Mathematical Society, 13(3):743-768, 1963.

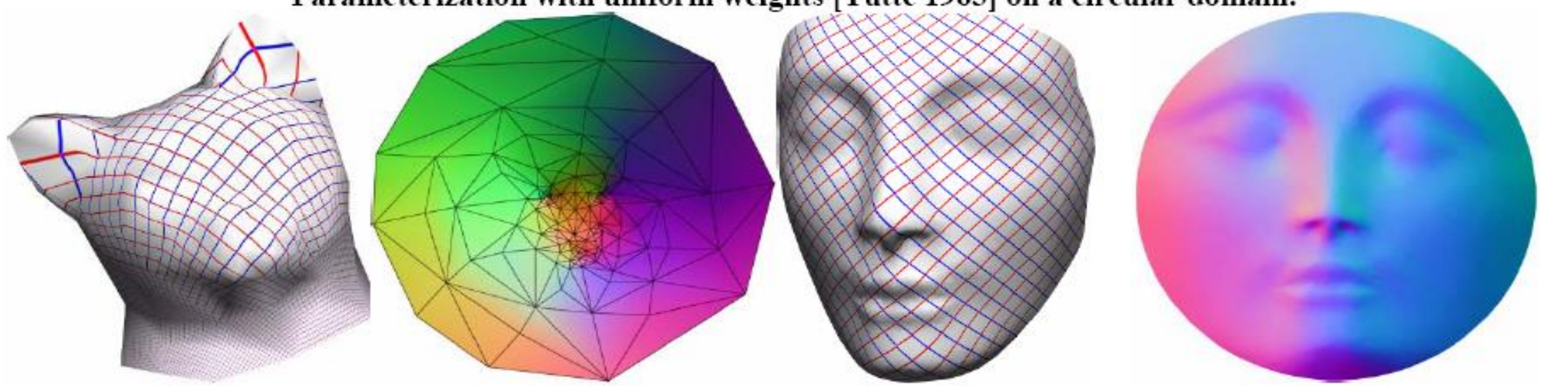
Comparison of Weights

uniform weights



Parameterization with uniform weights [Tutte 1963] on a circular domain.

cotan weights



Parameterization with harmonic weights [Eck et al. 1995] on a circular domain.

Eck et al. 1995, “Multiresolution analysis of arbitrary meshes”, SIGGRAPH 1995

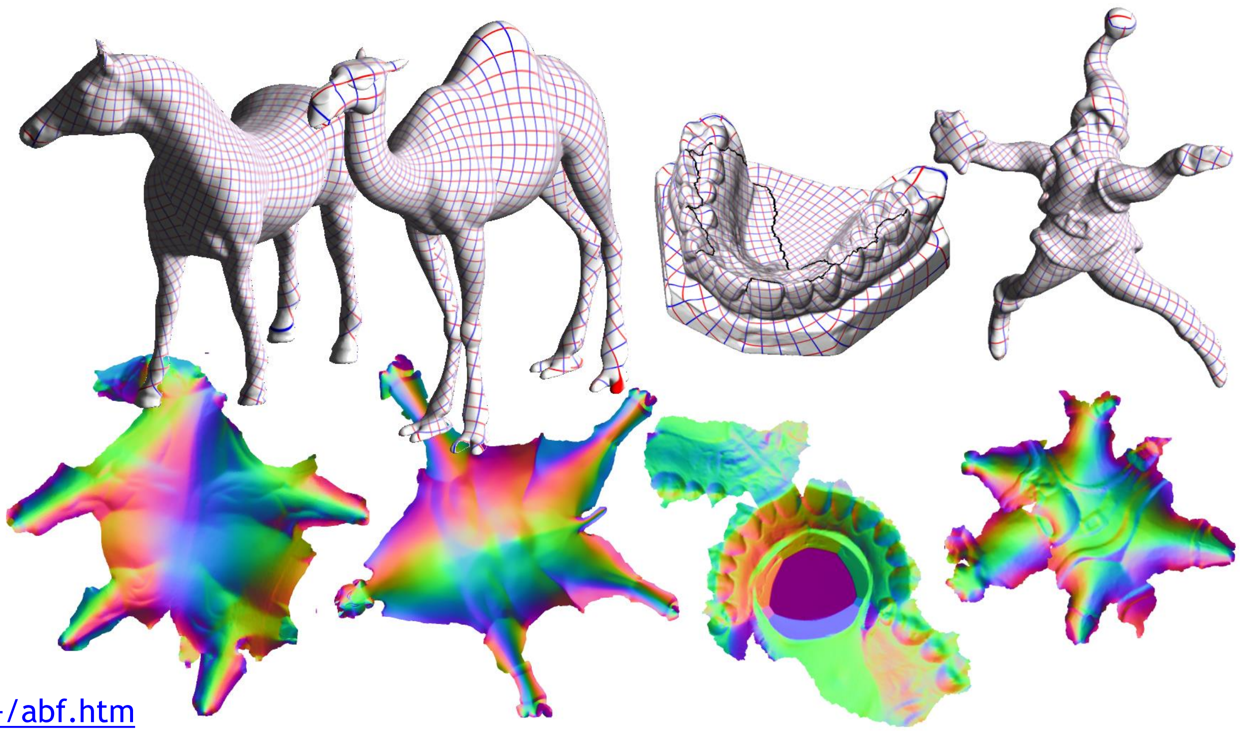
Discussion

- The results of cotan-weights mapping are better than those of uniform convex mapping (local area and angles preservation).
- But: the mapping is not always legal (the cotan weights can be negative for badly-shaped triangles…)

- In any case: sparse system to solve, so robust and efficient numerical solvers exist

Discussion

- Both mappings have the problem of fixed boundary – it constrains the minimization and causes distortion.
- More advanced methods do not require boundary conditions.



ABF++ method,
Sheffer et al. 2005
http://www.cs.ubc.ca/~sheffa/ABF++/abf.htm

Thank You