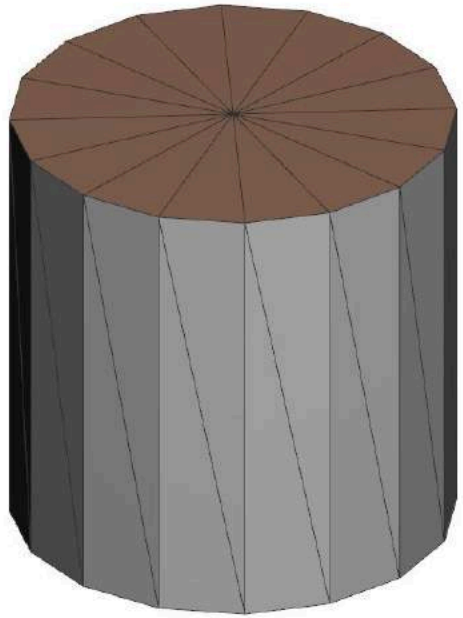
Surface Parameterization

History, etc

Introduction

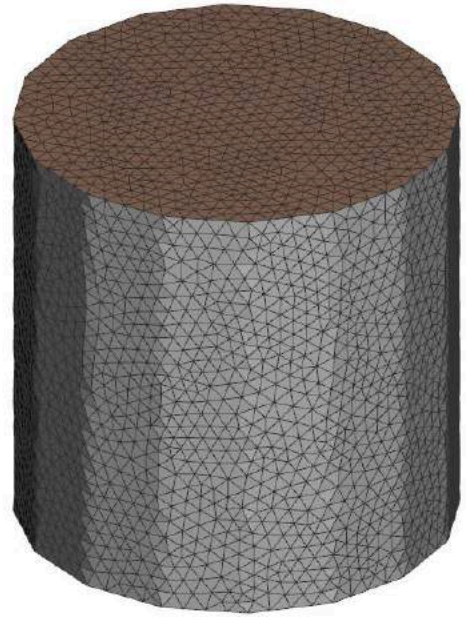
- Hi, I am Ryan – http://www.dgp.toronto.edu/~rms / @rms80
- Head of Design & Fabrication Research at Autodesk until last October
- Developed Autodesk Meshmixer
  - Lots of mesh processing tools, based on lots of GeomProc research
- Now starting a “geometry processing startup” – gradientspace.com

- Open-source mesh-processing library geometry3Sharp (C#)
  - https://github.com/gradientspace/geometry3Sharp
  - Remeshing!

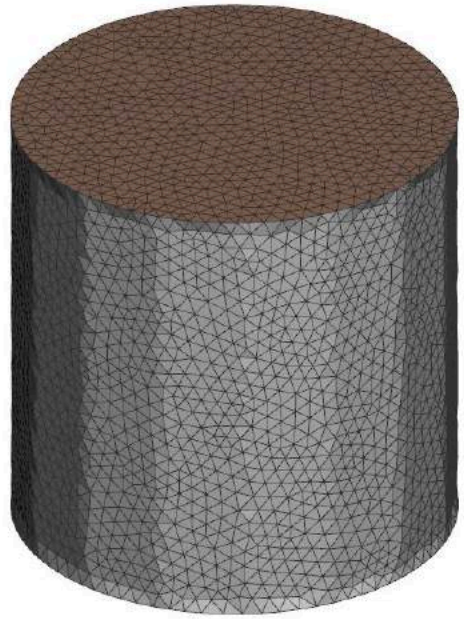


Original Mesh

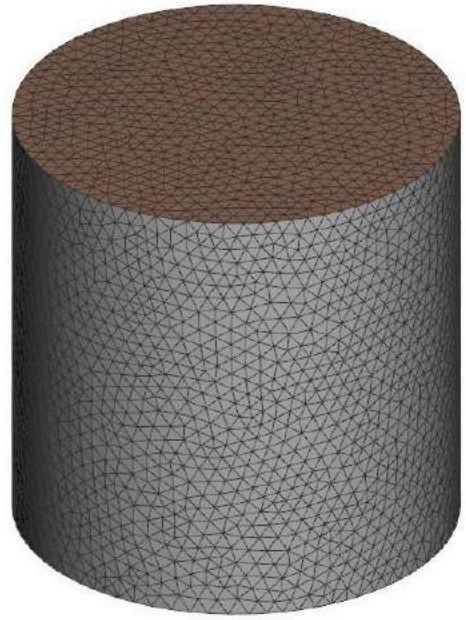
Remesh w/ Back-Projection



Sharp Edge Constraints



Analytic Circle Loop Constraints

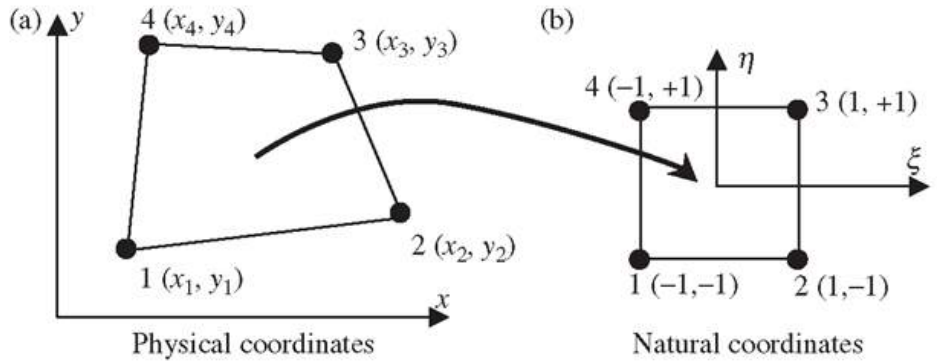


Projection onto Analytic Cylinder

Parameteri(z|s)ation

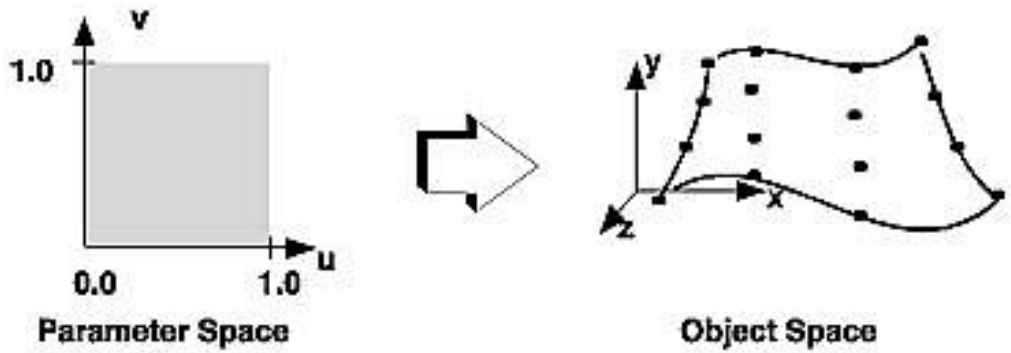
- Assigning a M-dimensional coordinate system to an N dimensional space, where M <= N

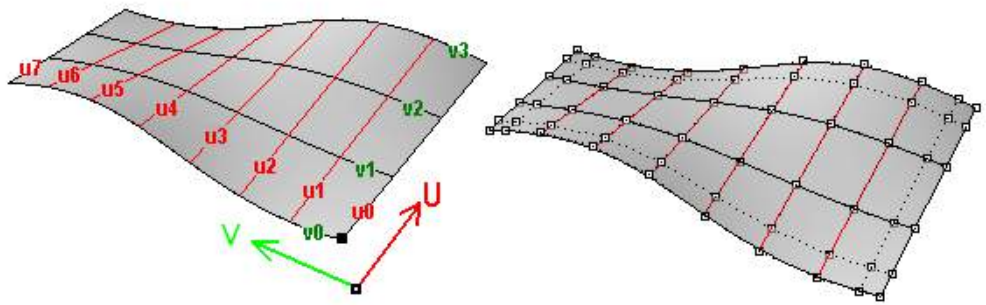
- Examples:
  - 3D space curve to real line
  - Hexahedral Element to Unit Box
  - Arbitrary genus-N 3D shape to N-dimensional Torus



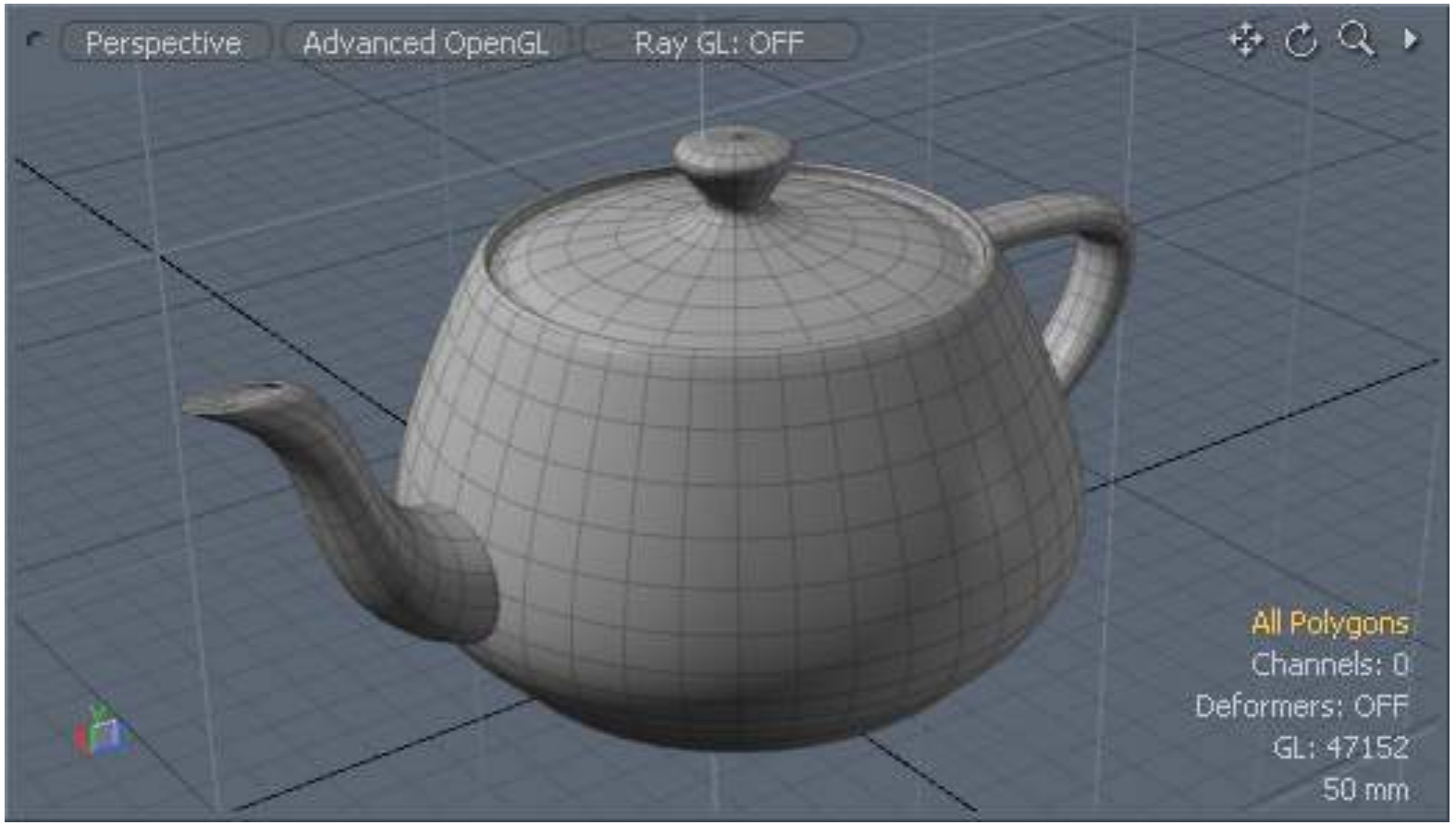
Surface Parameterization

- N == 3, M == 2
- NURBS Patch is canonical example
- **Parameter Domain** is a 2D rectangle
- **Embedding** is a smooth 3D surface

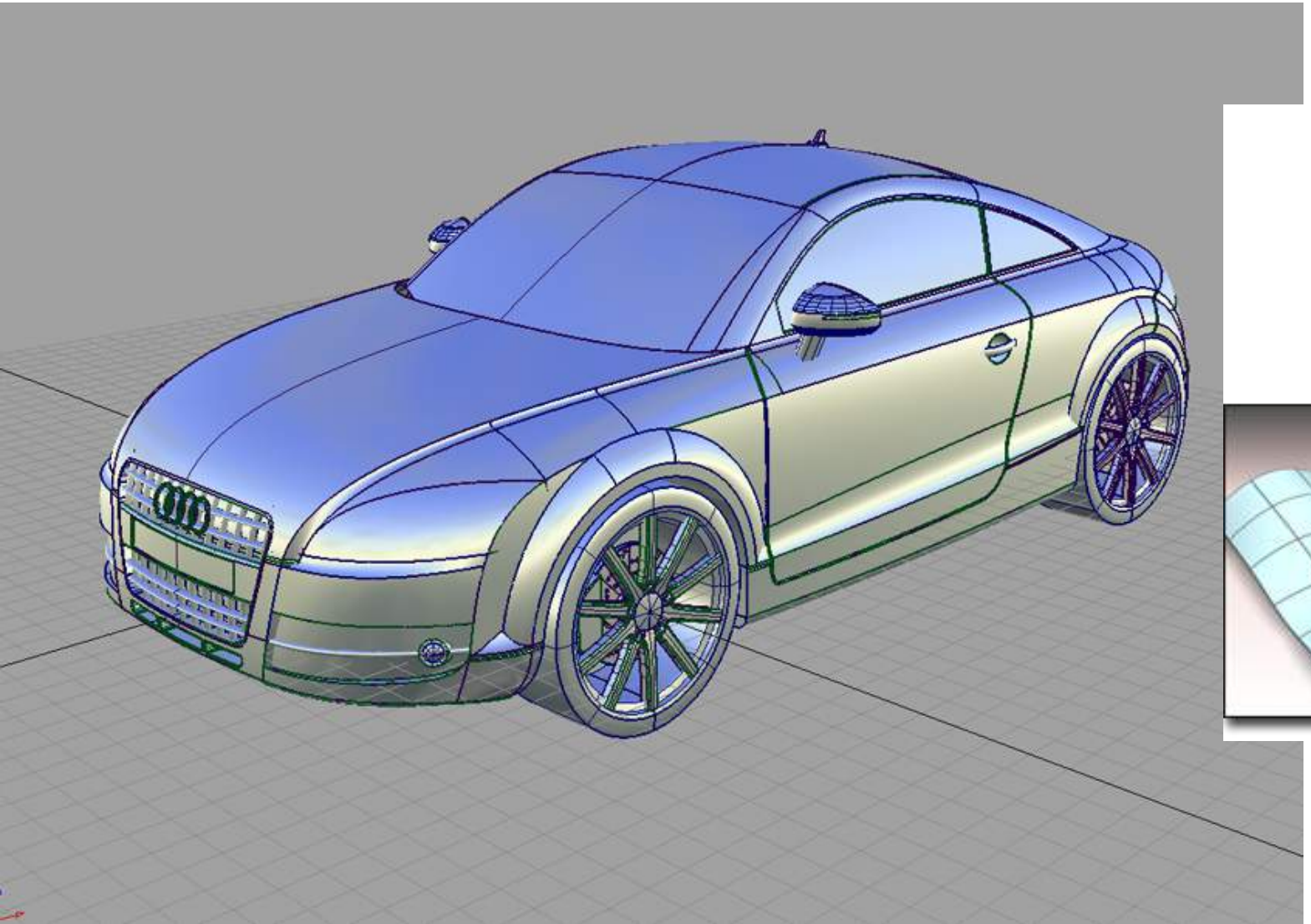


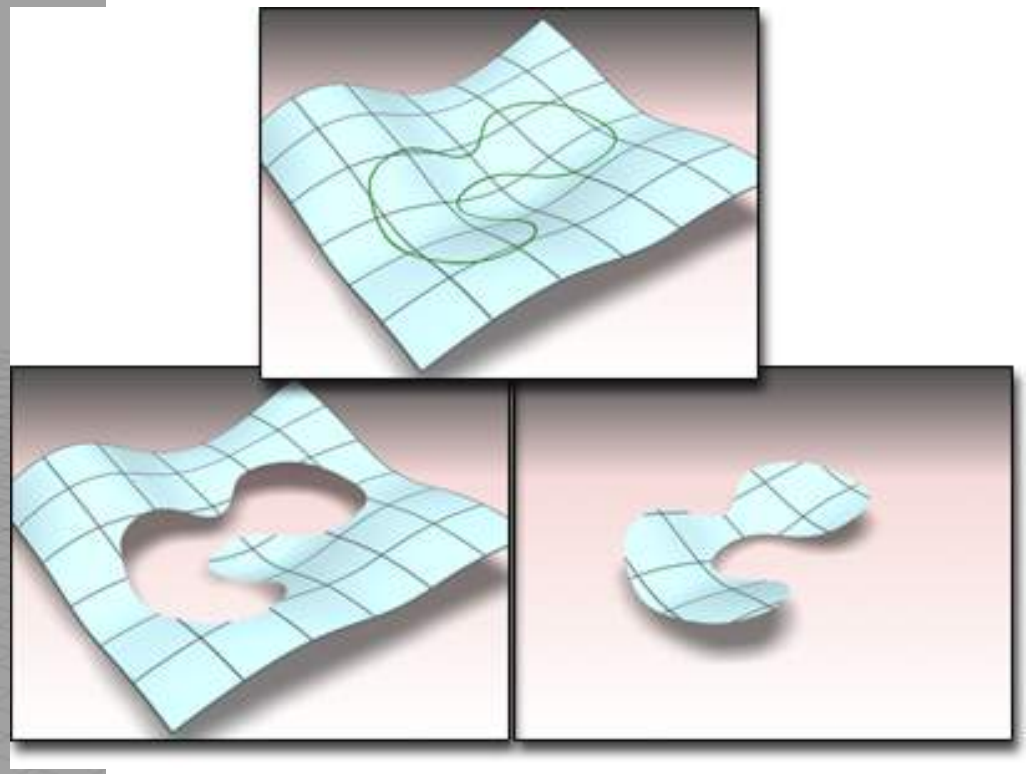


Parameter Domains are Useful



Parameter Domains are Useful

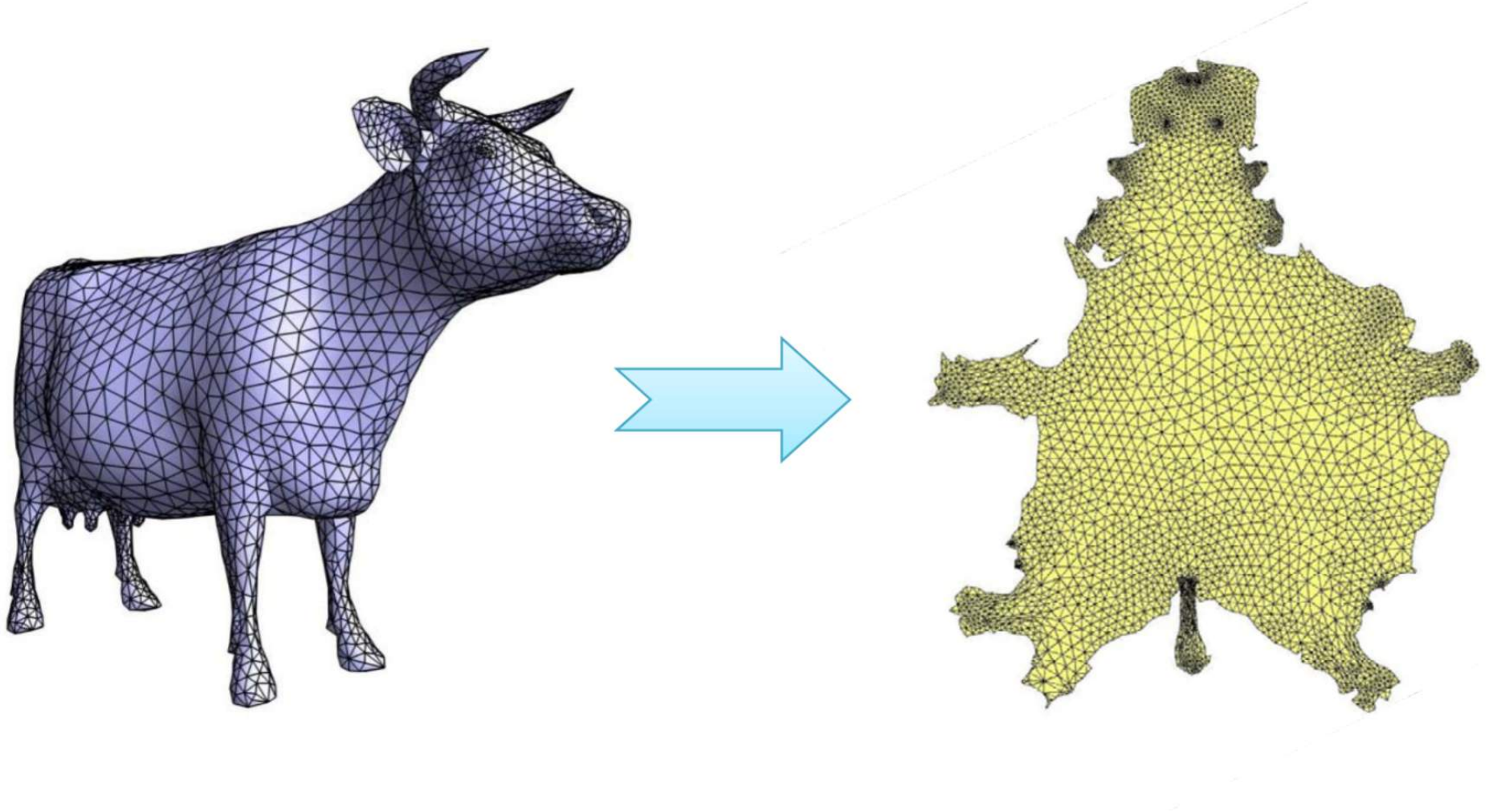




A small note on terminology

- “Parameter” is a pretty generic word
- When someone who is not a computer graphics person says “Parameterization”, even in a 3D context, they almost never mean Surface Parameterization like we will discuss today
- For example, if you hear a CAD person talking about “Parameterizing a Mesh”, or “Re-Parameterization”, they are talking about NURBS stuff

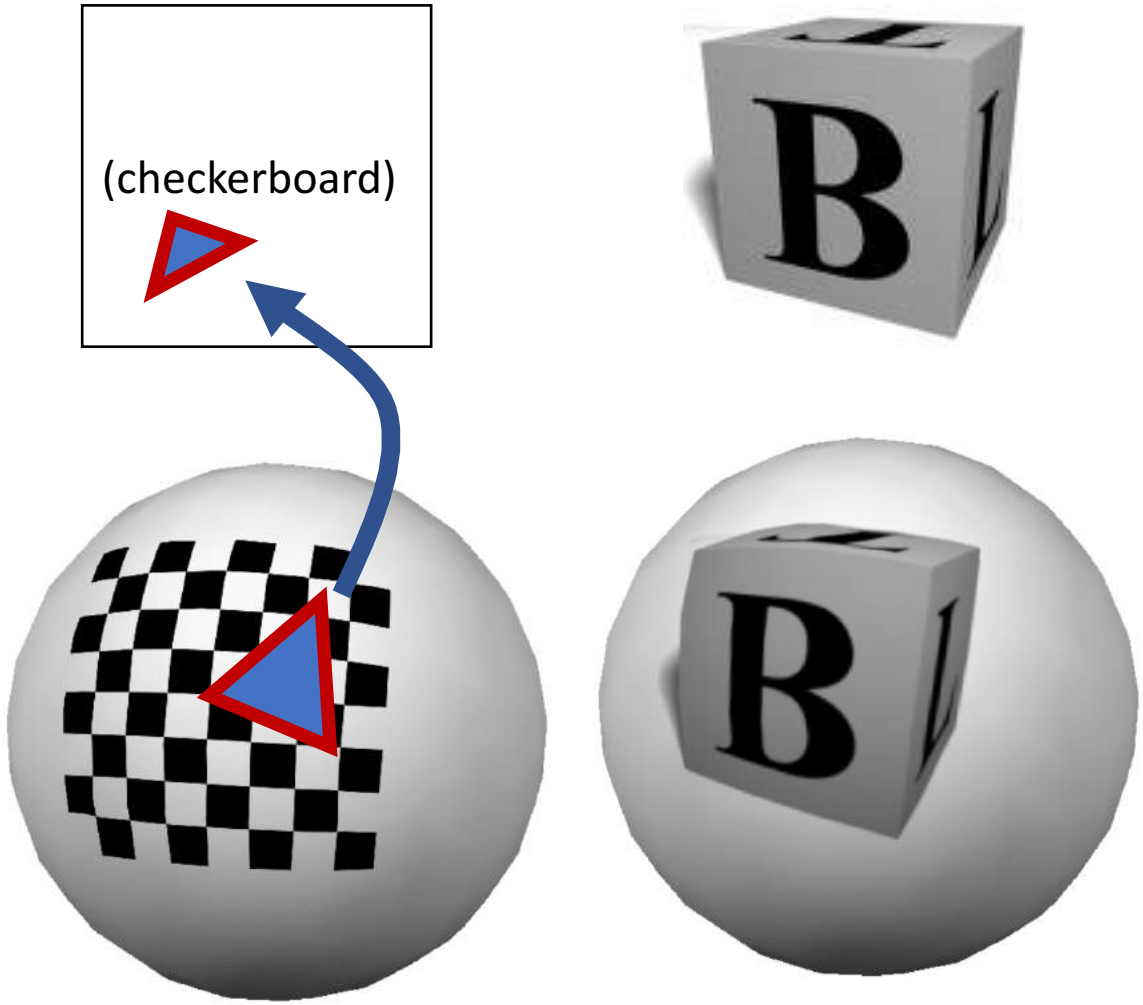
Mesh Parameterization



Texture Mapping

- Have square in image space
- Have 2D coordinates (u,v) for each 3D triangle *(“uv” coordinates)*

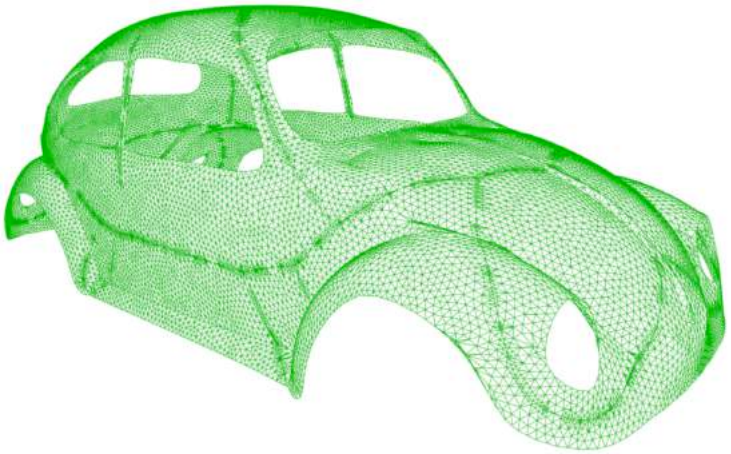
- For a point P in triangle:
  - Compute barycentric coordinates of P in 3D triangle
  - Reconstruct P in 2D triangle
  - Sample image

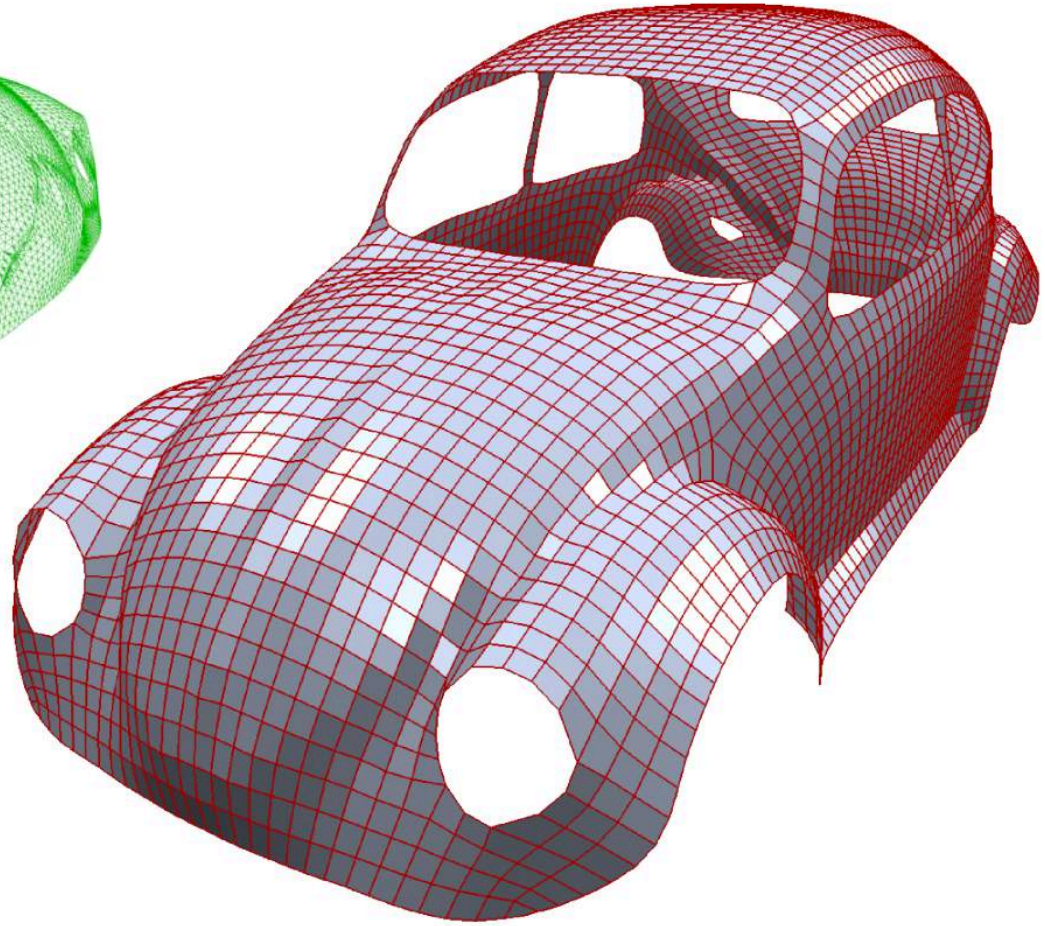


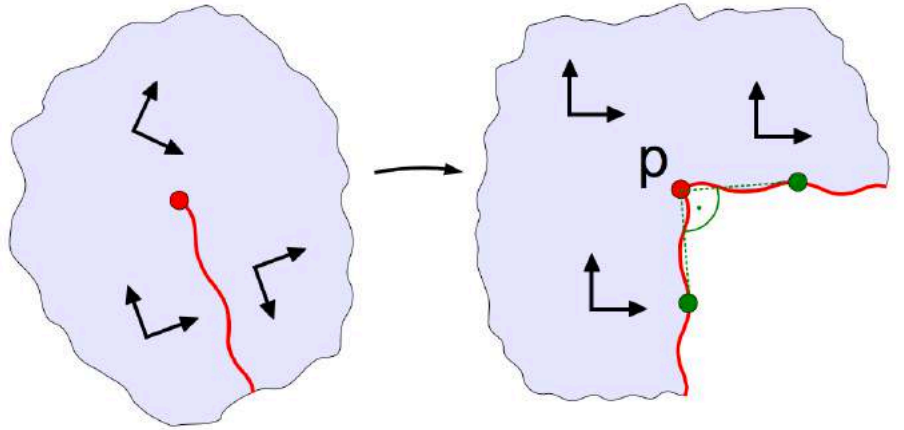
What else can you do with a parameterization?

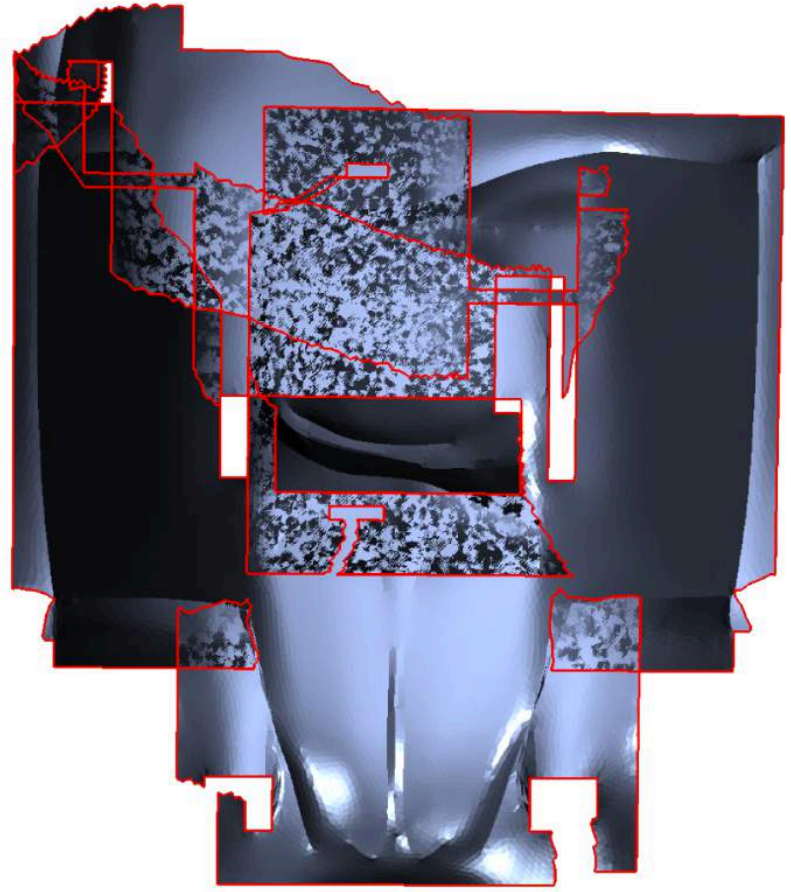
- Remeshing
- Mesh Editing
- Geometric Computation
- Shape Analysis
- …
- …
- …
- (Solve nearly any geometry problem)

Example: Quad Meshing



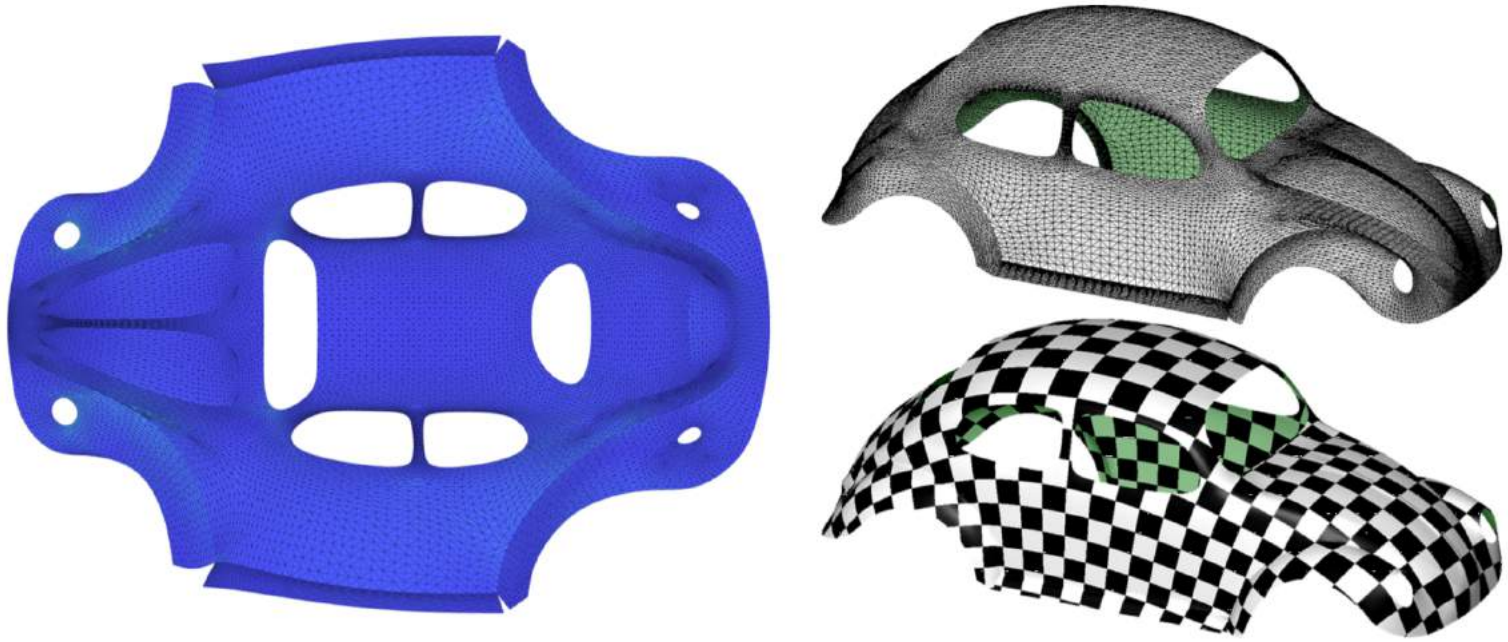




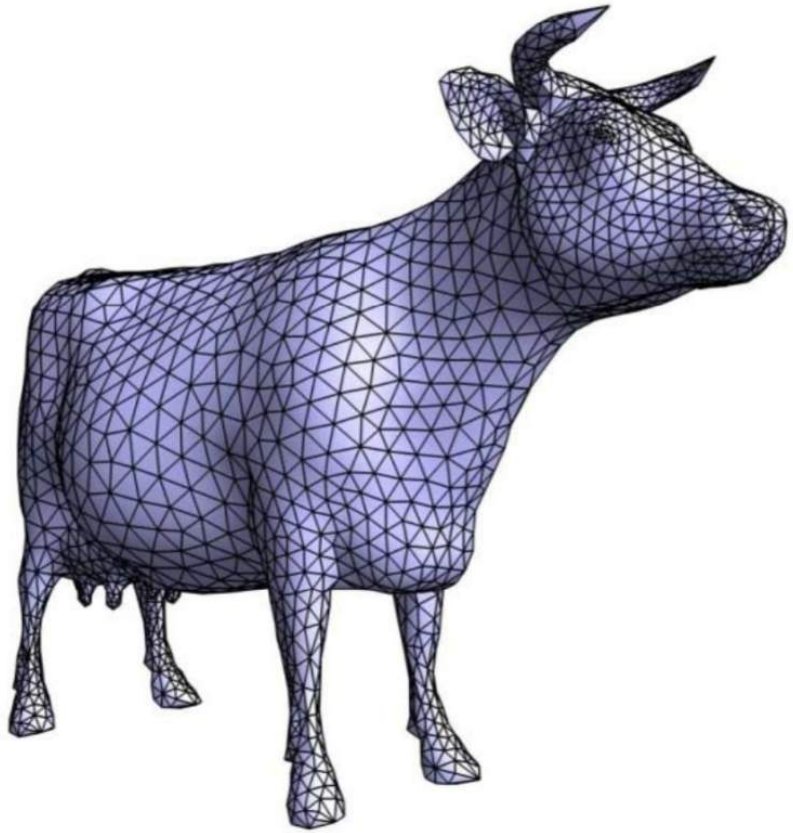


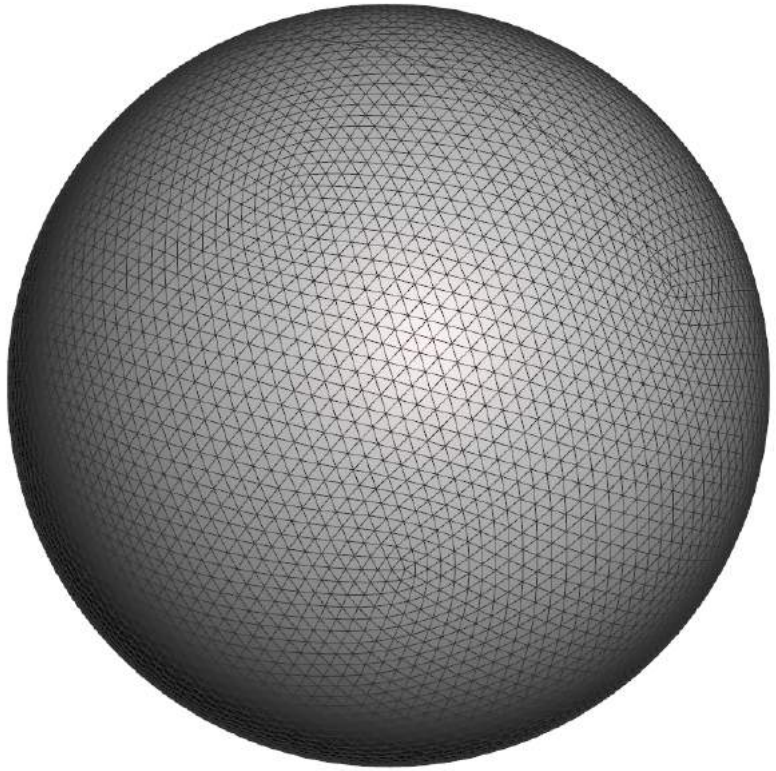
The Main Problem in Parameterization

- You have a triangle mesh homeomorphic to a disc, possibly with holes
- You want to flatten this mesh into 2D with some specific properties
  - No foldovers, minimize distortion



The hard problem we will ignore for now

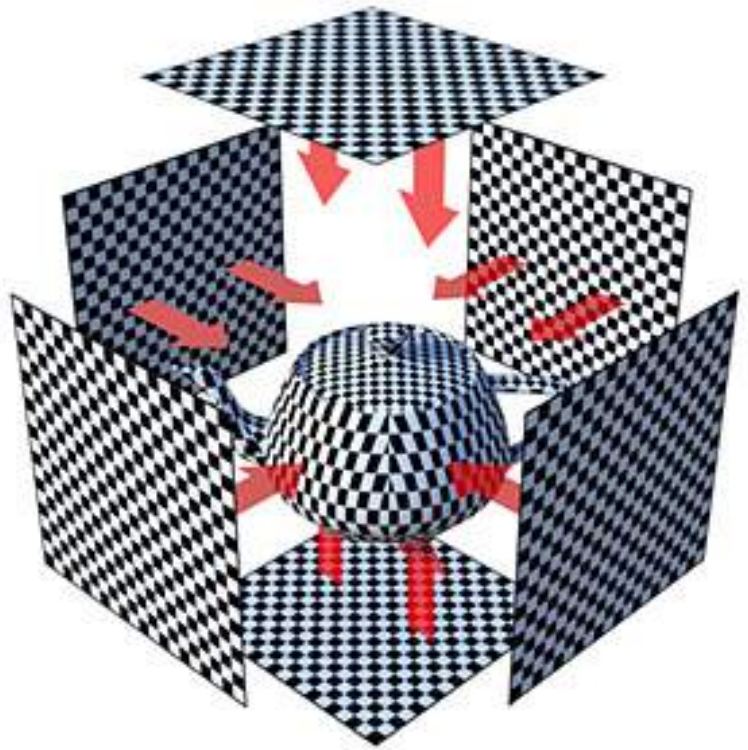




Not a Topological Disc

Projections

- Planar
- Cylindrical
- Spherical



Robot_004.3b - 3D-COAT 4.5.BETA15(CUDA)(DX64)

File Edit View Textures Symmetry Layers Freeze Hide Windows Scripts Help Always Connective Picki

Steady Stroke 15.000 Invert Tool Radius 18.447 Falloff 52% Depth 66% Smoothing 100% Opacity 103% Gloss Intensity 100% Metalness 0%

Texture Editor

? Reset Wireframe Color lambert5SG

Paint Tweak Retopo UV Sculpt Render [Camera]

Front

Alphas Brush Options Strips Color Palette

Stencils PBR Materials Presets

Layers Layer Blending Paint Objects Surface Materials VoxTree

Opacity 100% Standard Blend

Depth Opacity 100% Add depth

Gloss Opacity 100% Metal Opacity 100%

Camo
Layer 13
Layer 11
Normalmap(world-space)
AmbientOcclusion
CavityMask(Auto)
Layer 6
Layer 9
Layer 0

fps:201; Free: 96932 Mb,Pointers: 271 317; Face: 4790/4790

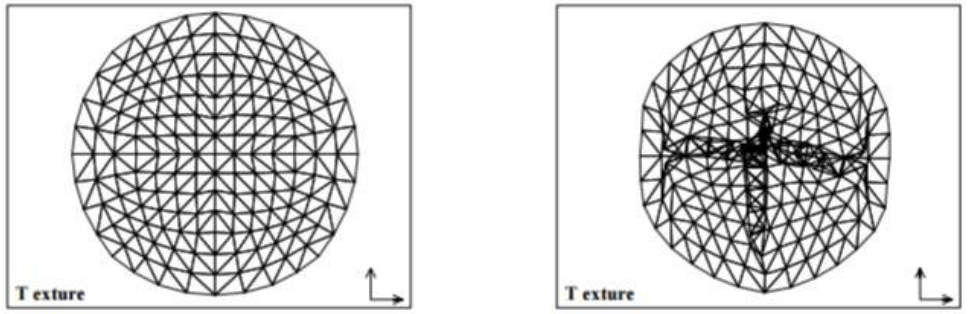
X=11.6660mm Y=128.103mm Z=31.5997mm

Mass-Spring-Type Approaches

- Define energies/forces based on lengths and angles
  - Turned out to be the “Green-Lagrange” deformation tensor
- Integrate!

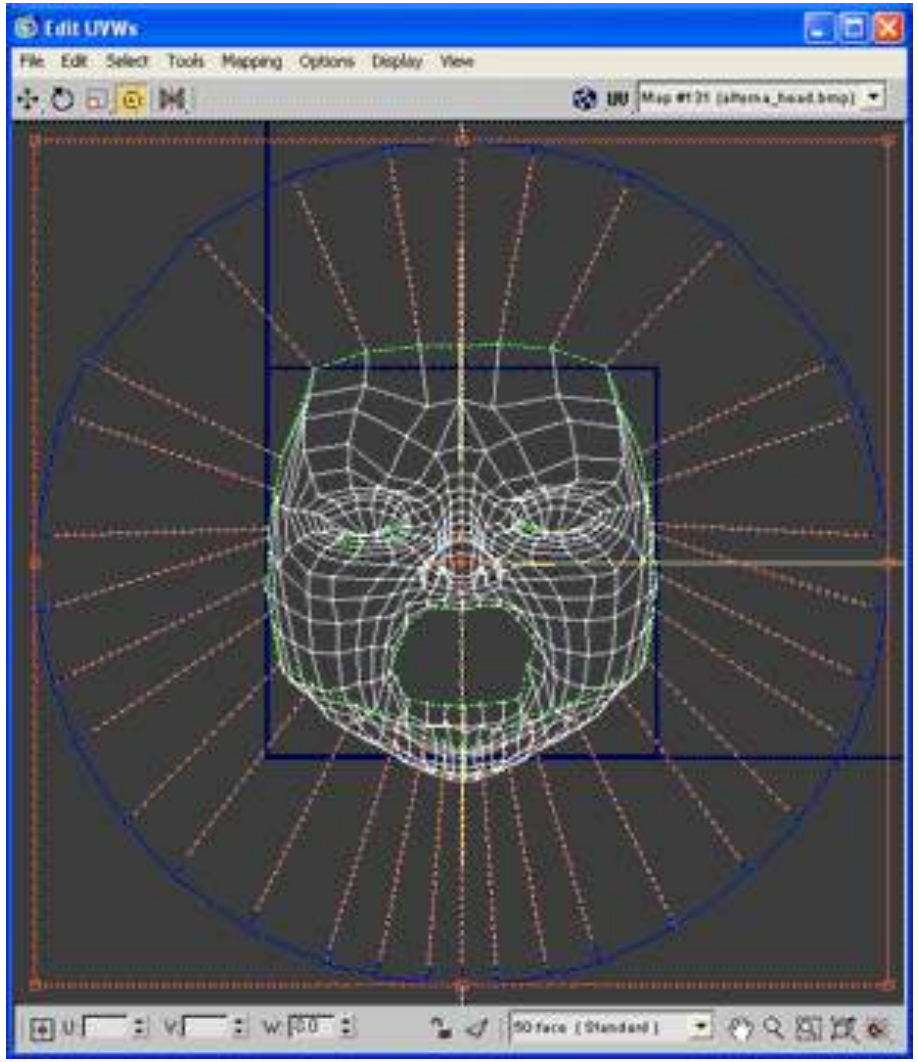
- Slow
- Not reliable
- …

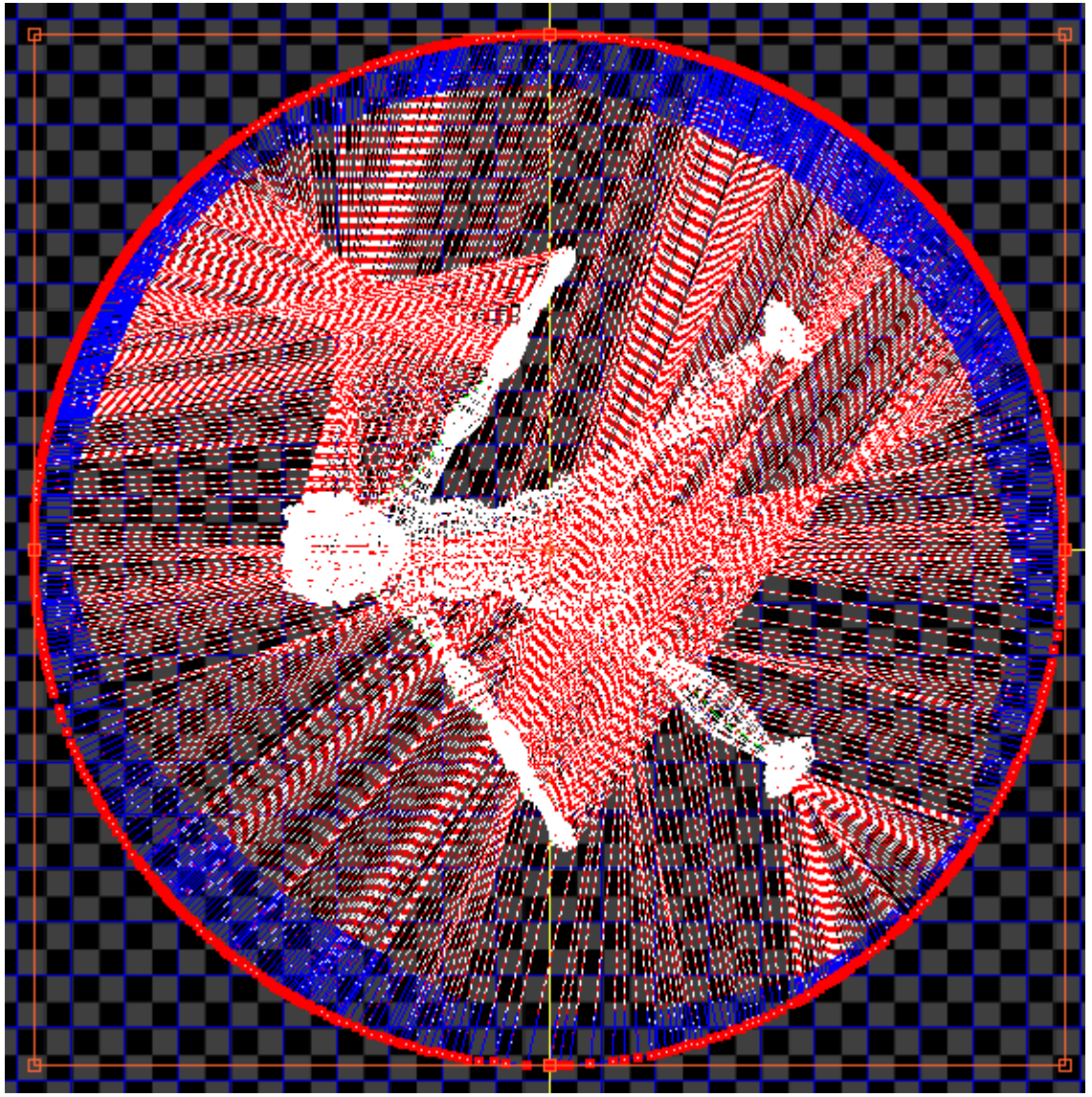
Fig. 3.a: Object 1 and object 2



Interactive Texture Mapping, Maillot et al SIGGRAPH 93

Pelt Mapping





First problem to solve: no foldovers

- “Bijection”
- Triangle meshes are Graphs
- There is a sub-topic of Graph Theory called “Graph Drawing”, which is much older than Graphics…

Tutte Embedding

HOW TO DRAW A GRAPH

By W. T. TUTTE

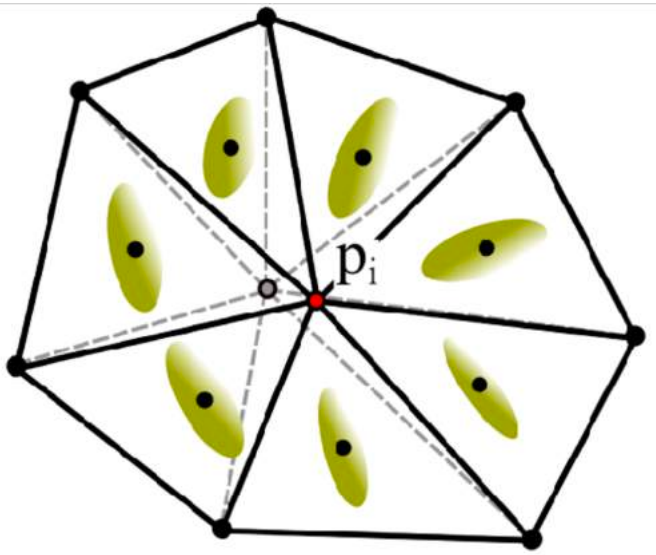
[Received 22 May 1962]

1. Introduction

WE use the definitions of (**11**). However, in deference to some recent attempts to unify the terminology of graph theory we replace the term 'circuit' by 'polygon', and 'degree' by 'valency'.

A graph G is 3-*connected* (*nodally* 3-*connected*) if it is simple and non-separable and satisfies the following condition; if G is the union of two proper subgraphs H and K such that $H \cap K$ consists solely of two vertices u and v, then one of H and K is a link-graph (arc-graph) with ends u and v.

Tutte Embedding



- You have a 3D network of nodes and edges
- Can you draw a 2D version of this network where no edges overlap?

- Define each vertex as convex combination of neighbours
  - Each vertex should lie at centroid of neighbours – “uniform” weights

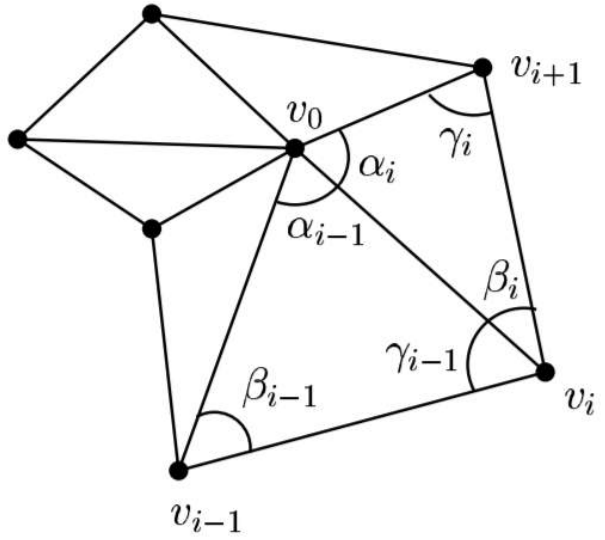
- Fix open boundary to a convex shape (circle, square, polygon)
  - Must be convex or interior vertices can “escape” the boundary in 2D

- Solve system of linear equations

The formula

$$\sum_{i=1}^{k} \lambda_i v_i = v_0$$

Vertex v_0 is is a linear combination of its neighbours v_i
\lambda_i are the scalar “weights”



Weights are constructed locally!

The formula

$$\sum_{i=1}^{k} \lambda_i v_i = v_0$$

Vertex v_0 is is a linear combination of its neighbours v_i
\lambda_i are the scalar “weights”

$$\sum_{i=1}^{k} \lambda_i = 1$$

The weights sum to 1

The formula

$$\sum_{i=1}^{k} \lambda_i v_i = v_0$$

Vertex v_0 is is a linear combination of its neighbours v_i
\lambda_i are the scalar “weights”

$$\sum_{i=1}^{k} \lambda_i = 1$$

The weights sum to 1

…If not: $\lambda_i = \frac{w_i}{\sum_{j=1}^{k} w_j}$

The formula

$$\sum_{i=1}^{k} \lambda_i v_i = v_0$$

Vertex v_0 is is a linear combination of its neighbours v_i
\lambda_i are the scalar “weights”

$$\sum_{i=1}^{k} \lambda_i = 1$$

The weights sum to 1

$$\lambda_1, \ldots, \lambda_k \geq 0$$

The weights are positive (often this rule is “bent”)

(If weights are not positive, v0 can escape from one-ring)

The formula

Sparse Linear System!

$$\sum_{i=1}^{k} \lambda_i v_i = v_0$$

Vertex v_0 is is a linear combination of its neighbours v_i
\lambda_i are the scalar “weights”

$$\sum_{i=1}^{k} \lambda_i = 1$$

The weights sum to 1

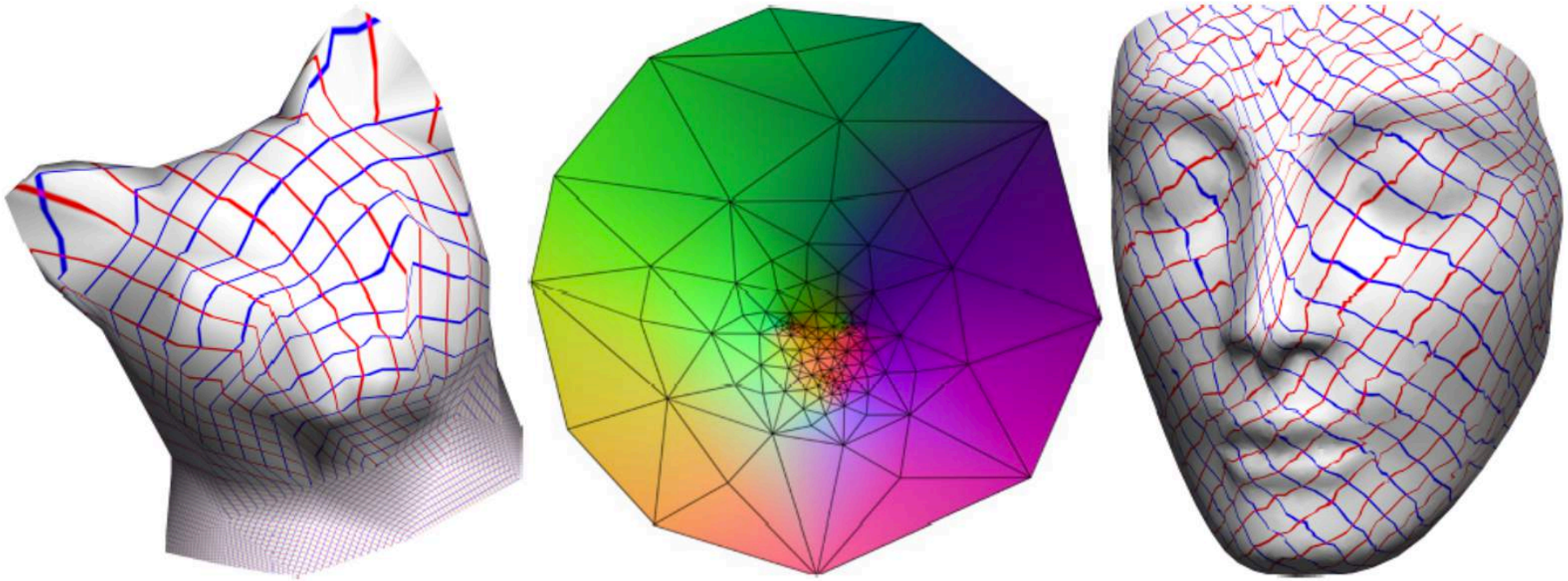
$$\lambda_1, \ldots, \lambda_k \geq 0$$

The weights are positive (often this rule is “bent”)

(If weights are not positive, v0 can escape from one-ring)

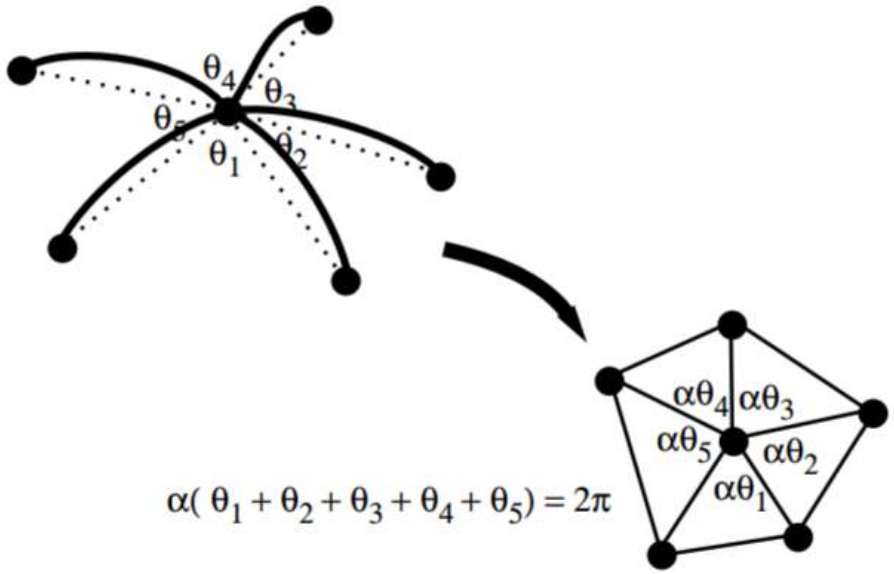
Ok, Bijection. But…

- Uniform weights completely discard triangle shape information
- Even in 2D mesh, most vertices not at centroids of neighbours…

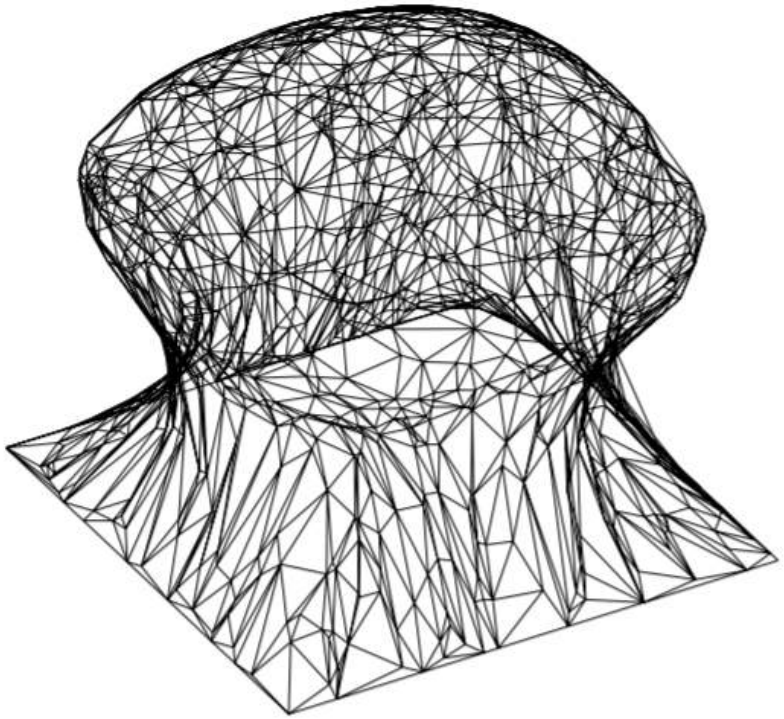


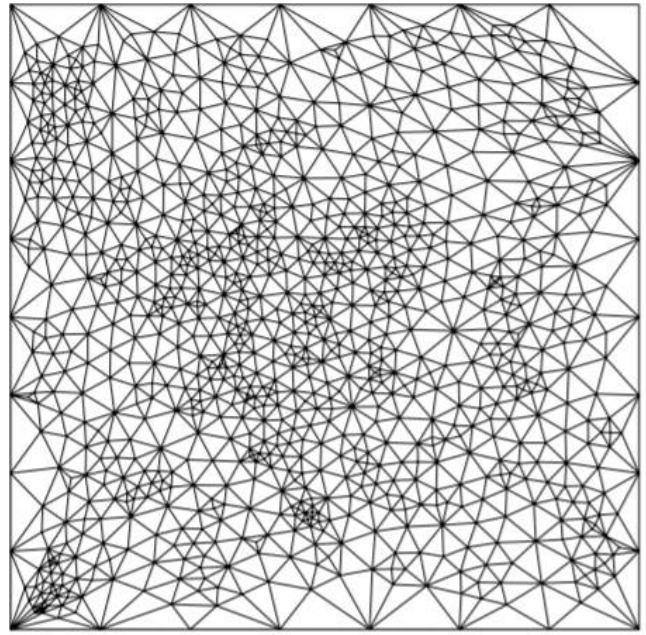
Shape-Preserving Weights

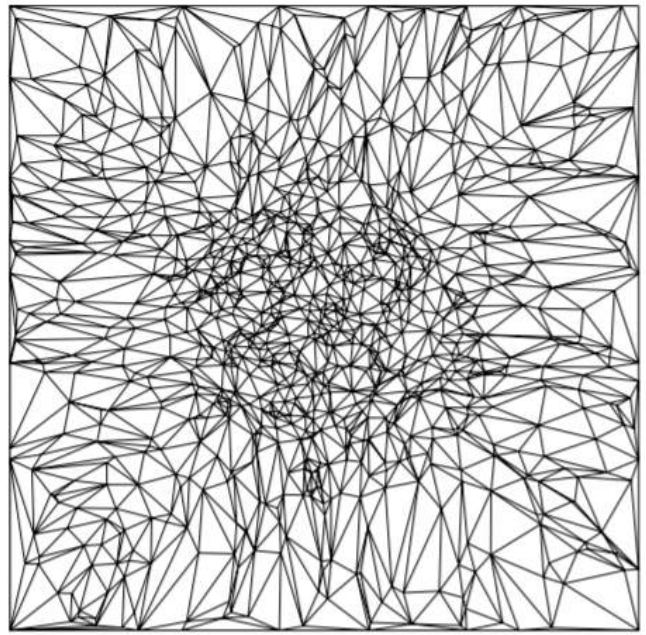
- Construct weights that would be correct for planar meshes
- Locally flatten each one-ring
  - “geodesic polar map”
- In this local 2D space, geometrically construct weights that reproduce P exactly (see paper)

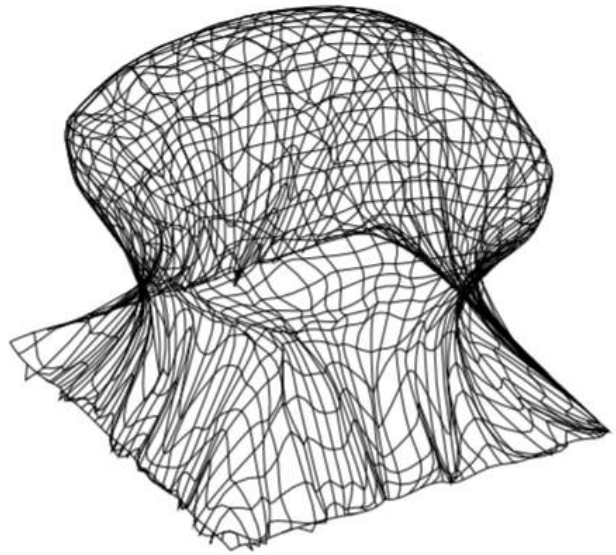


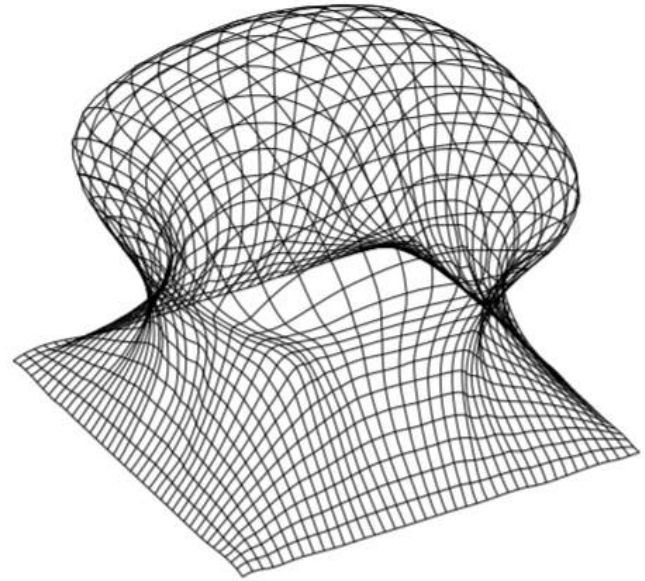
Parametrization and smooth approximation of surface triangulations, Floater 97









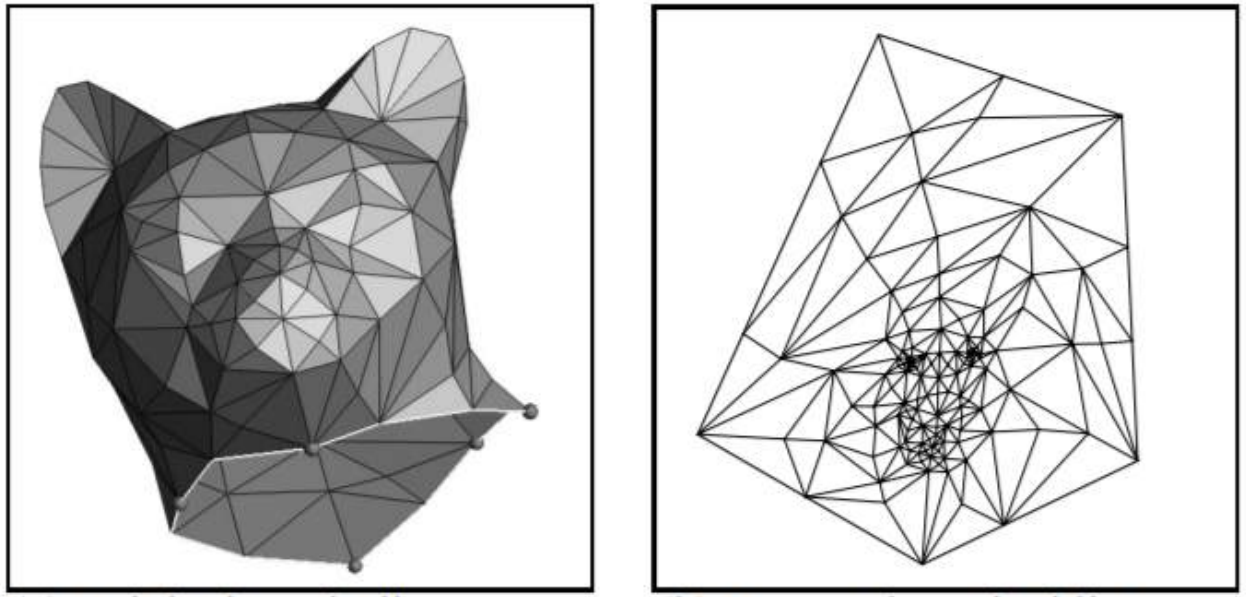


Harmonic Maps

- Weights based on edge lengths and areas
  - Can be negative, so parameterization can have fold-overs

- Map open boundary to Convex N-gon

- Solve sparse linear system



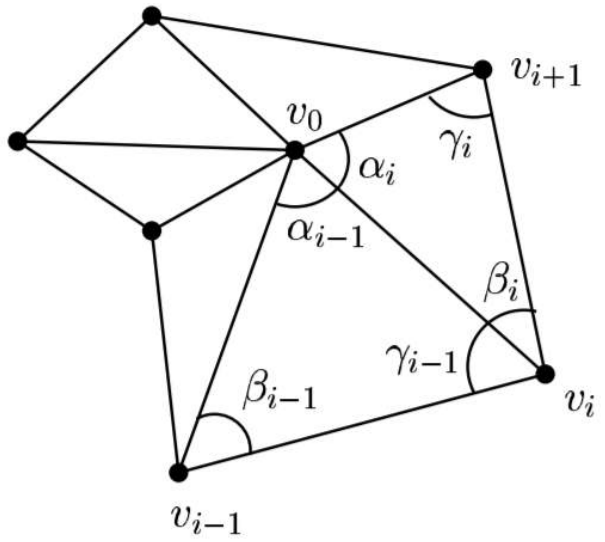
(a) Original mesh tile (b) Harmonic embedding

Multiresolution Analysis of Arbitrary Meshes, Eck et al, SIGGRAPH 95

Mean Value Weights

- Positive by construction

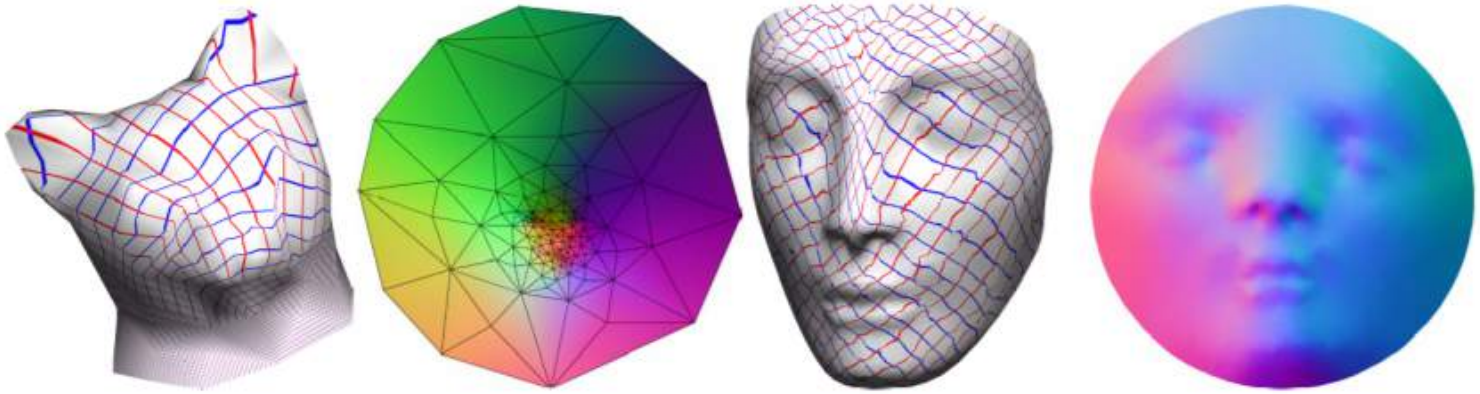
$$w_i = \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{||v_i - v_0||}$$



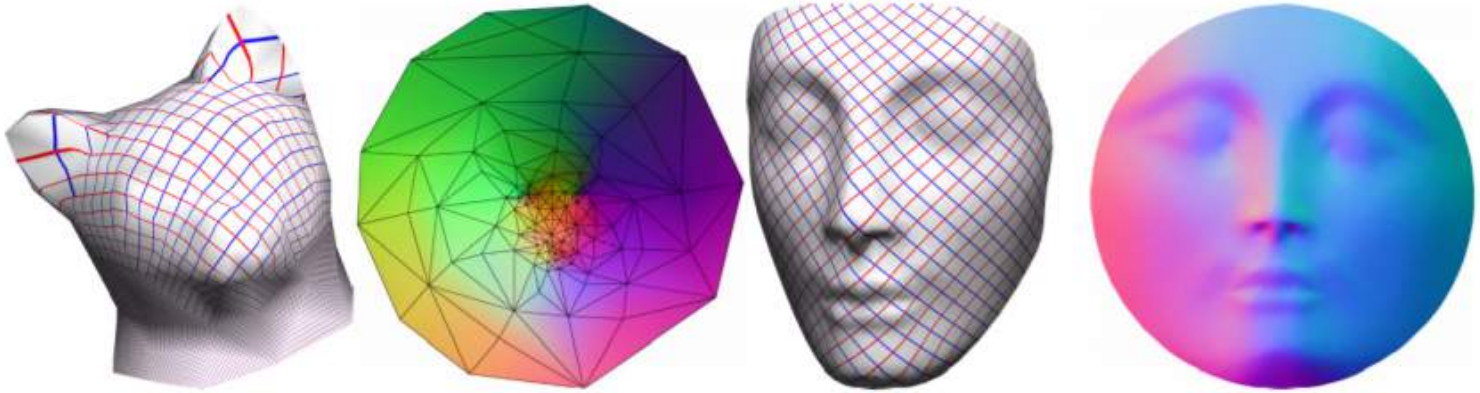
- "Reproducing" / "Linear Precision" in planar mesh case
- Have been extended to general N-gons, 3D meshes, …

tan(theta) >= 0 in [0,90)

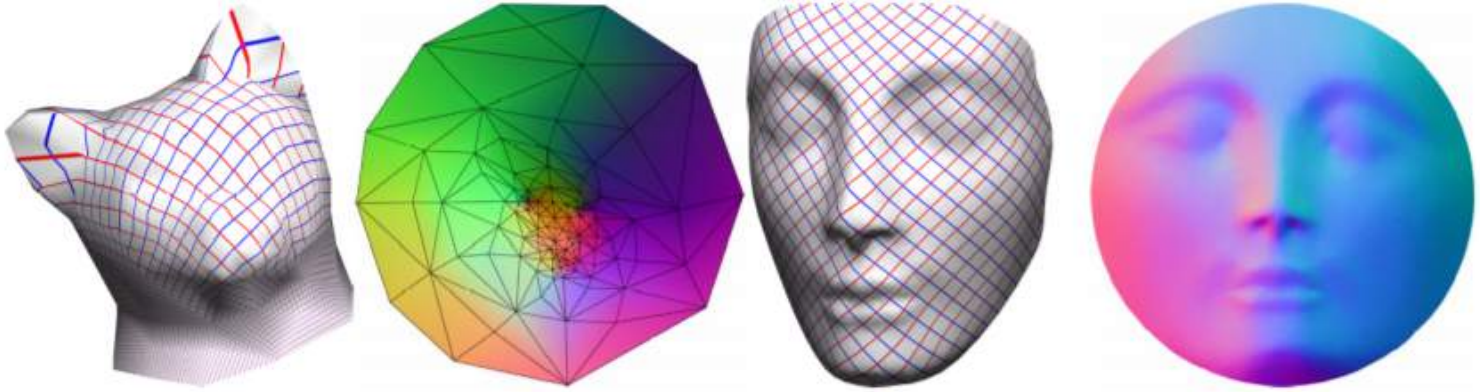
Mean Value Coordinates, Floater, 2000



Parameterization with uniform weights [128] on a circular domain.



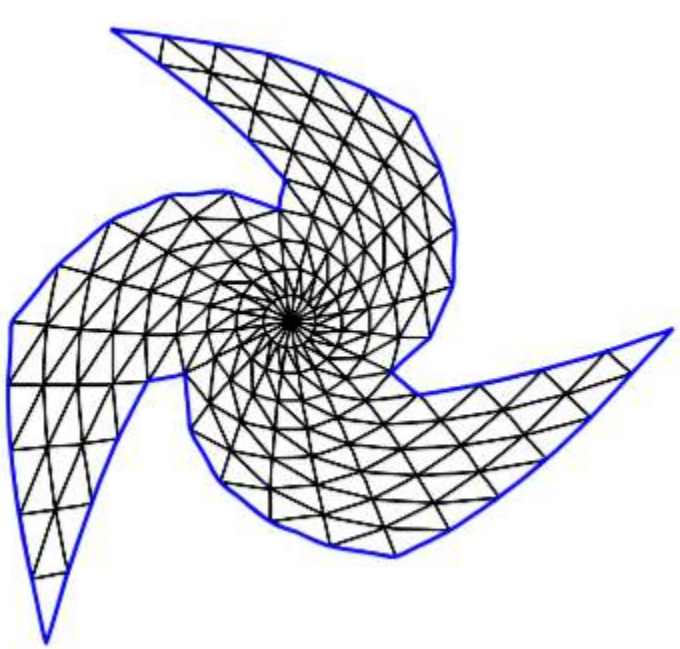
Parameterization with harmonic weights [28] on a circular domain.

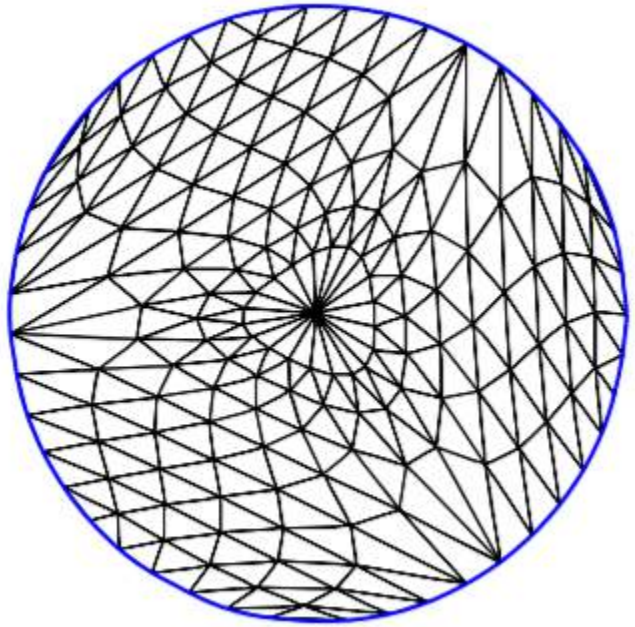


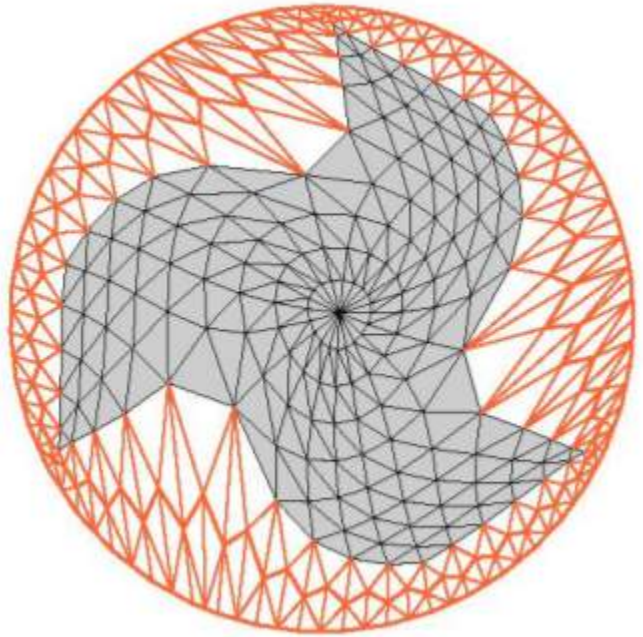
Parameterization with mean value weights [33] on a circular domain.

Not a lot of flexibility once boundary is fixed…

“Virtual Boundary”







(hack)

MIPS

- Re-cast existing work as energy minimization

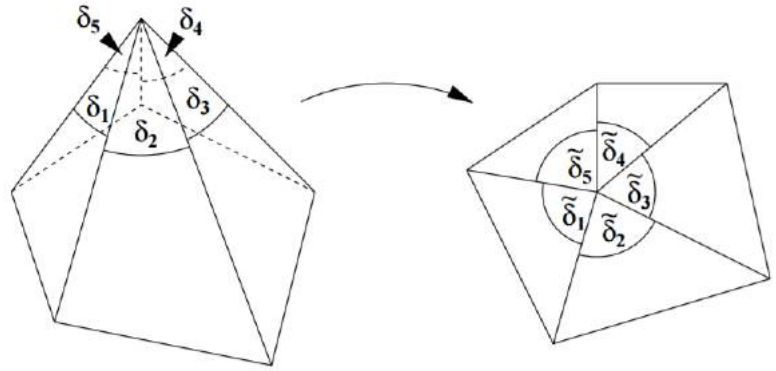
1) find a parametrization for the *boundary points*, and
2) minimize an edge-based energy function

$$E = \frac{1}{2} \sum_{\{i,j\} \in \text{Edges}} c_{ij} \, \|p_i - p_j\|^2$$

to determine the parametrization for the *inner points*.

MIPS: An Efficient Global Parameterization Method, Hormann and Greiner 2000

MIPS



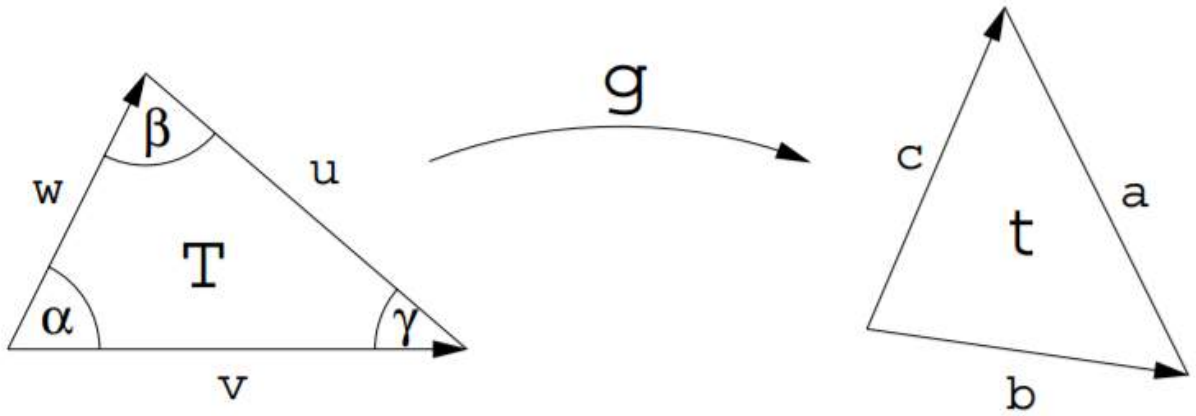
- Asked “What is the best-case scenario?”

- **Isometric** mapping – lengths and angles preserved
- Not possible for a 3D shape – but how close can we get?

- **M**ost **I**sometric **P**arameterization**S**
  - Basically, minimize a function of per-triangle deformation
  - iterative nonlinear optimization

MIPS: An Efficient Global Parameterization Method, Hormann and Greiner 2000

Distortion of Linear Map

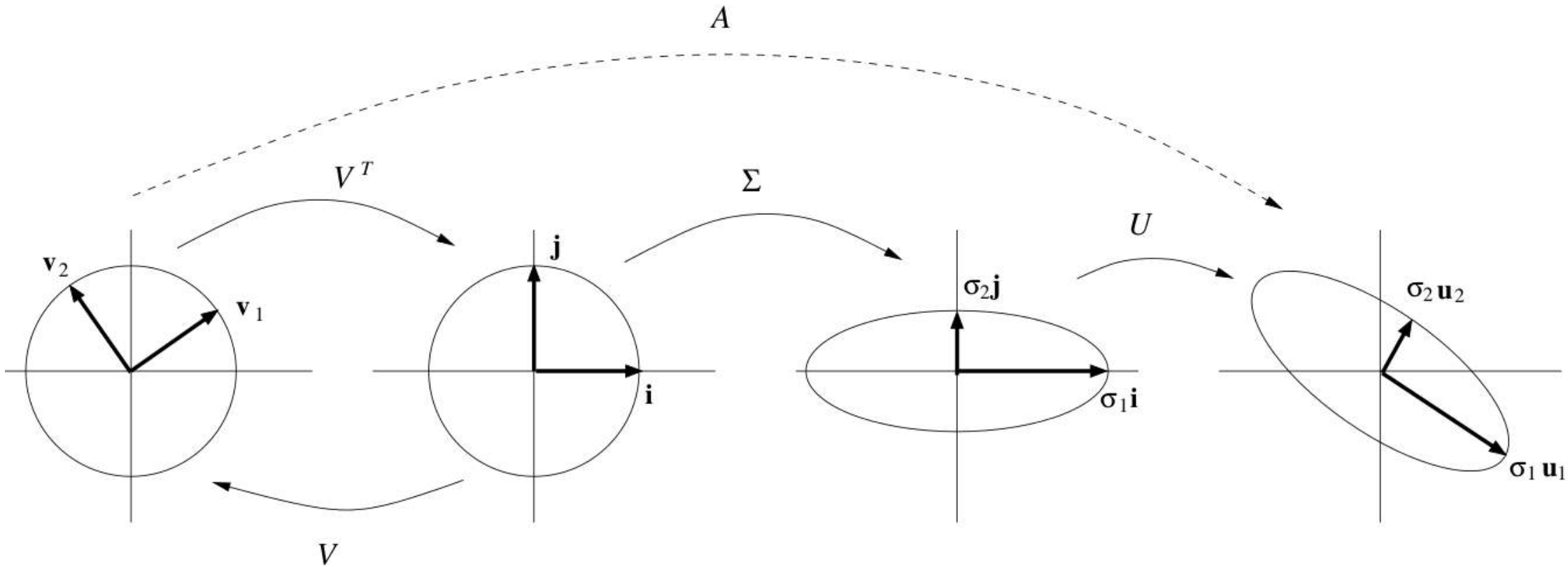


$$g(x) = Ax + b$$

$$U^t AV = \Sigma = \begin{pmatrix} \sigma_1 & \\ & \sigma_2 \end{pmatrix}$$

“Metric Distortion”

(Singular Value Decomposition)


A
V^T
Σ
U
$\mathbf{v}_2$
$\mathbf{v}_1$
j
i
$\sigma_2\mathbf{j}$
$\sigma_1\mathbf{i}$
$\sigma_2\,\mathbf{u}_2$
$\sigma_1\,\mathbf{u}_1$
V

MIPS

(Start with foldover-free convex-boundary embedding – eg Tutte)

repeat

choose a vertex p_i by random

let $t_{i_1}, \ldots, t_{i_n}$ be the triangles that surround p_i

fix the positions of all parameter values except for p_i

minimize the local deformation energy $\kappa_i = \sum_{k=1}^{n} \kappa_F(A_{i_k})$ in order to get the optimal position $\tilde{p}_i$ of p_i

until numerical convergence

$$\kappa_F(A) = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1 \sigma_2} = \frac{4E_D(g)}{\det \partial\psi} = \frac{\mathrm{trace}(A^t A)}{\det A}$$

Using Most Isometric Parametrizations for Remeshing Polygonal Surfaces, Labsik et al, 2000

MIPS

(Start with foldover-free convex-boundary embedding – eg Tutte)

repeat

choose a vertex p_i by random

let $t_{i_1}, \ldots, t_{i_n}$ be the triangles that surround p_i

fix the positions of all parameter values except for p_i

minimize the local deformation energy $\kappa_i = \sum_{k=1}^{n} \kappa_F(A_{i_k})$ in order to get the optimal position $\tilde{p}_i$ of p_i

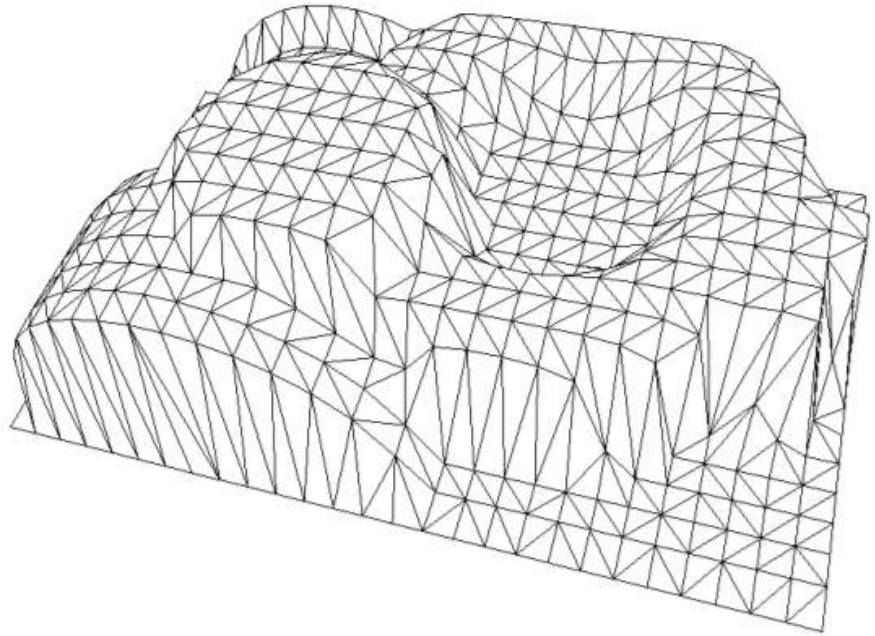
until numerical convergence

$$\kappa_F(A) = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1 \sigma_2}$$

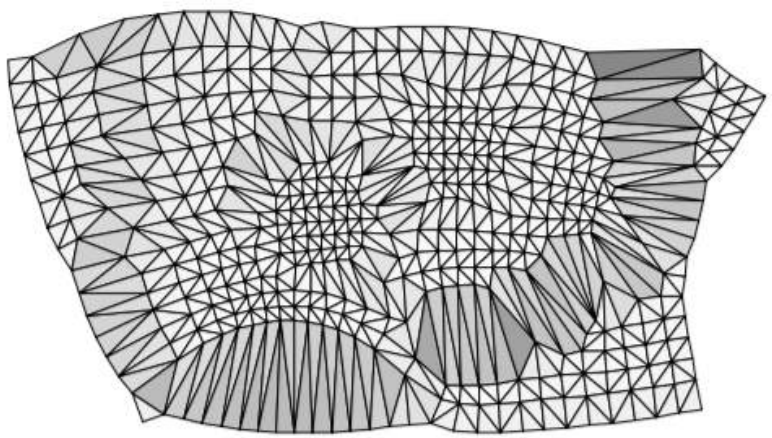
$$= \frac{4E_D(g)}{\det \partial\psi}$$

Dirichlet Energy

Using Most Isometric Parametrizations for Remeshing Polygonal Surfaces, Labsik et al, 2000



Most Isometric Parametrization



Discrete Harmonic Parametrization

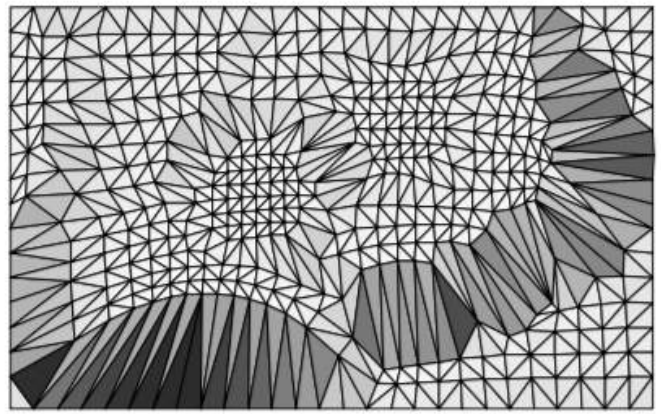


Fig. 4. Data set and gray-coded Dirichlet energy of different parametrizations.

MIPS: An Efficient Global Parameterization Method

???

repeat

choose a vertex p_i by random

let $t_{i_1}, \ldots, t_{i_n}$ be the triangles that surround p_i

fix the positions of all parameter values except for p_i

minimize the local deformation energy $\kappa_i = \sum_{k=1}^{n} \kappa_F(A_{i_k})$ in order to get the optimal position $\tilde{p}_i$ of p_i

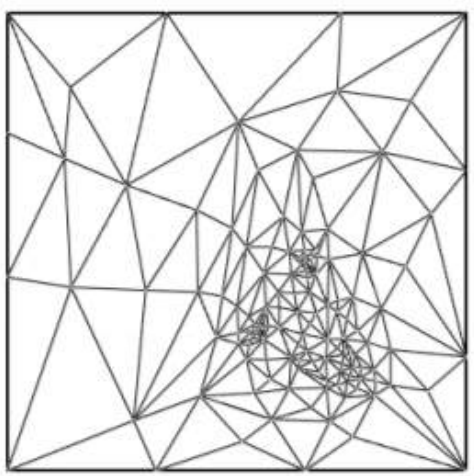
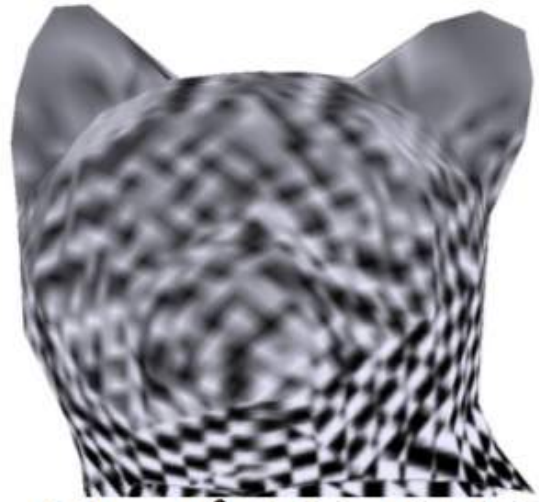
until numerical convergence

Stretch-Minimizing Parameterization

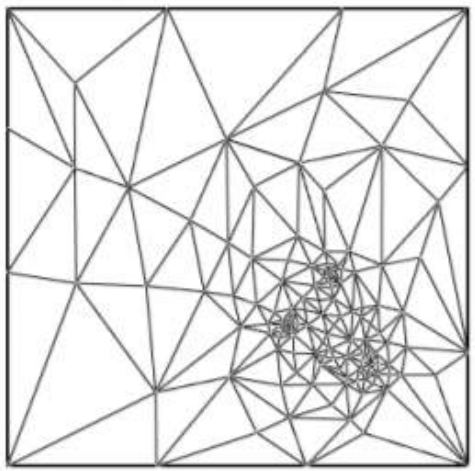
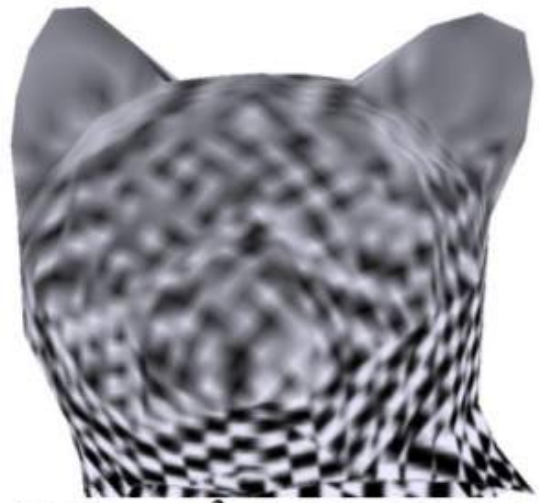
$$L_2(T) = \sqrt{(\sigma_1^2 + \sigma_2^2)/2} \qquad L_\infty(T) = \max(\sigma_1, \sigma_2)$$

- Iterative optimization starting from initial embedding
- Uses line searches – faster but requires fixed boundary (?)
- Not Smooth!!

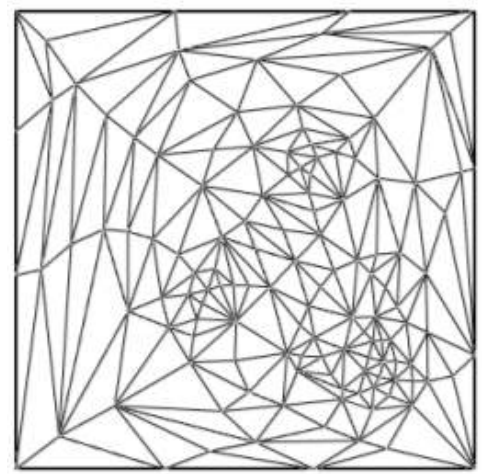
Texture Mapping Progressive Meshes, Sander et al, SIGGRAPH 2001

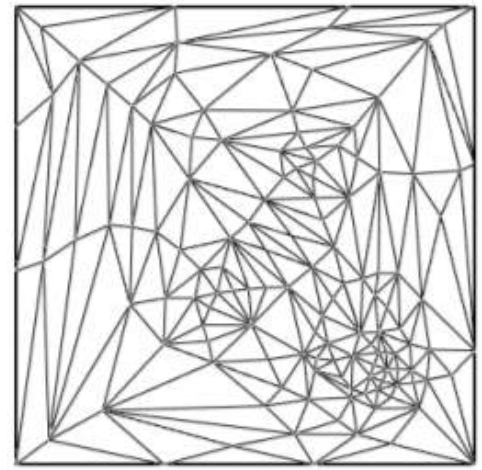
(b) Floater [5] L^2=2.26 L^∞=9.86

(c) MIPS [11] L^2=2.31 L^∞=7.46



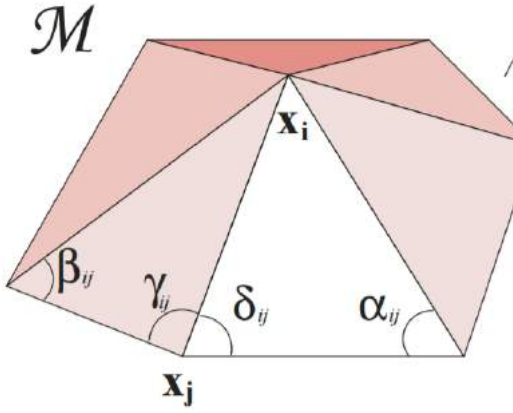

(f) our L^2 stretch L^2=1.22 L^∞=2.13




(g) our L^∞ stretch L^2=1.28 L^∞=1.65

Dirichlet Energy / Cotangent weights

$$\frac{\partial E_A}{\partial \mathbf{u}_i} = \sum_{j \in N(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) (\mathbf{u}_i - \mathbf{u}_j) = 0$$



- Harmonic Map (Eck 94, Pinkall & Polthier 93)
- Can be negative! But, **Symmetric** (good for linear solvers)
- Gradient of 1-ring area wrt $\mathbf{u}_i$

Computing Discrete Minimal Surfaces and their Conjugates, Pinkall & Polthier, 1993

Isometric / Conformal / Authalic

- Isometric: Preserve Angles and Areas
  - Impossible
- Conformal: Preserve Angles
  - Harmonic
- Authalic: Preserve Areas
  - ???

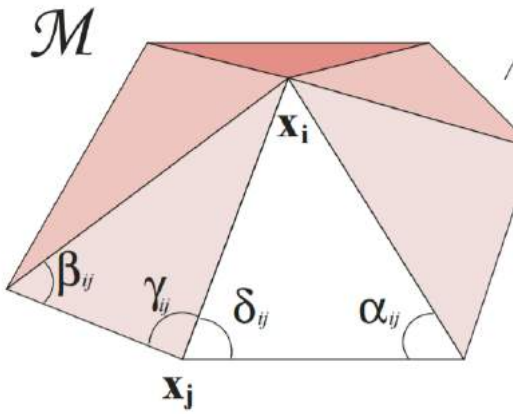
Intrinsic Parameterization of Surface Meshes, Desbrun et al, Eurographics 2002

Authalic Parameterization

- “Chi Energy”

$$\frac{\partial E_\chi}{\partial \mathbf{u}_i} = \sum_{j \in N(i)} \frac{(\cot \gamma_{ij} + \cot \delta_{ij})}{|\mathbf{x}_i - \mathbf{x}_j|^2} (\mathbf{u}_i - \mathbf{u}_j) = 0$$

- Preserves relative areas of one-rings, “locally”



Intrinsic Parameterization of Surface Meshes, Desbrun et al, Eurographics 2002

Natural Conformal Parameterization (DNCP)

- “the derivative of the Dirichlet energy on a triangle with respect to one of its vertices is equal to the opposite edge rotated by 90 degrees”

$$\sum_{\Delta_{ijk}} \cot\alpha(\mathbf{u}_i - \mathbf{u}_j) + \cot\beta(\mathbf{u}_i - \mathbf{u}_k) = \sum_{\Delta_{ijk}} R^{90}(\mathbf{u}_k - \mathbf{u}_j)$$

- Couples x and y!
- Still have to (arbitrarily) pin two vertices to fix rotation/translation

Intrinsic Parameterization of Surface Meshes, Desbrun et al, Eurographics 2002

Dirichlet Energy

$$E_{\mathrm{D}}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^t L_{\mathrm{D}}\mathbf{u}$$

Mesh Area

$$\mathcal{A}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^t A\mathbf{u}$$

Conformal Energy

$$E_{\mathrm{C}}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^t L_{\mathrm{C}}\mathbf{u}$$

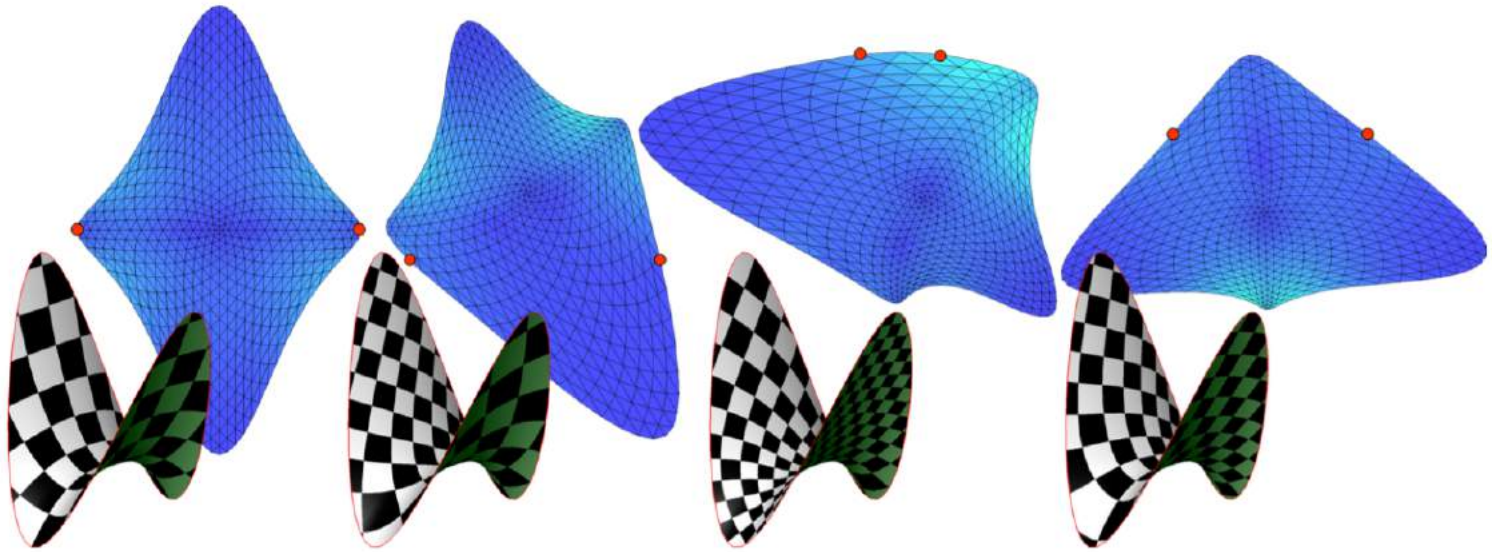
$$L_{\mathrm{C}} = L_{\mathrm{D}} - A$$

DNCP Matrix

By Definition: $A(\mathbf{u}) <= E_{\mathrm{D}}(\mathbf{u})$

DNCP = LSCM

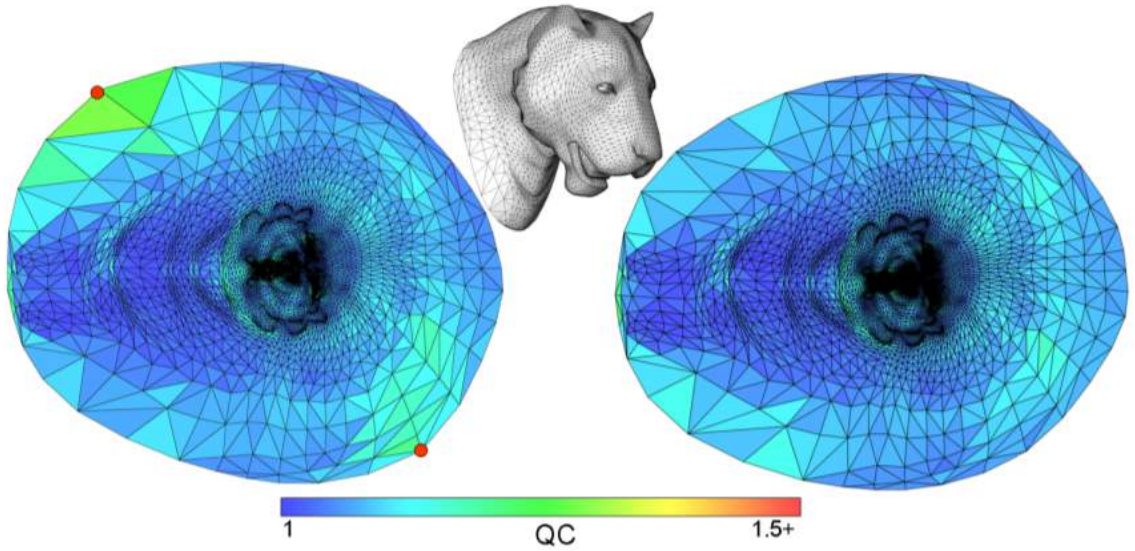
- Least Squares Conformal Maps published at same time as DNCP
- Math looks different, but ultimately $E_{\mathrm{LSCM}}(\mathbf{u}) = E_D(\mathbf{u}) - \mathcal{A}(\mathbf{u})$
- Both share problem: two arbitrary vertices must be pinned, introduces distortion:

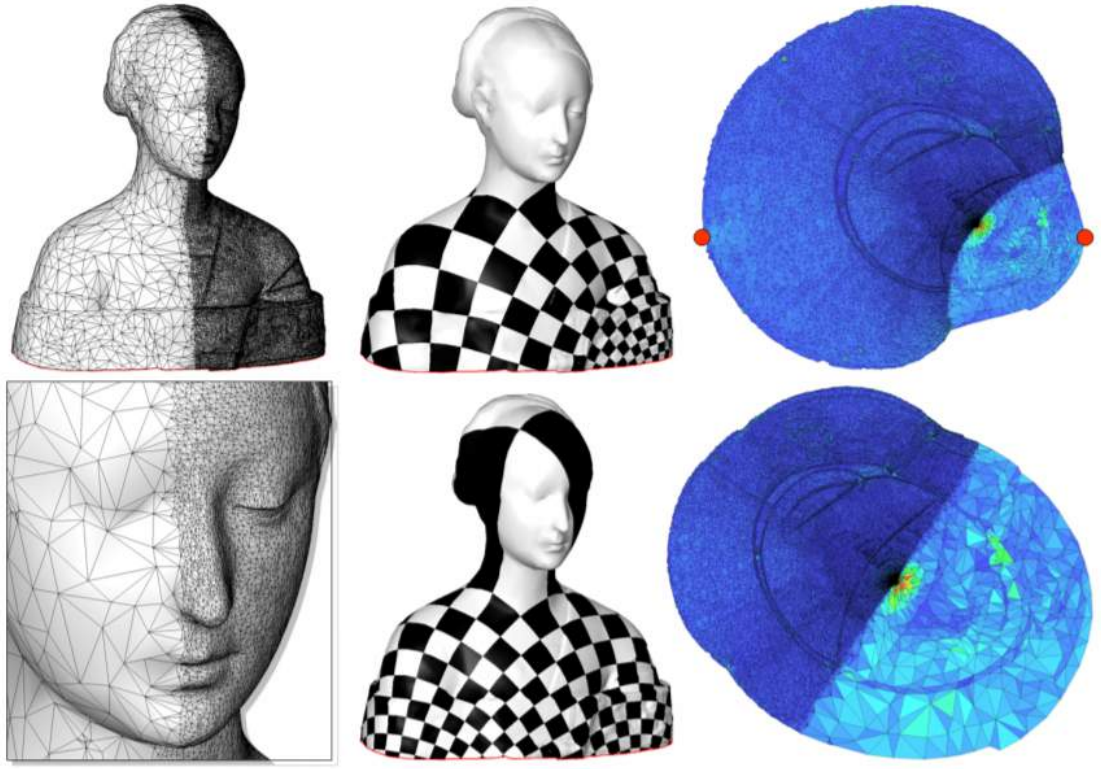


Least Squares Conformal Maps for Automatic Texture Atlas Generation, Levy et al, SIGGRAPH 2002

Spectral Conformal Parameterization

- Instead of using a linear solver, use an Eigensolver

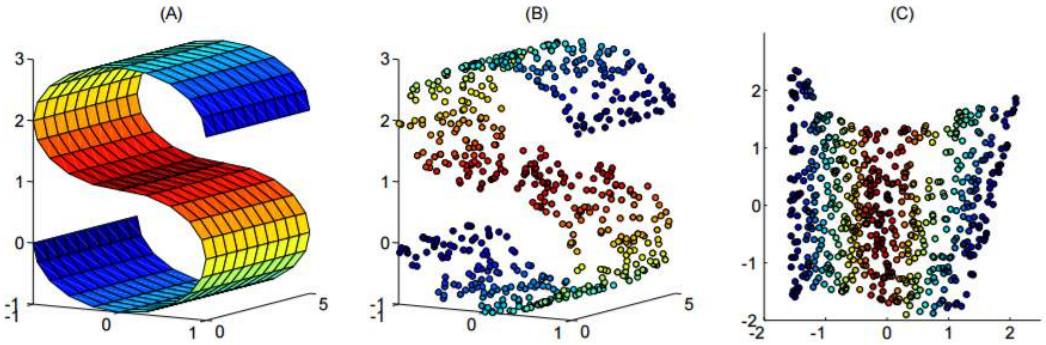




Spectral Conformal Parameterization, Mullen et al, SGP 2008

Machine Learning / Dimensionality Reduction

- Operate on points, not meshes

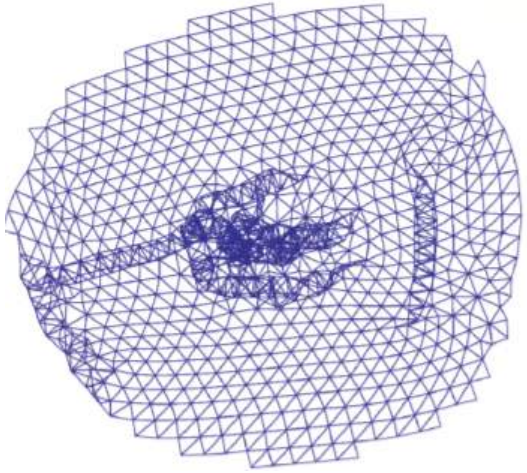


- LEM – Laplacian EigenMaps
  - Eigensolver applied to Dirichlet energy

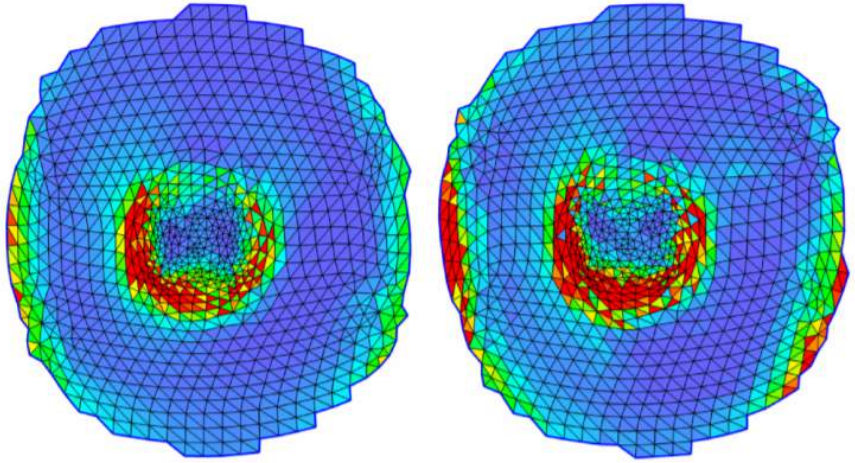
- LLE – Locally Linear Embedding
  - Weights found by local minimization of covariance in planar projection
  - Minimize Bilaplacian w/ Eigensolver

Dimensionality Reduction Cont.

- Maximum-Variance Unfolding
  - Use Semi-definite programming to attempt to preserve all edge lengths



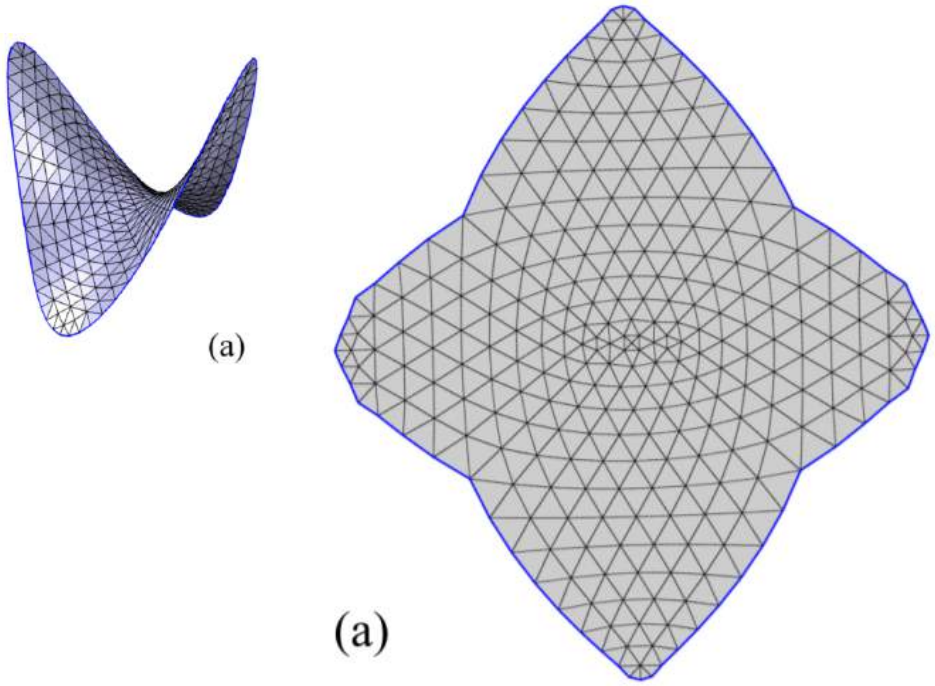
- Hessian Eigenmaps
  - LLE with local Hessians

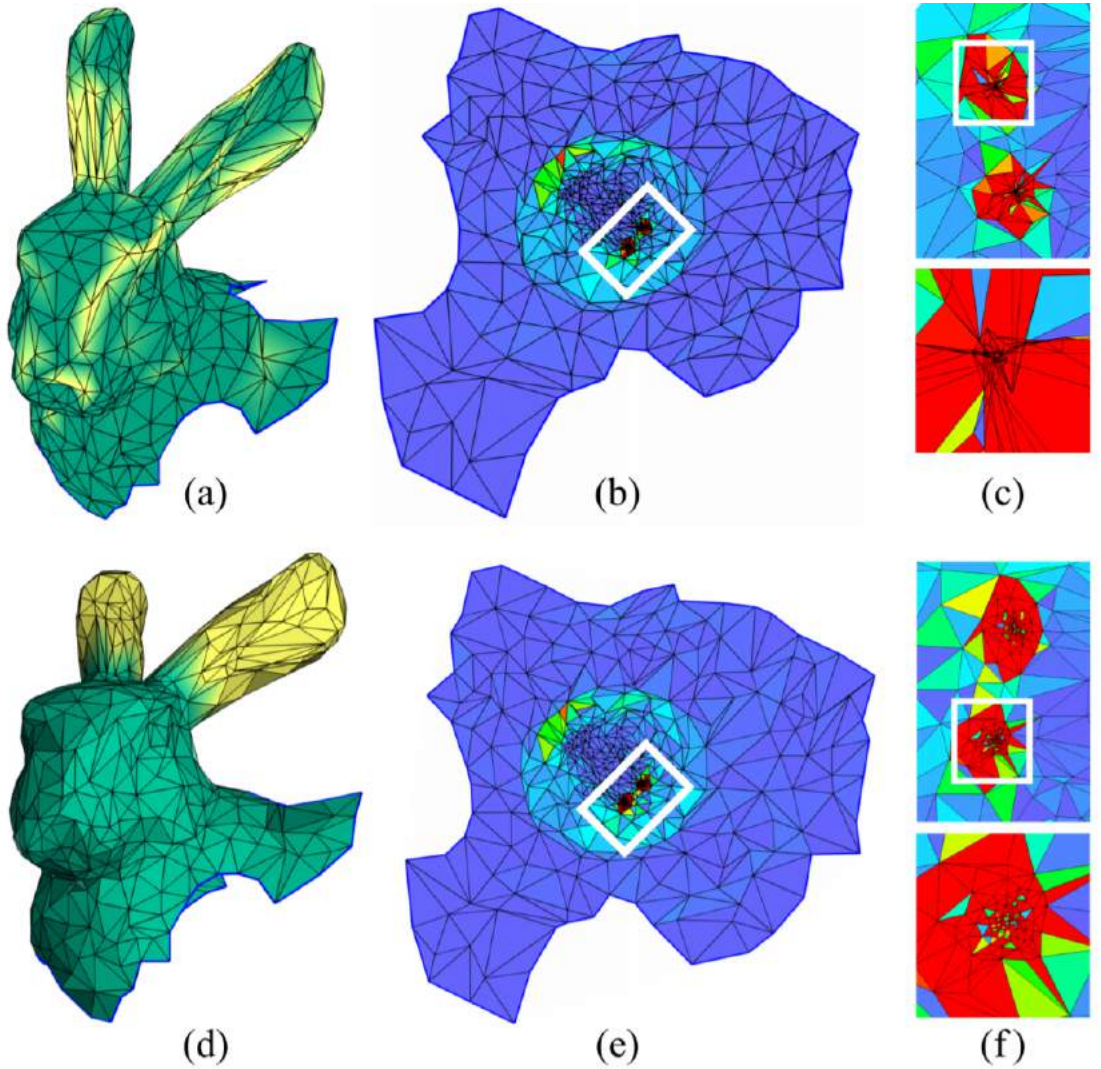


- LTSA: Local Tangent-Space Alignment
  - Flatten local regions separately, then solve stitching problem

Weird Stuff

Free-boundary Tutte Paramaterization

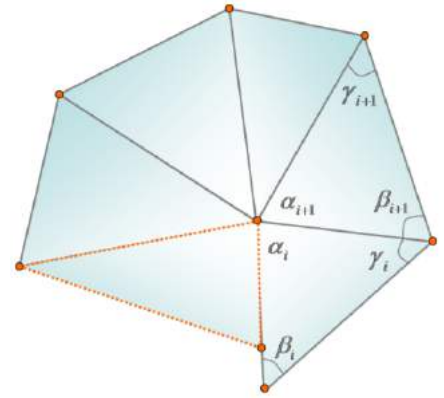




“Parameterization Repair” – use uniform weights where cotan < 0

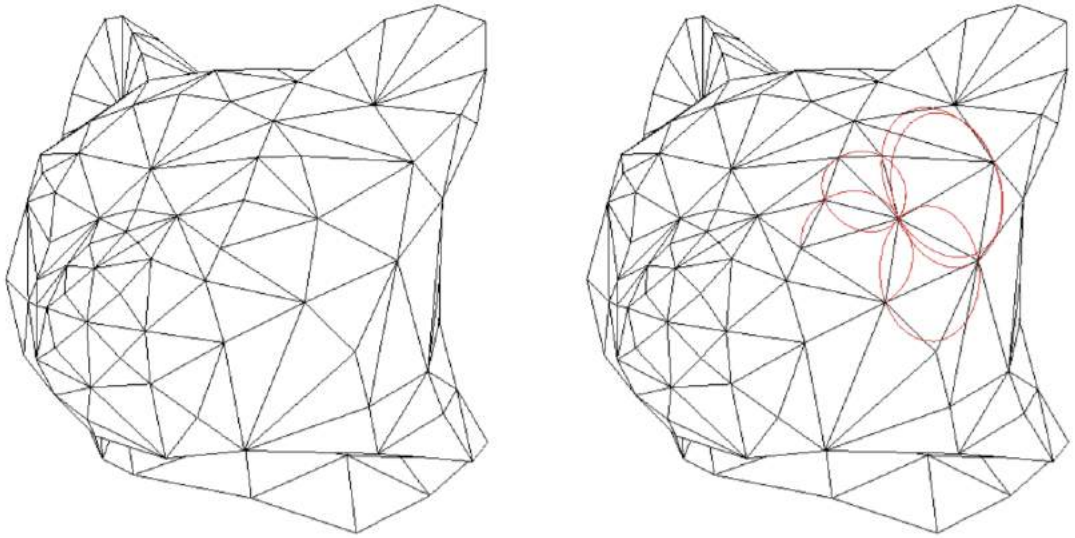
Conformal: Angle-Based Flattening

- Iterative nonlinear optimization to explicitly preserve one-ring angles
  - Sum of ring angles = 2pi, sum of triangle angles = pi
  - “Wheel consistency” – one-rings match up
- Provable convergence, but slow
- ABF++: faster, but still iterative solve
- LABF: linearized, “less” conformal but ~10x faster
- Even the Linear variant handles some situations DNCP fails on
  - Because it does not have the negative-weights problem



Parameterization of Faceted Surfaces for Meshing using Angle-Based Flattening, Sheffer and de Sturler, 2001

Circle patterns



- Conformal == Preserves Circles and circle intersections

- Place circle at each face
- They intersect at vertices
- Solve for radii of 2D circles that preserve angles

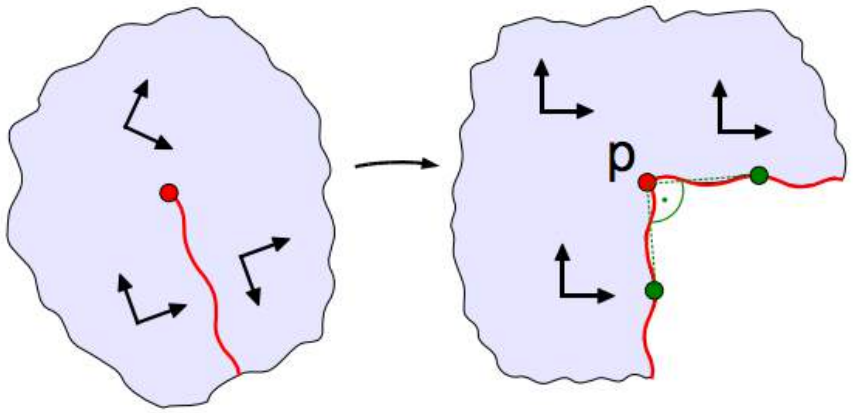
- Hard.

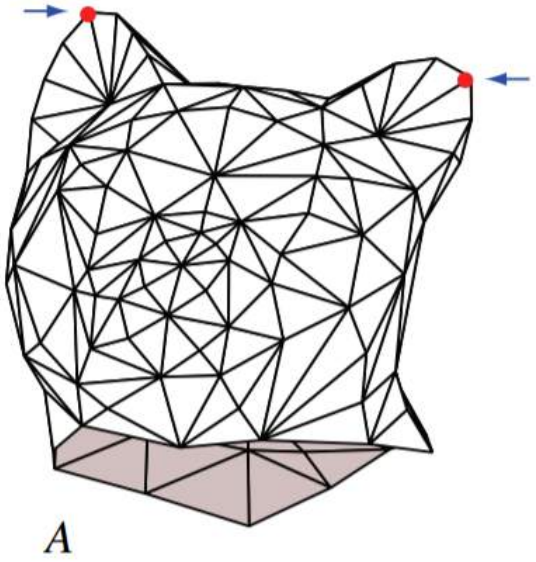
Discrete Conformal Parameterizations via Circle Patterns, Kharevych et al, TOG 2006

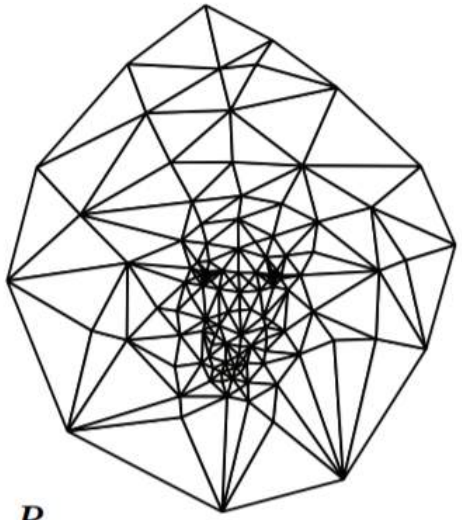
Cone Singularities

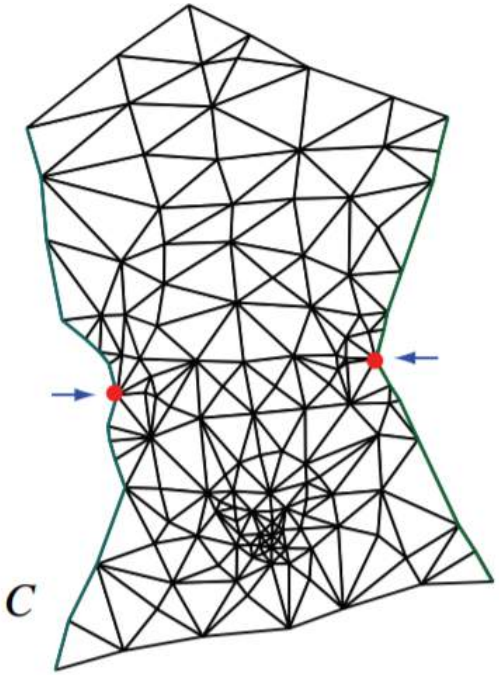
- Imagine you could just “redistribute” curvature between vertices
- “Push” all curvature to a few vertices, so the rest have 0 curvature
  - So mesh is Developable except at a few points
- Cut mesh between those points
- Now entire mesh is developable!
- If done properly, parameterization along cuts perfectly lines up

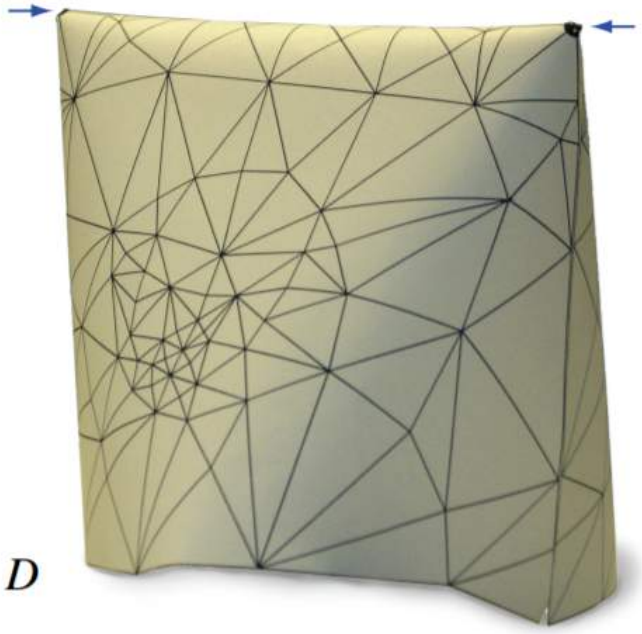
Cone Singularities



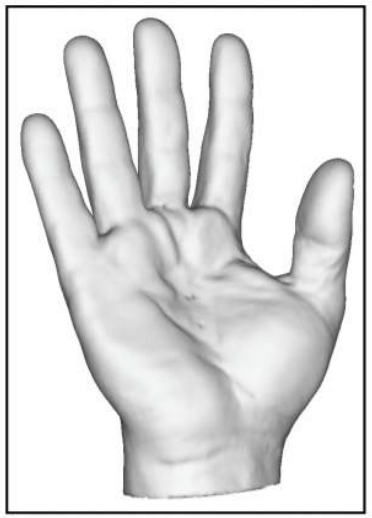


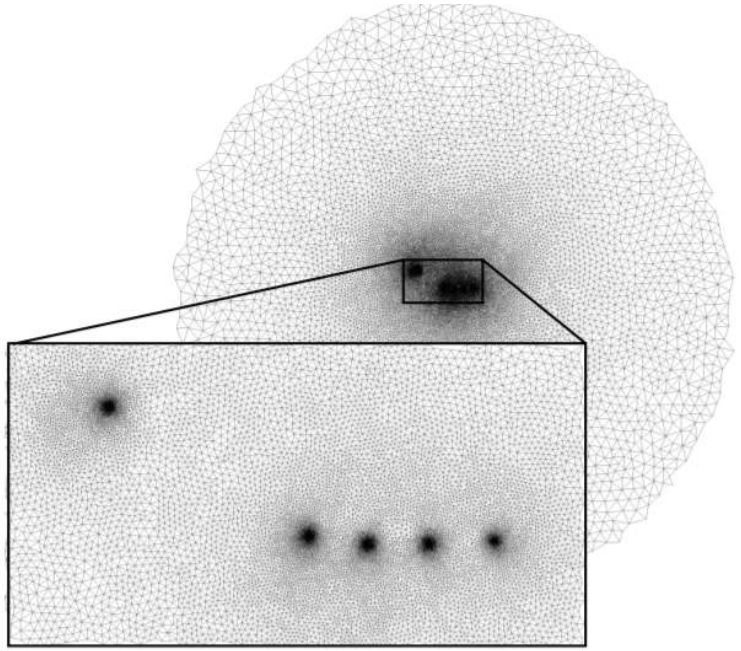


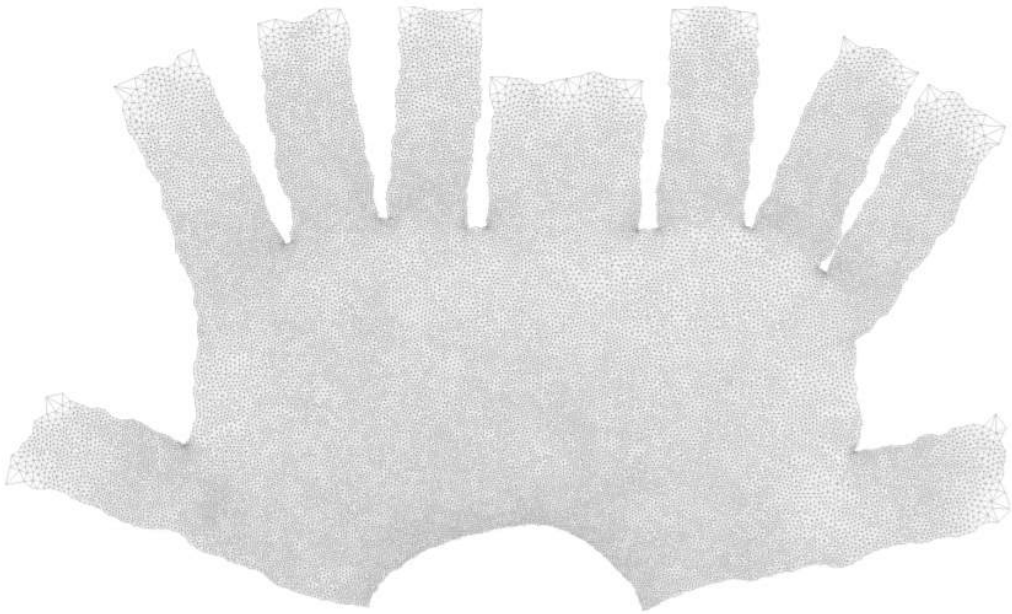




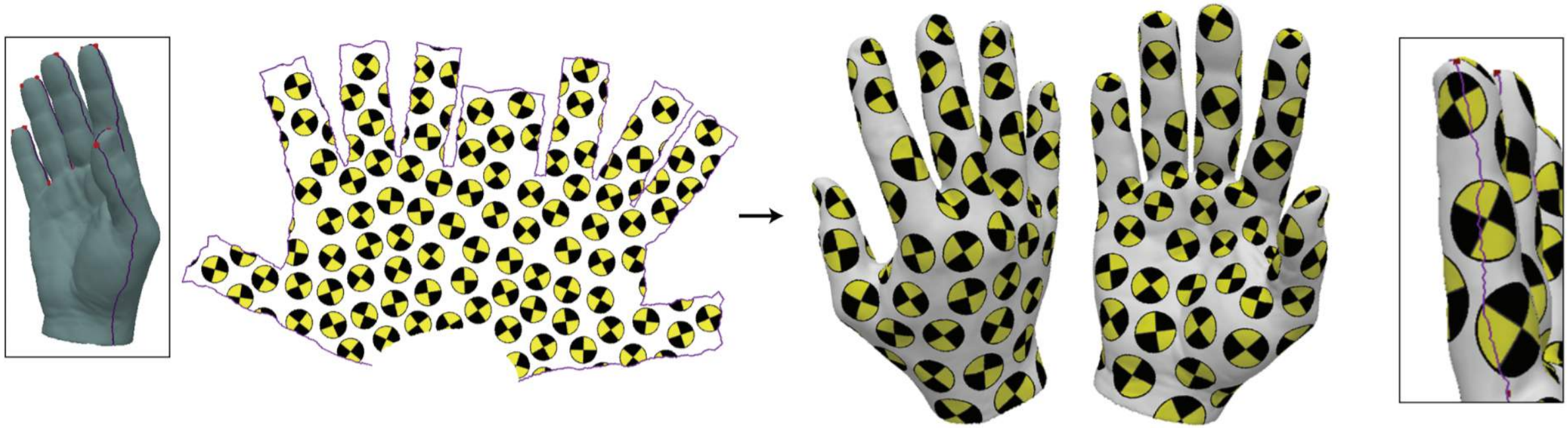
Circle Patterns w/ Cone Singularities





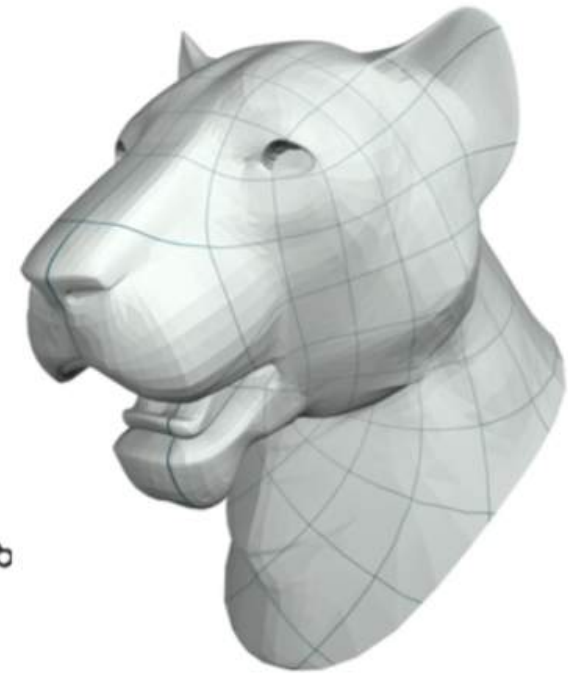


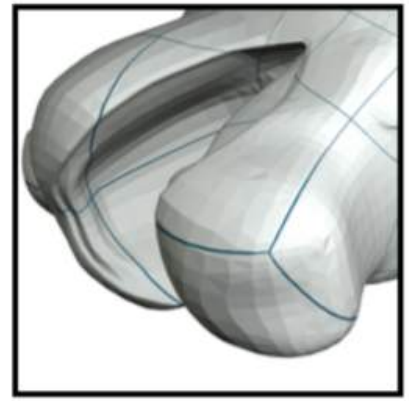
Circle Patterns w/ Cone Singularities



CONTINUOUS ACROSS CUTS!!!

Conformal Flattening by Curvature Prescription and Metric Scaling, Ben-Chen et al, Eurographics 2008
Discrete Conformal Parameterizations via Circle Patterns, Kharevych et al, TOG 2006

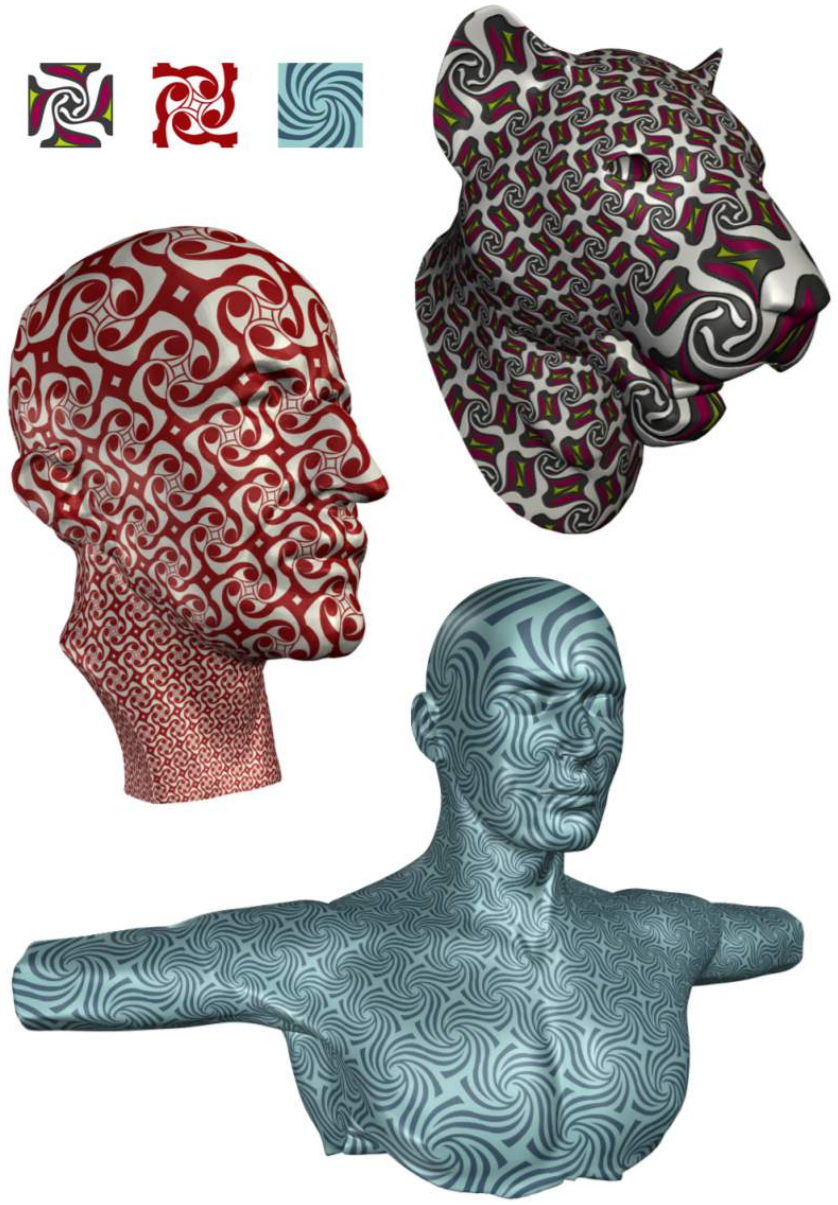




Conformal Equivalence

- DDG formulation of “conformal equivalence”
- Convex energy, efficient solution
  - Hessian is cot-Laplace op, of course
- Cone singularities are automatic
- Desired distortion can be prescribed
- **They win. Global Conformal Parameterization solved.**

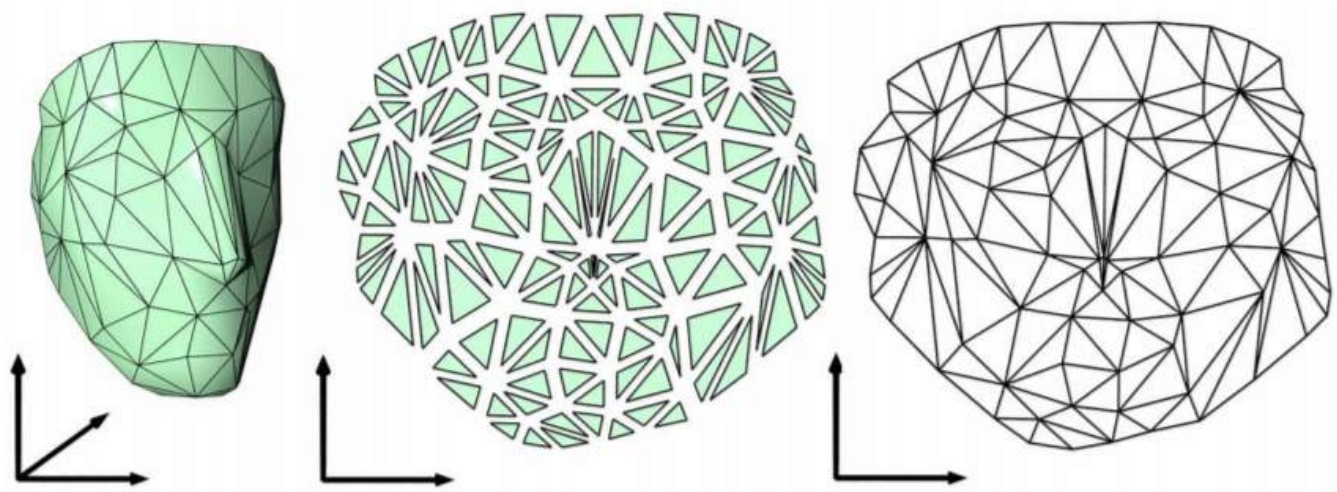
* Texture has to be periodic



Conformal Equivalence of Triangle Meshes, Springborn et al, SIGGRAPH 2008

ARAP: As Rigid As Possible

- Start from initial flattening (DNCP, Shape-Preserving)
- Iterate:
  - Solve for separate optimal rotation of each triangle
  - Solve for vertex positions that best satisfy rotations and triangle shapes (global Poisson problem)



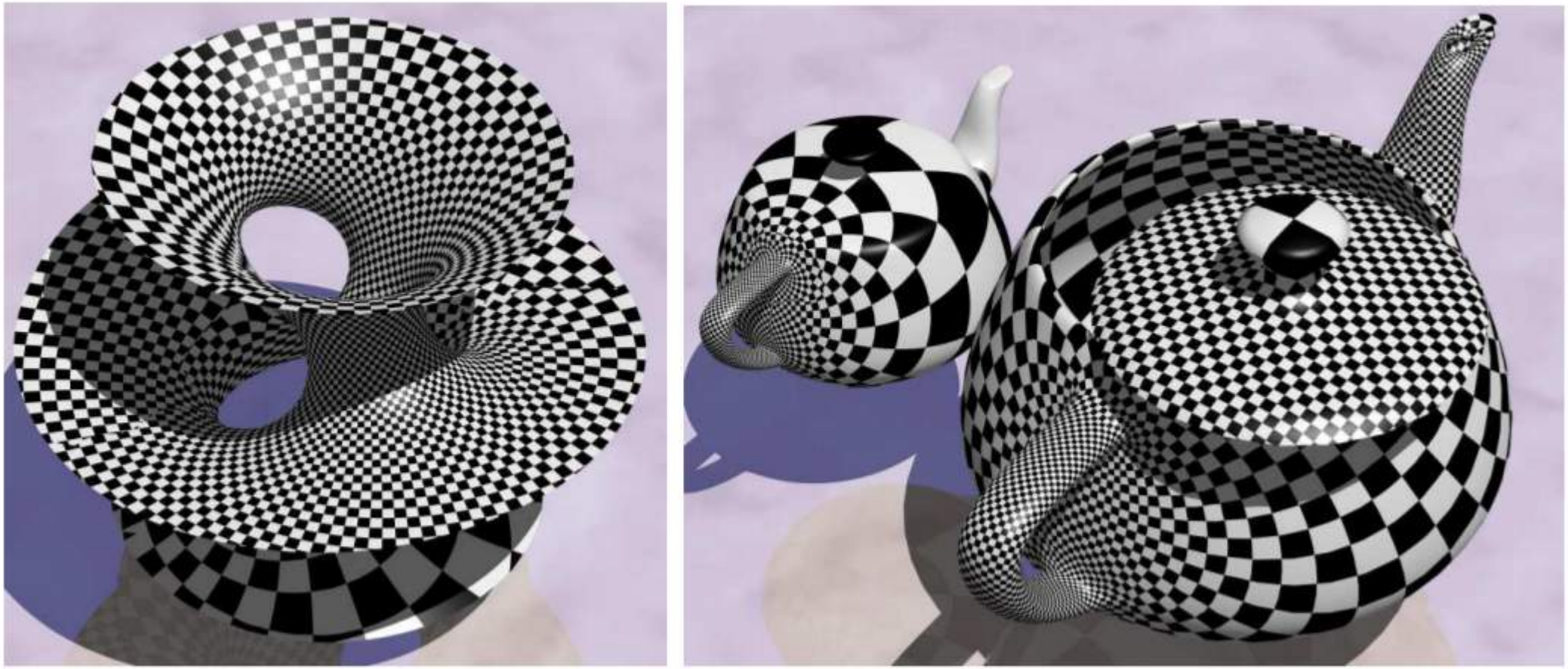
- “Unwrap” tool in Meshmixer

A Local/Global Approach to Mesh Parameterization, Liu et al, 2008

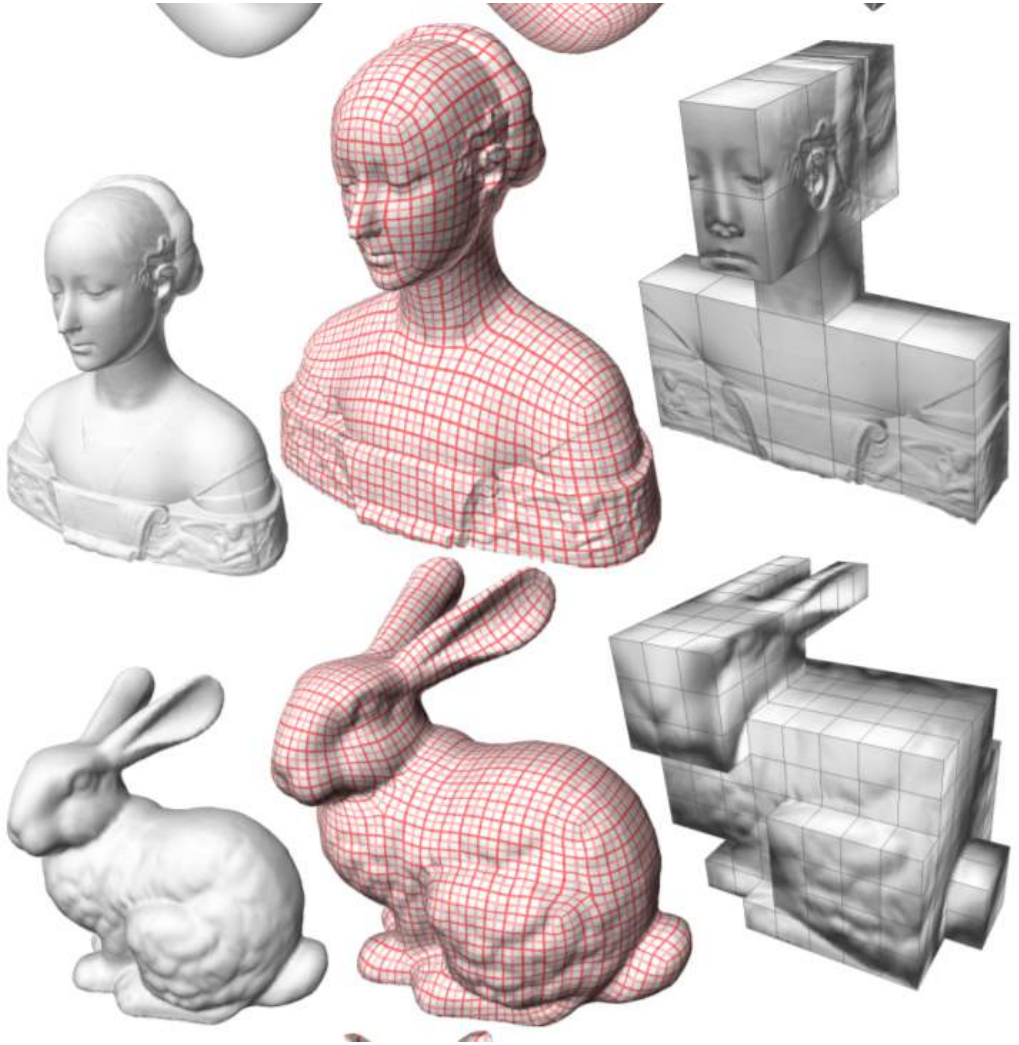
Spherical Parameterization

- Many adaptations: Spherical ABF, Conformal, ARAP, …

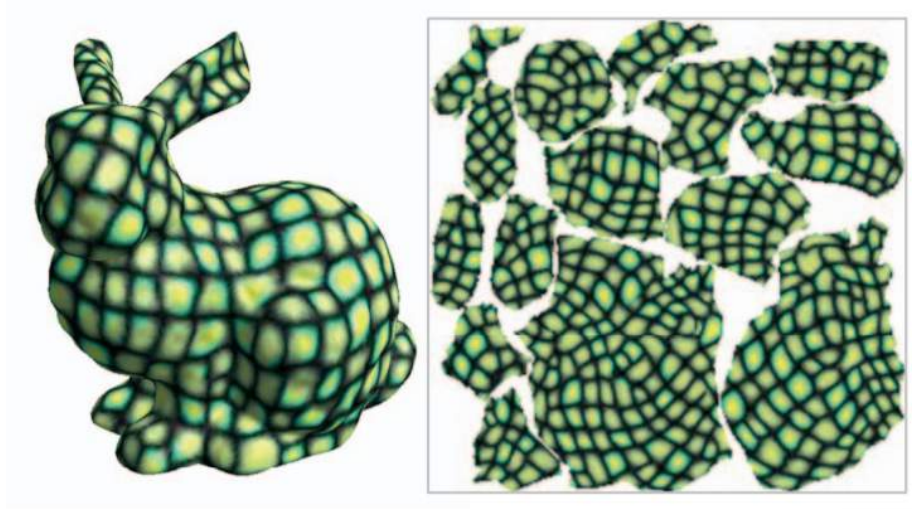
Toroidal Parameterization

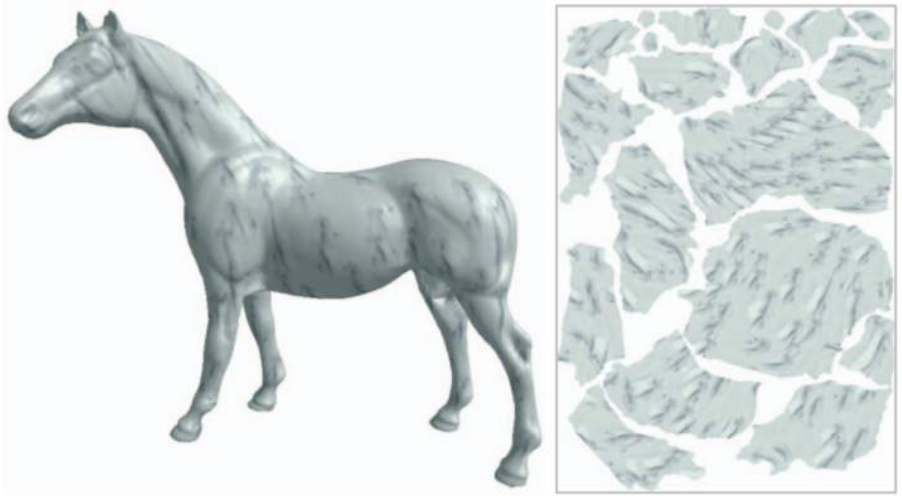


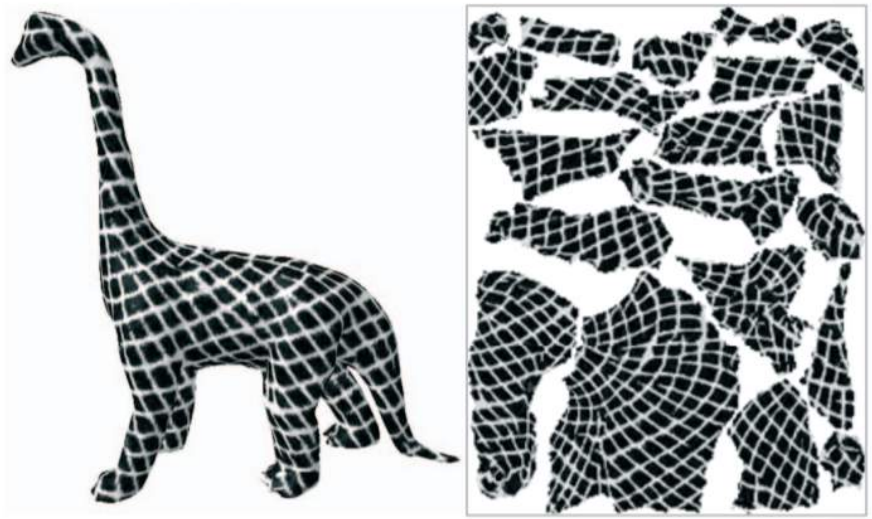
Polycube Maps

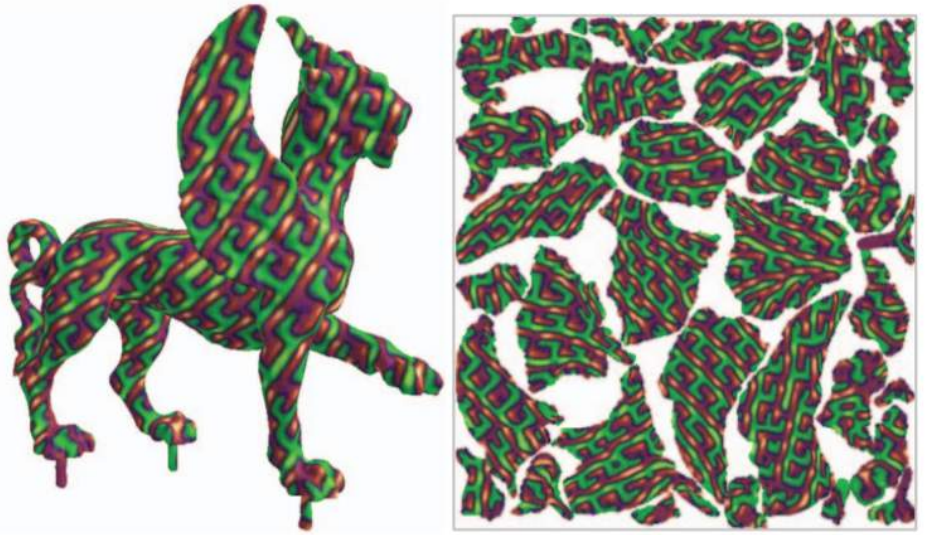


Seams / Charting

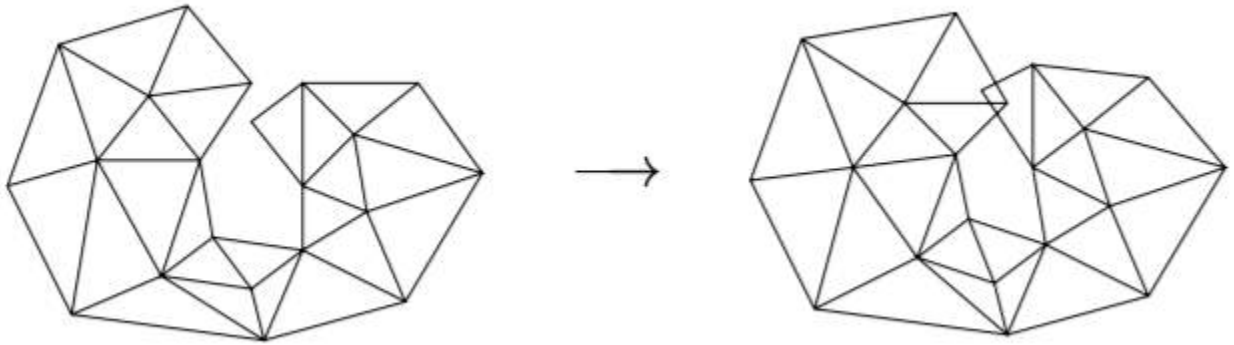


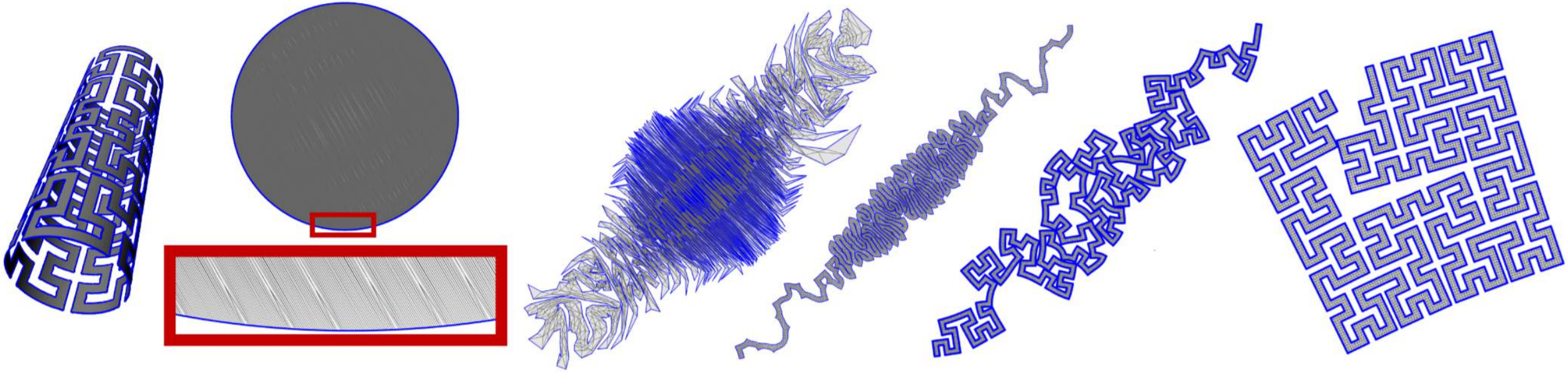






Global Bijections



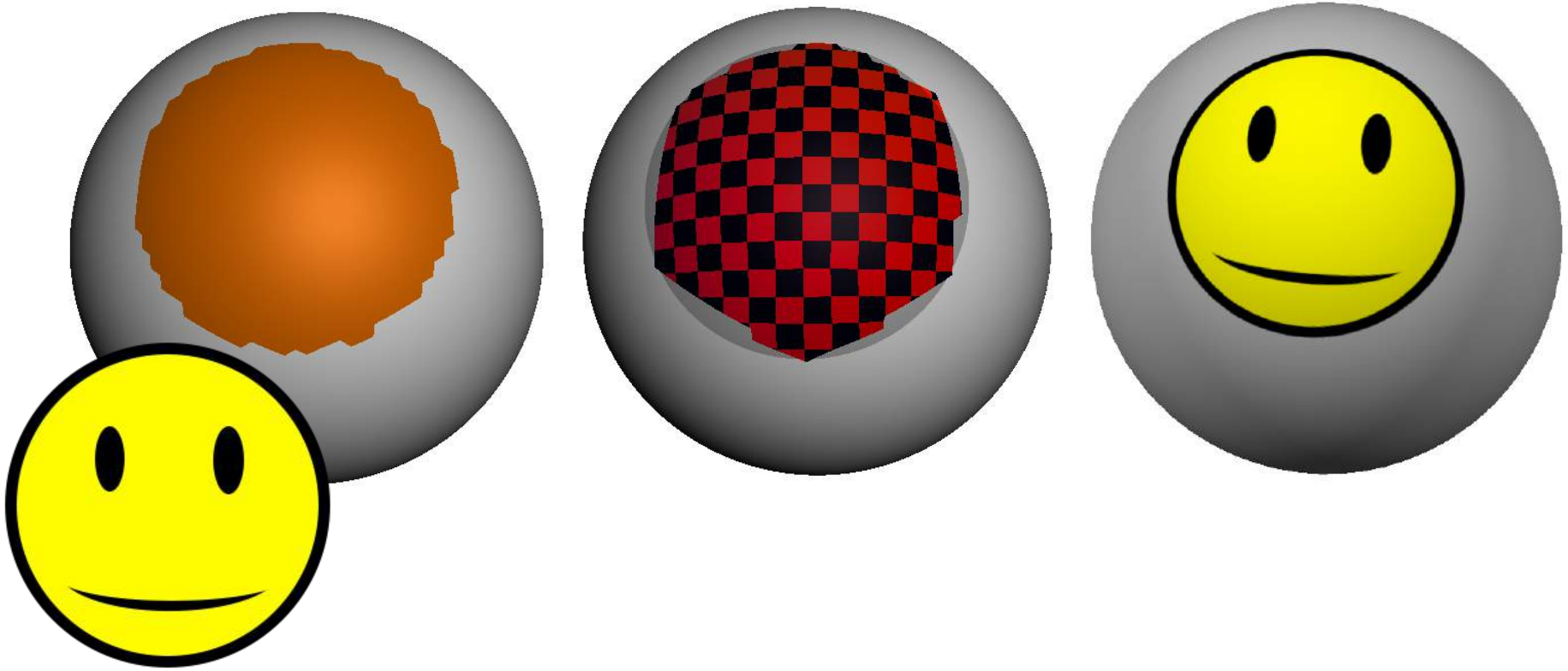


Local Parameterization…

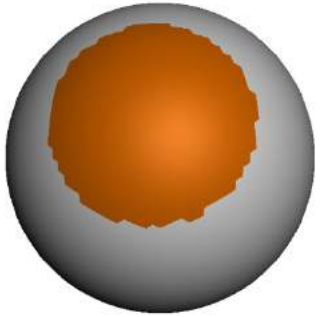
Local Parameterization

Ryan Schmidt

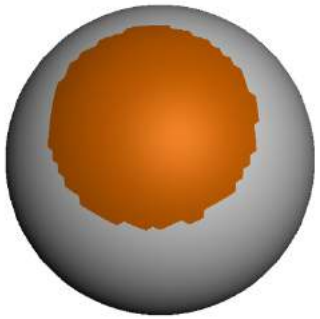
Decal Parameterization Problem



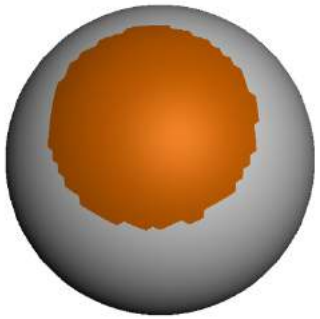
Global Parameterization
(Applied Locally)



Global Parameterization
(Applied Locally)



Global Parameterization
(Applied Locally)



Desirable Properties

- **Predictable** behavior
  - Distortion is consistent
  - Bounded UV-space
- Fast
- Smooth

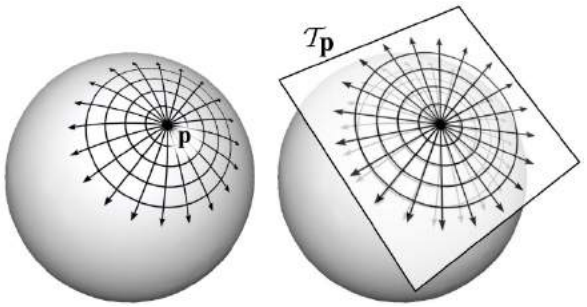
Usually!

Yes!

Sometimes!

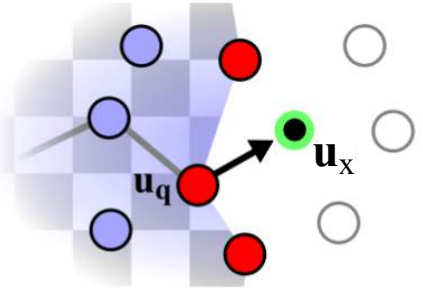
Discrete Exponential Map

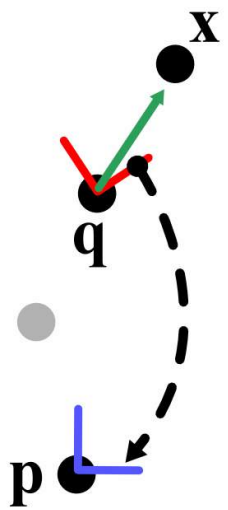
- Assume isometric maps exist between tangent spaces $T_{\mathbf{x}}$
  - Only true on Sphere
- Then transformation between tangent spaces is affine
- Incrementally lift neighbours of $\mathbf{p}$ into $T_{\mathbf{p}}$ via tangent-space vector summation

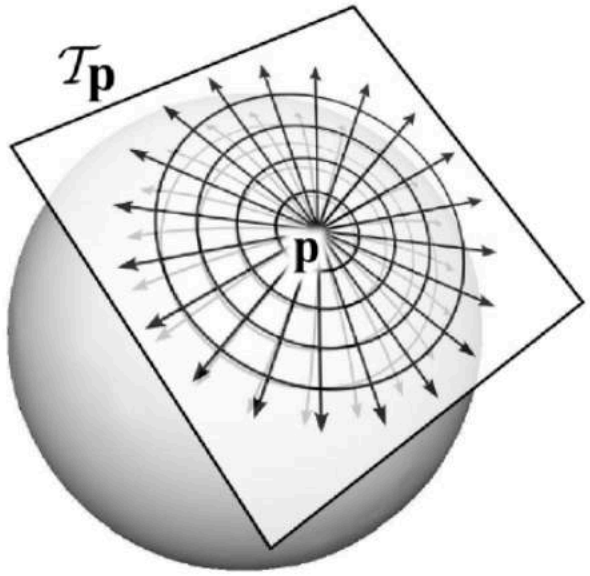


Discrete Exponential Map

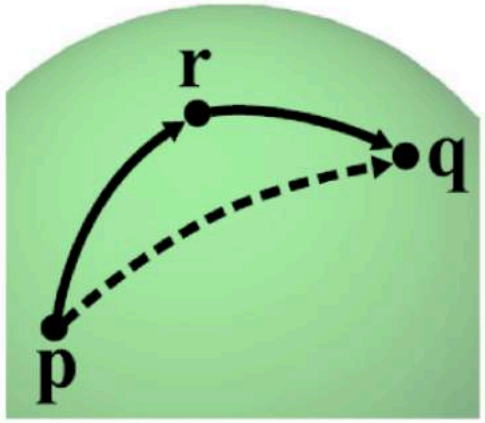
- Input is set of points and normals
  - Does not require mesh, just neighbours
- Propagate parameterization U from seed point $\mathbf{p}$
- For point $\mathbf{x}$ on propagation front
  1) Project $\mathbf{(x-q)}$ into tangent plane at $\mathbf{q}$
  2) Apply rotation R that takes frame at $\mathbf{q}$ to frame at $\mathbf{p}$
  3) Add to $U\mathbf{(q)}$ to get $U\mathbf{(x)}$





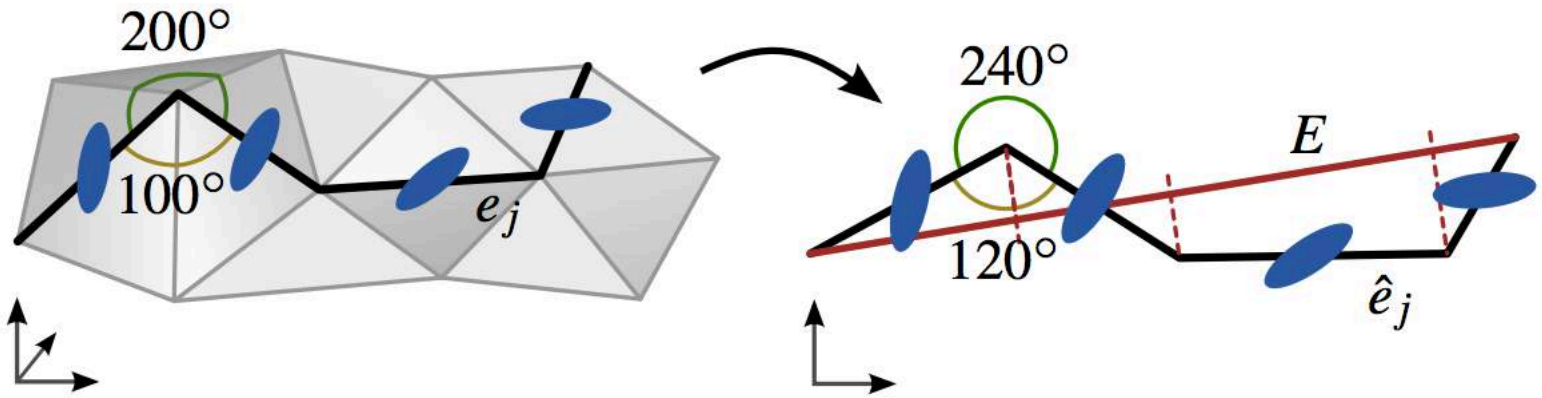


(a)



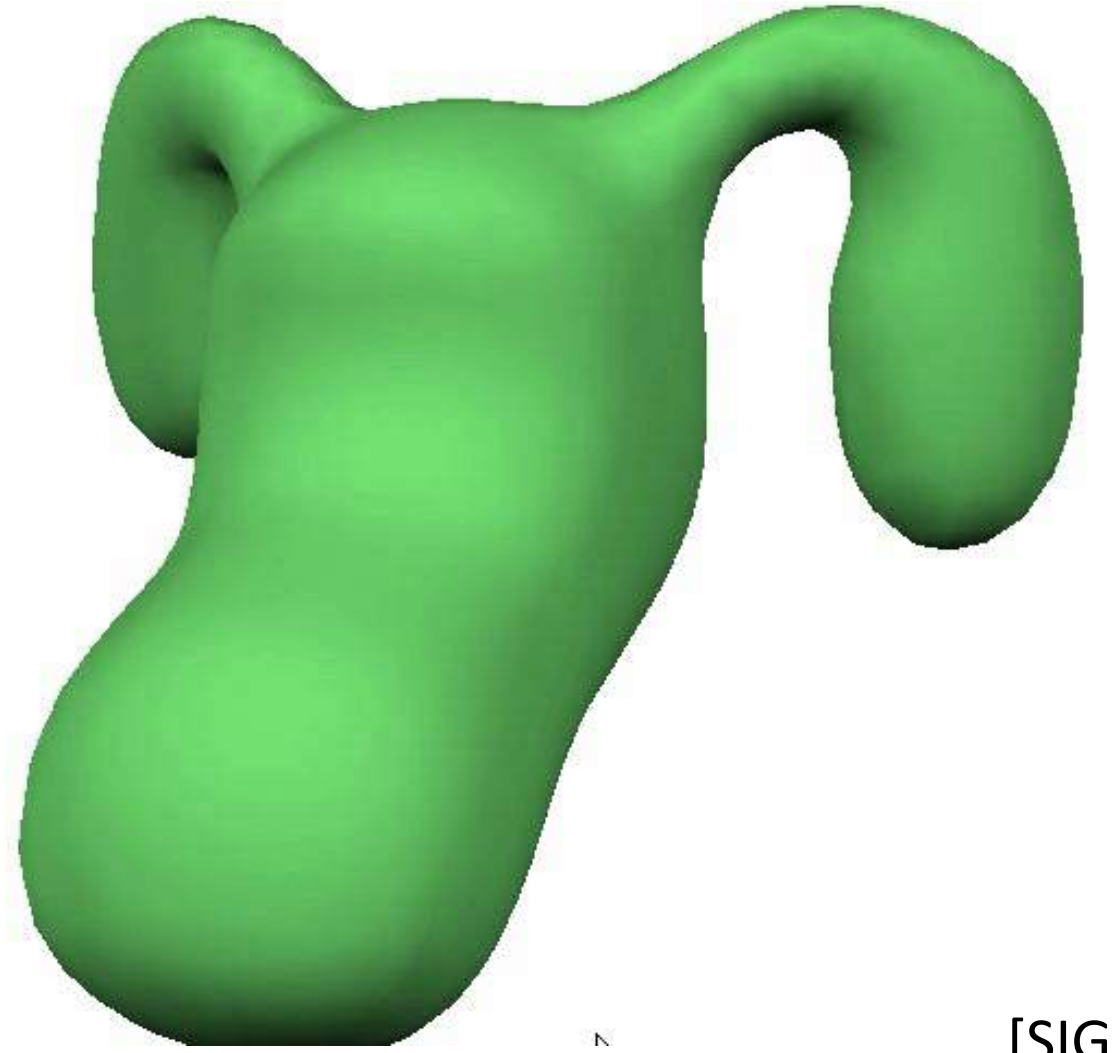
Radial Geodesics

(b)



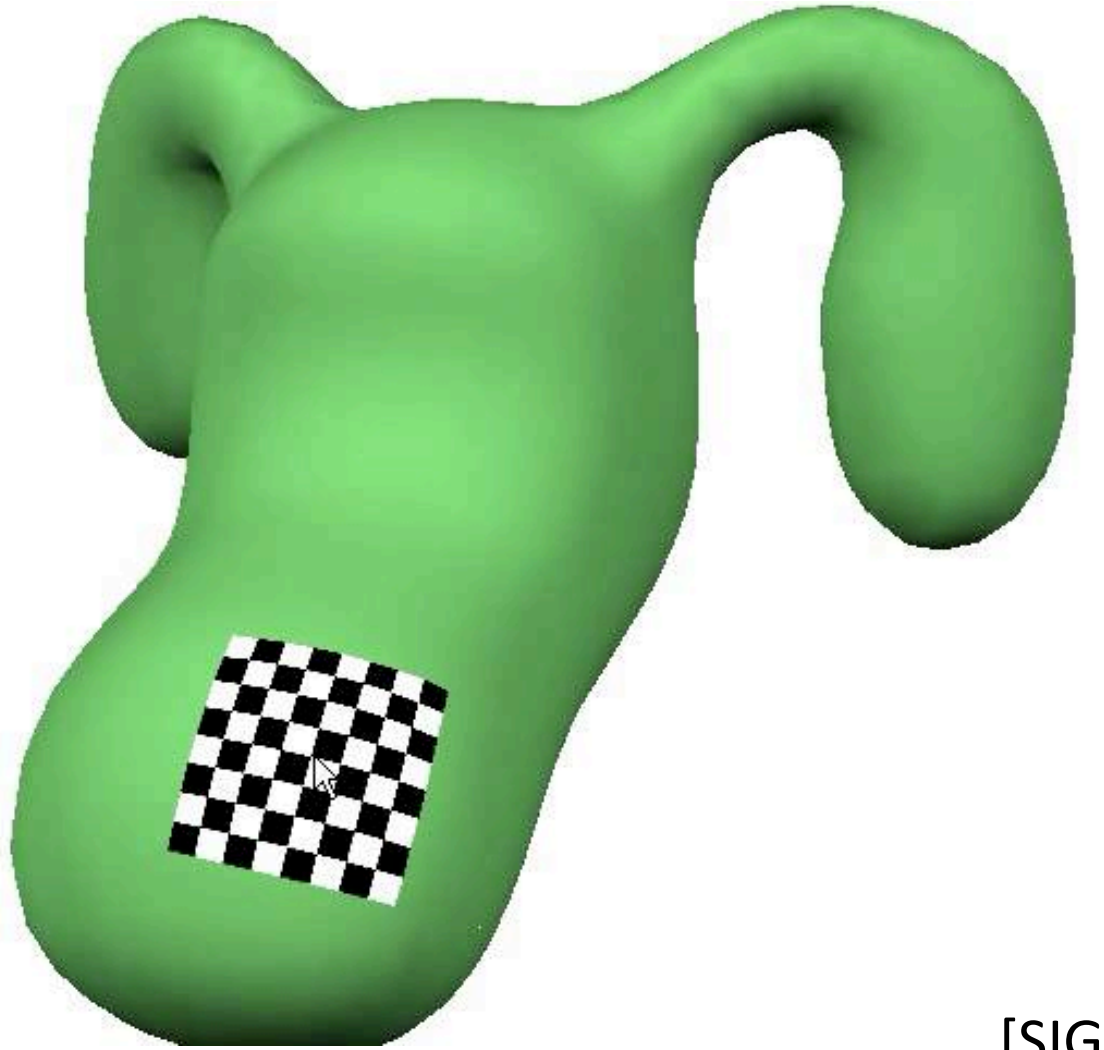
"Vector Dijkstra"

Discrete Exponential Map



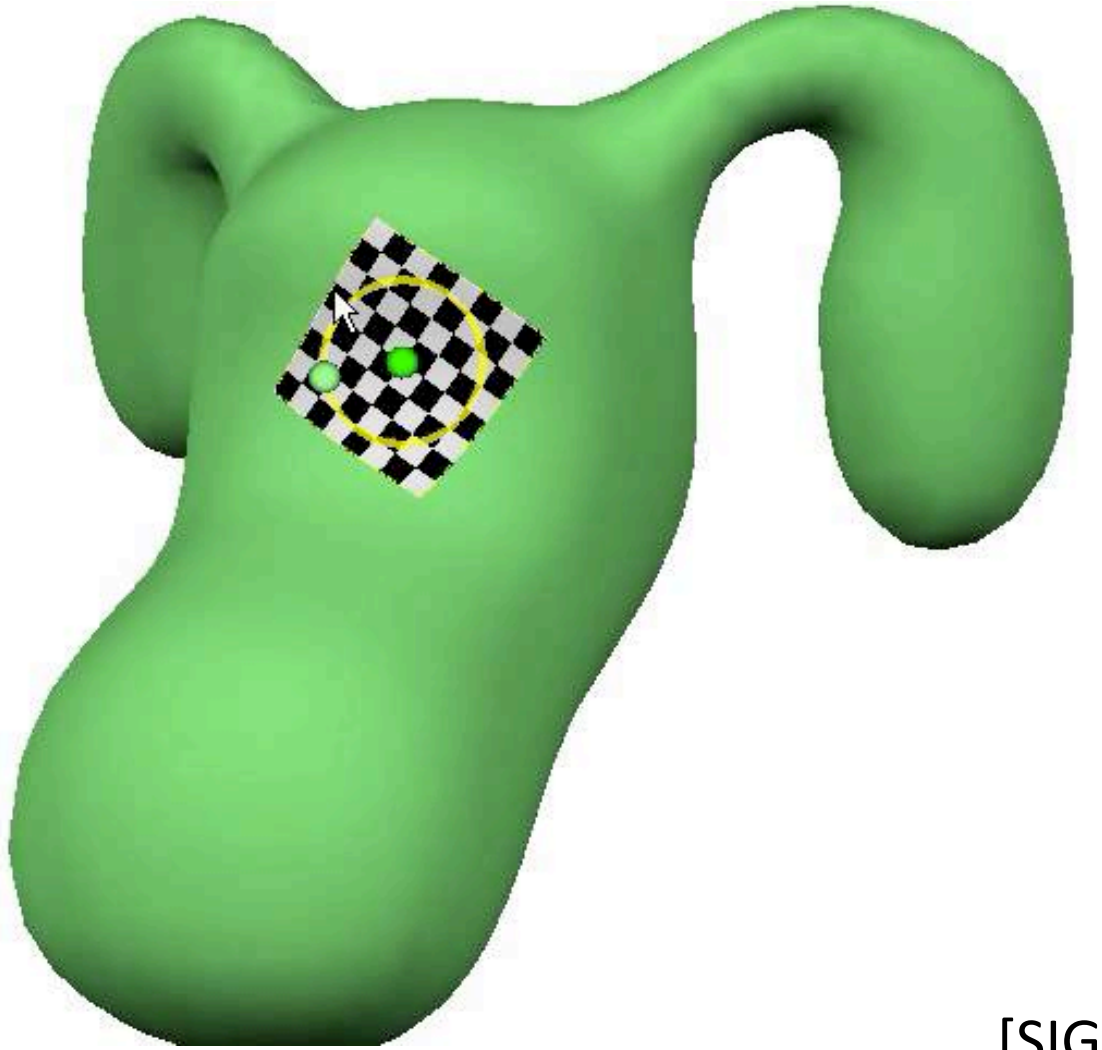
[SIGGRAPH 2006]

Discrete Exponential Map



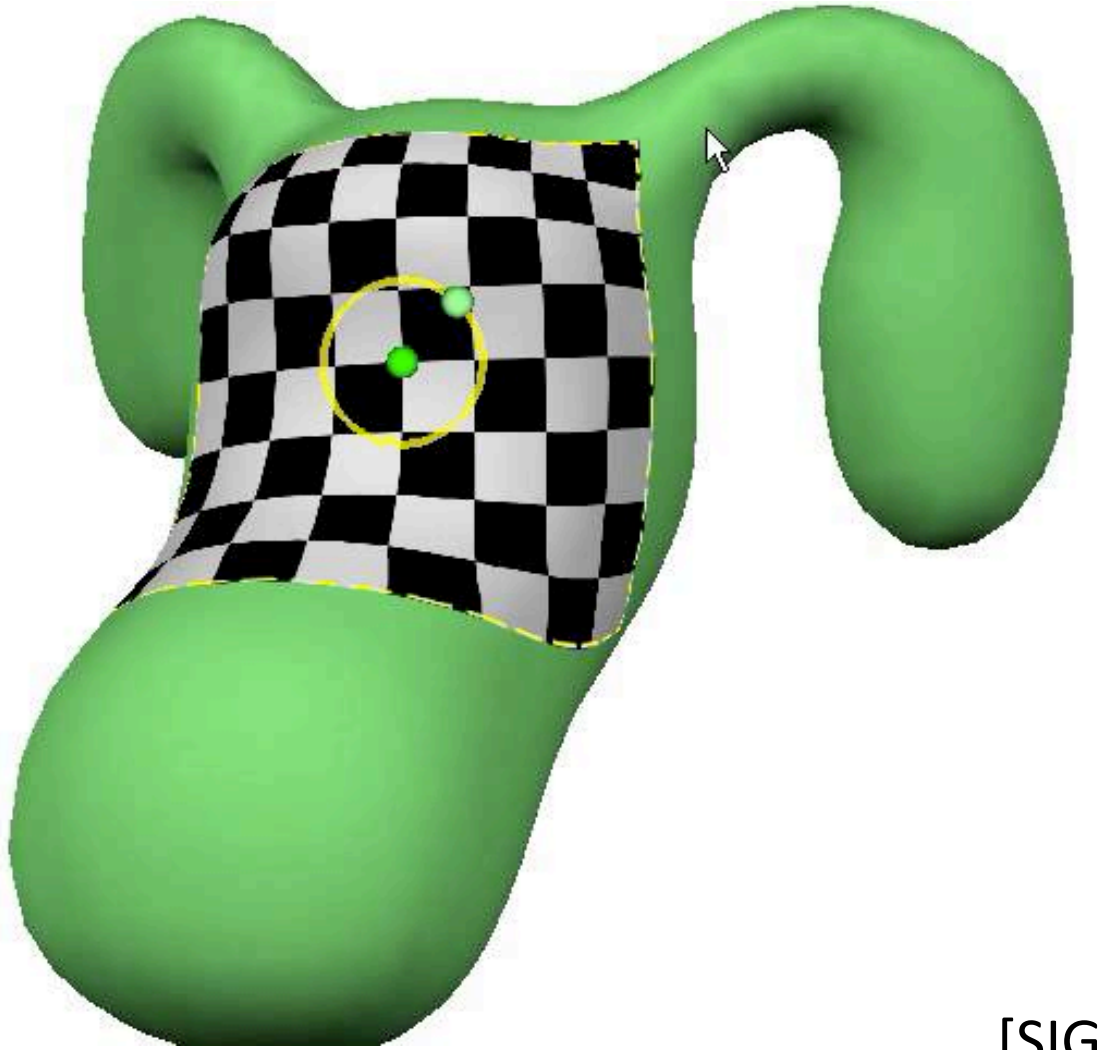
[SIGGRAPH 2006]

Discrete Exponential Map



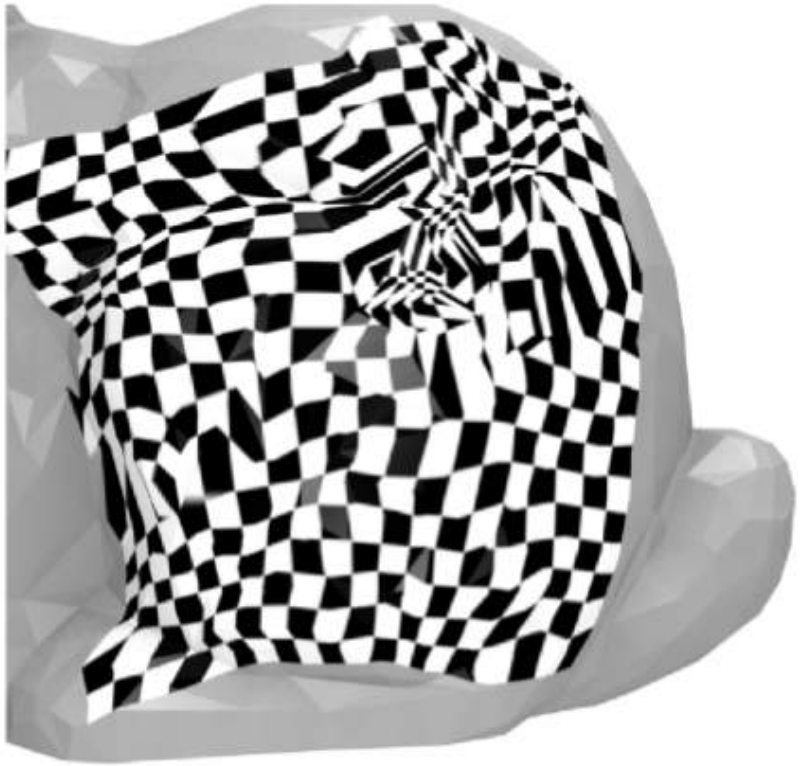
[SIGGRAPH 2006]

Discrete Exponential Map



[SIGGRAPH 2006]

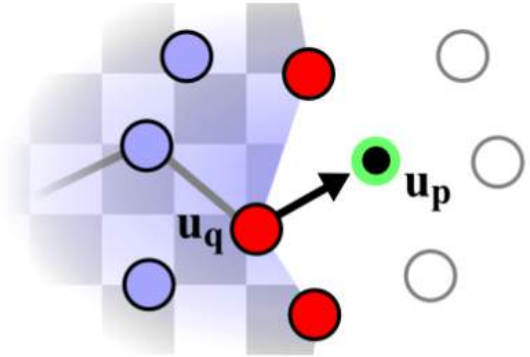
Bad meshes were a problem…



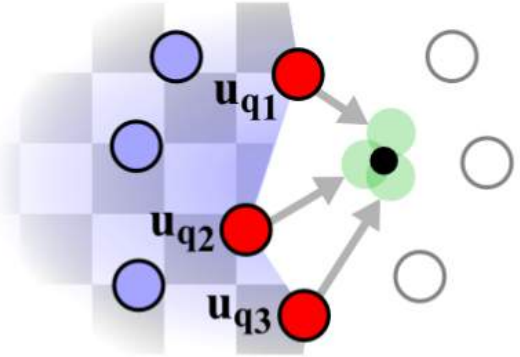
One bad projection ruins everything “downwind”

Improved Discrete ExpMap

1) Average uv-predictions from multiple upwind samples

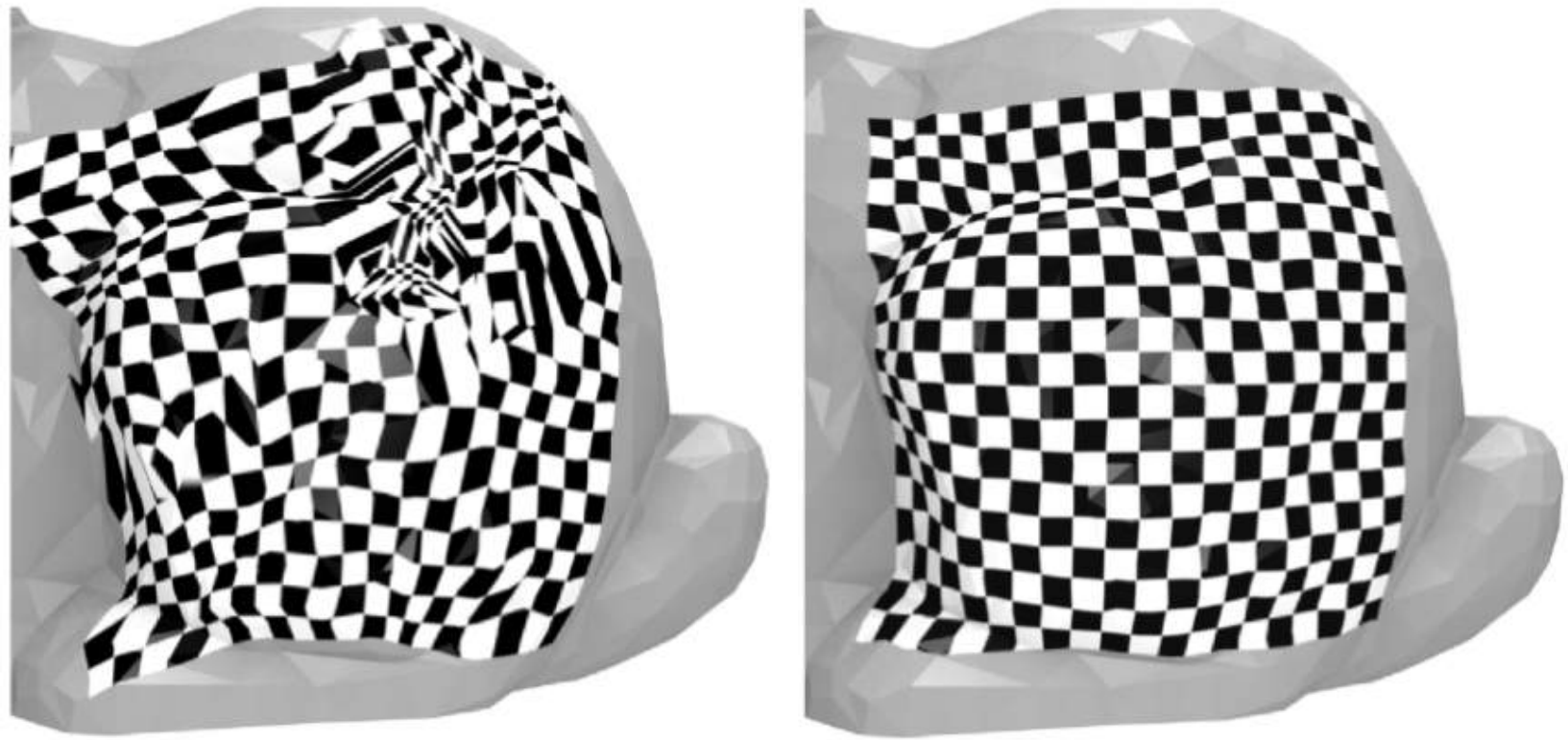


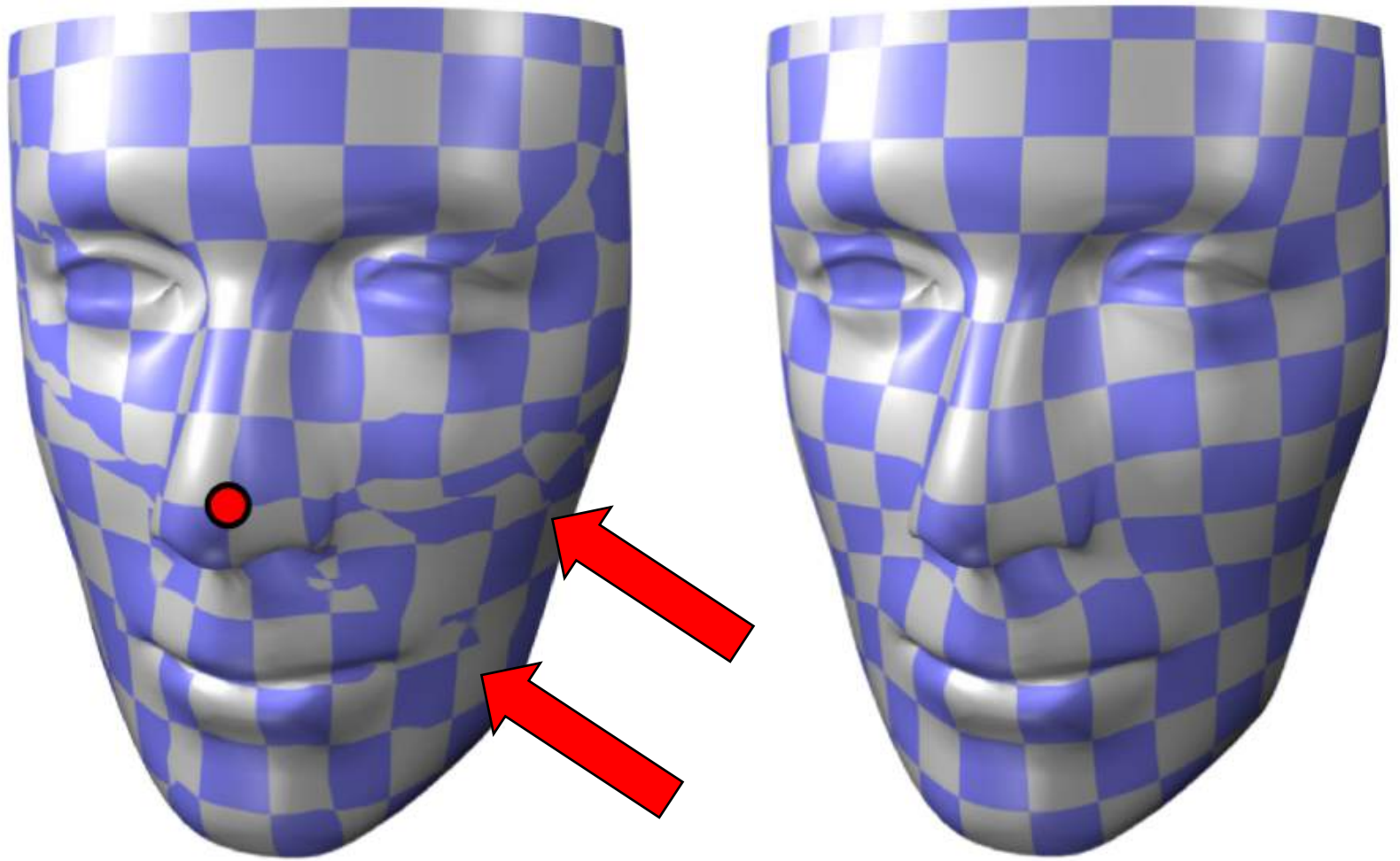
Original DEM

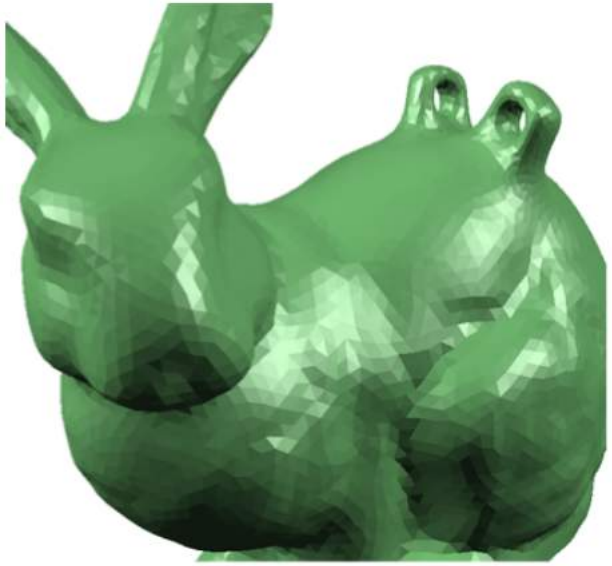


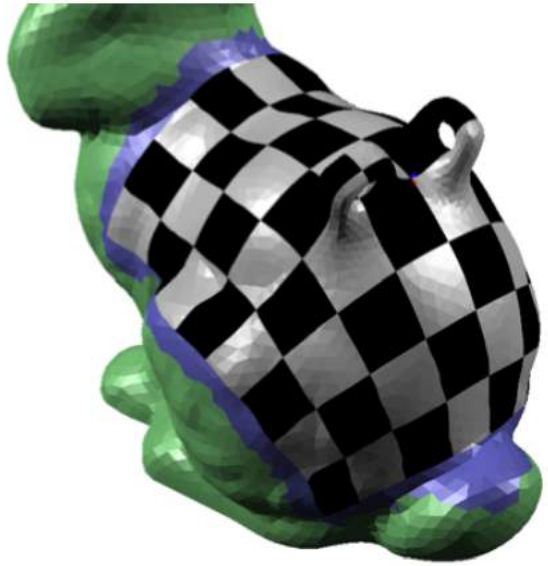
“Upwind-Average” DEM

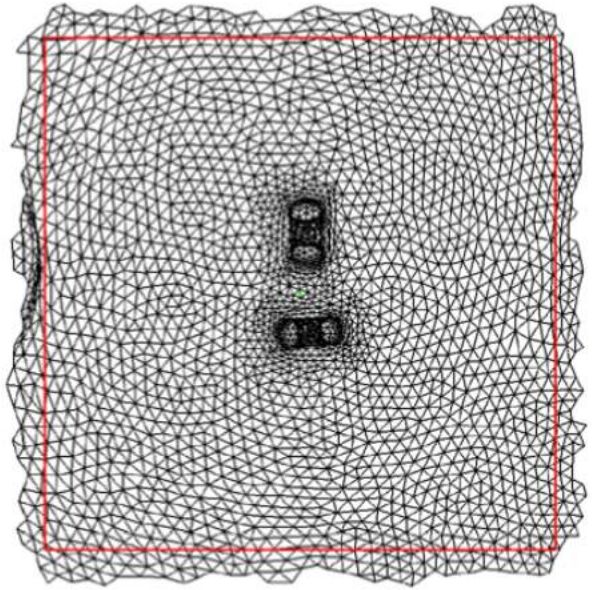
2) Use smoothed normal field





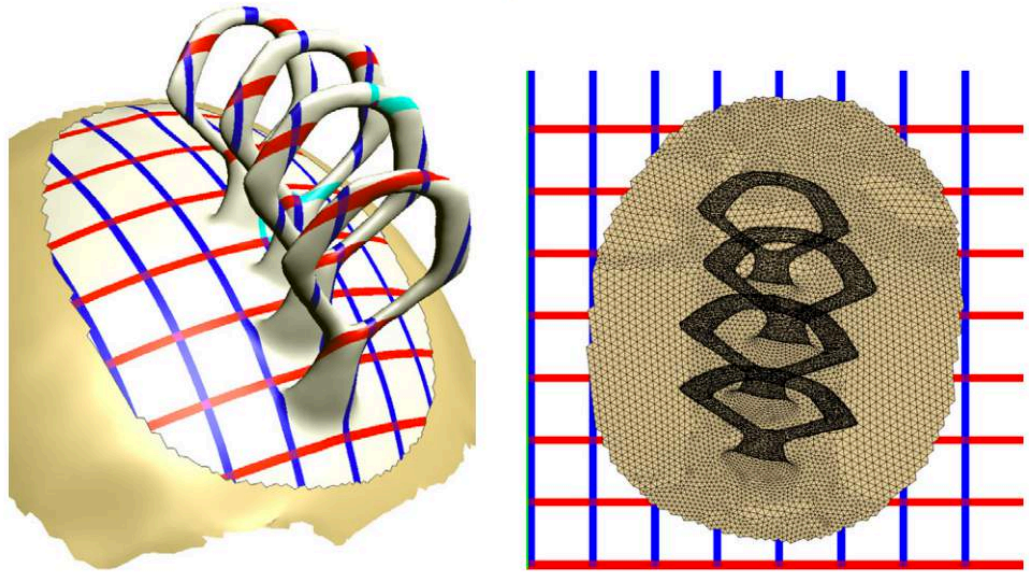




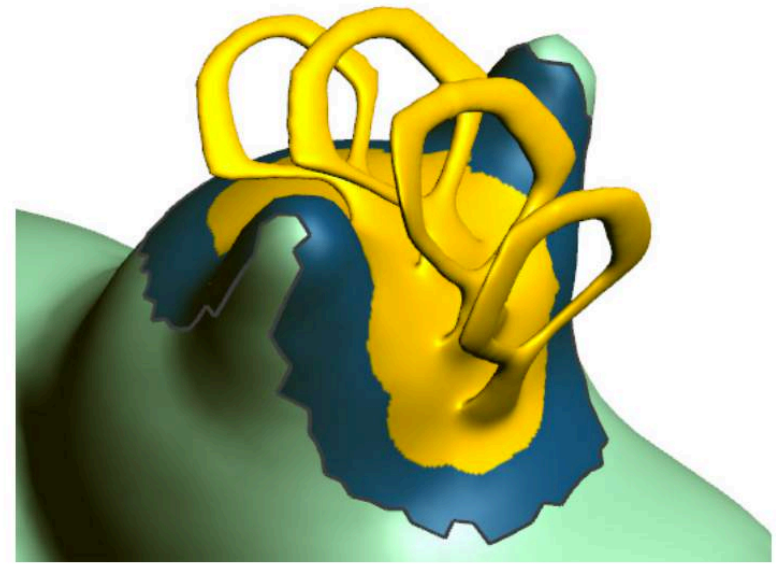


“moving planar projection”

Source canvas parameterization



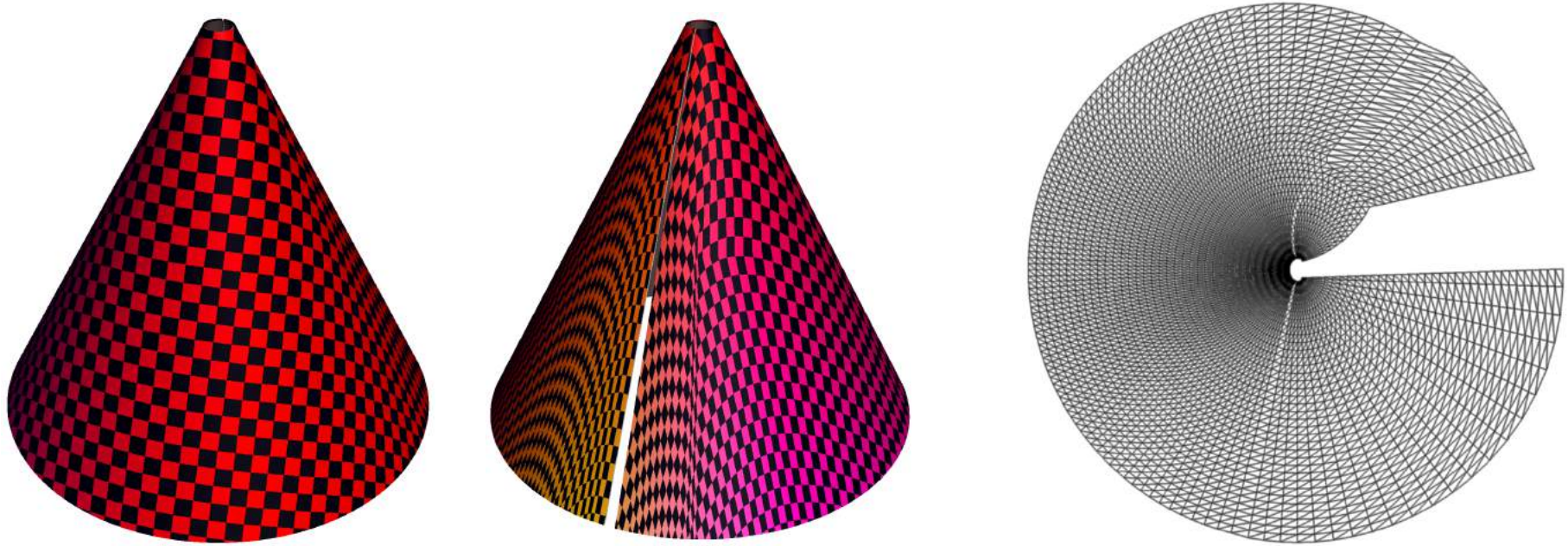
Painted ROI and cloned result



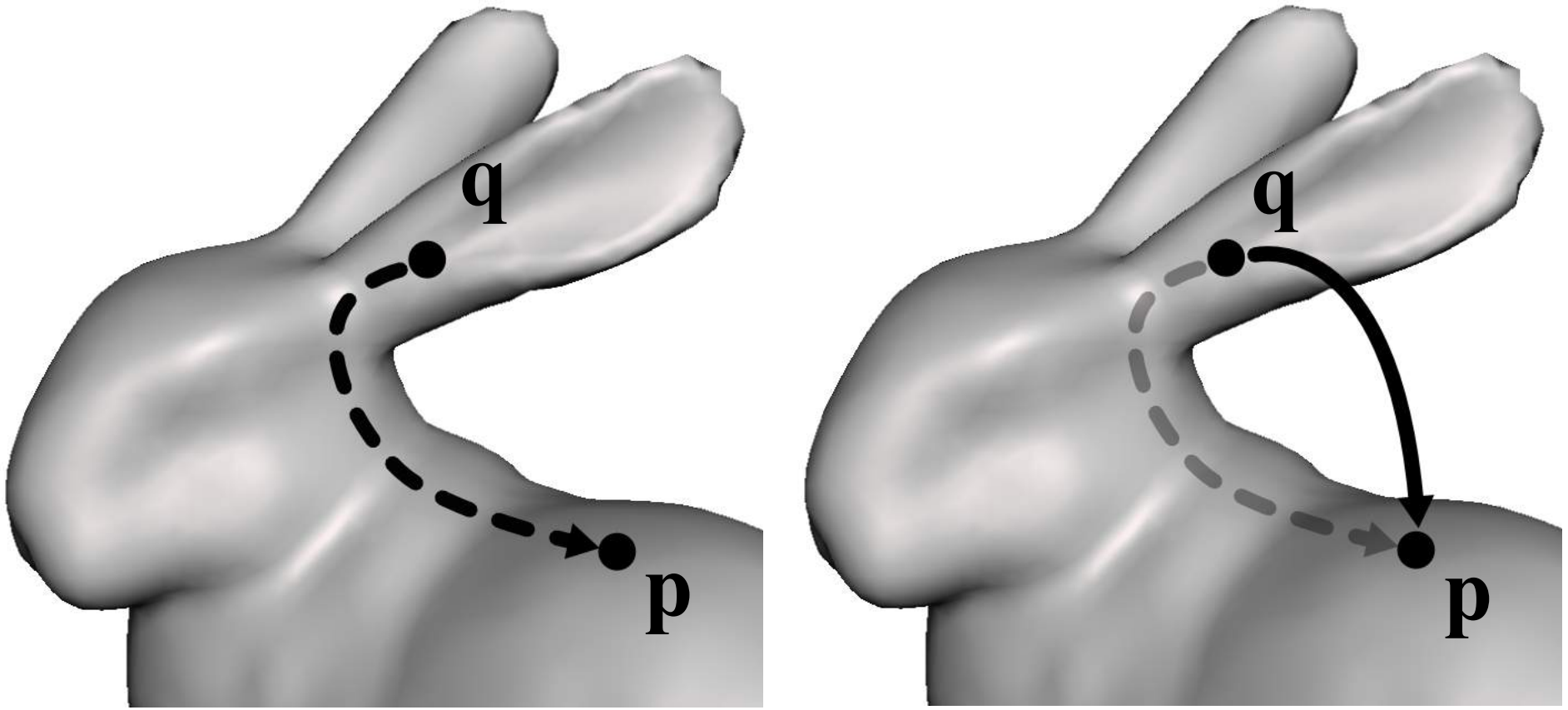
Lots of applications…

- ~100 citations
- Very useful in 3D sculpting/painting
  - Basis for meshmixer
- Local geometry editing
  - Low-distortion 2D domain for stitching/etc
- Local Geodesics
  - Surface signal analysis, descriptors, …

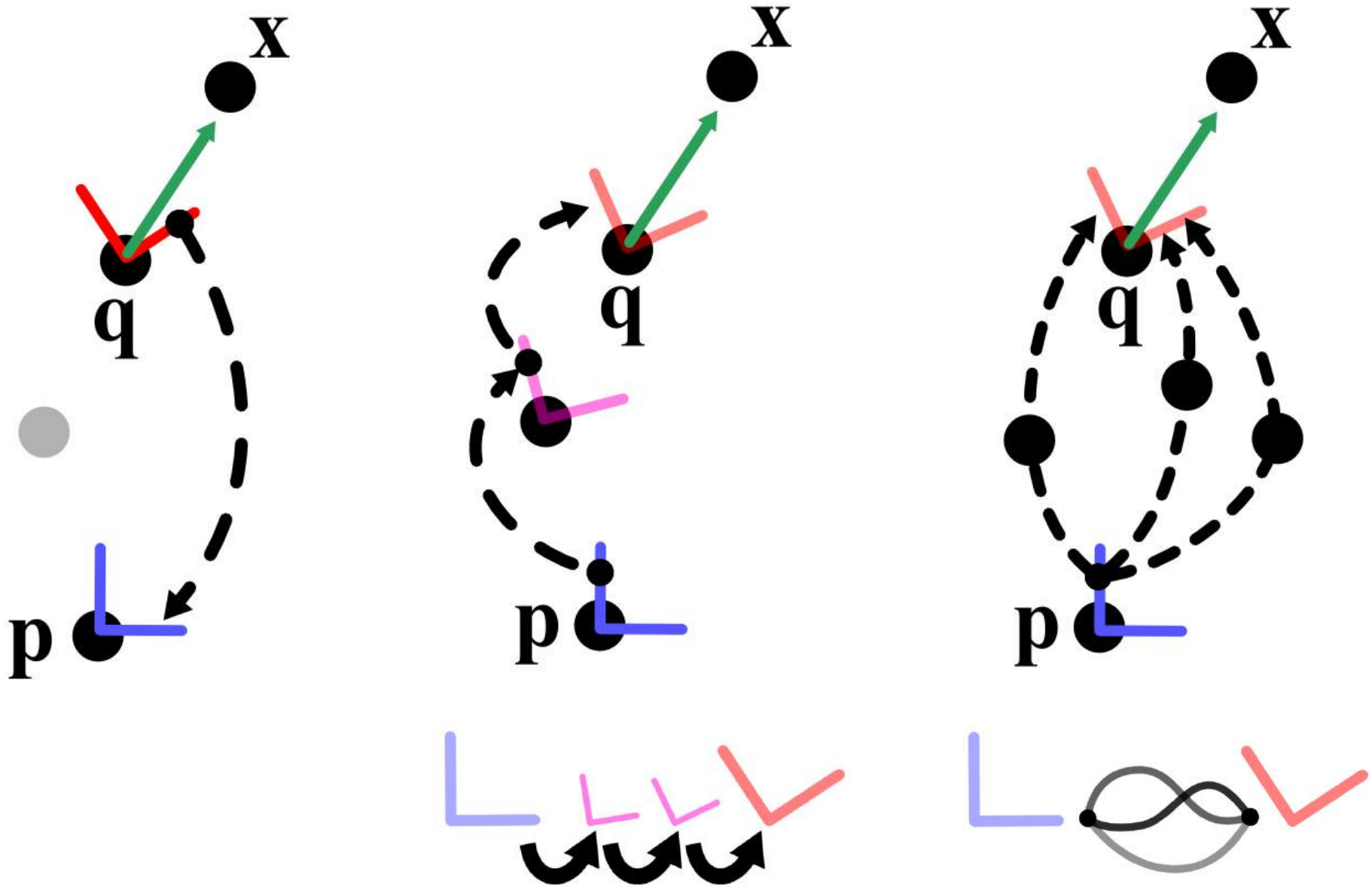
- Improved by Campen et al
  - More robust but more complicated to implement



Parallel Transport



"Apply rotation R that takes frame at $\mathbf{q}$ to frame at $\mathbf{p}$"

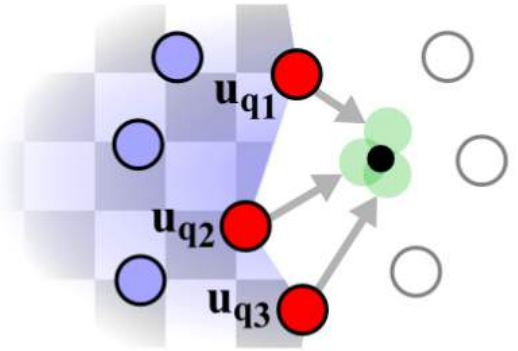

x
q
p
x
q
p
x
q
p

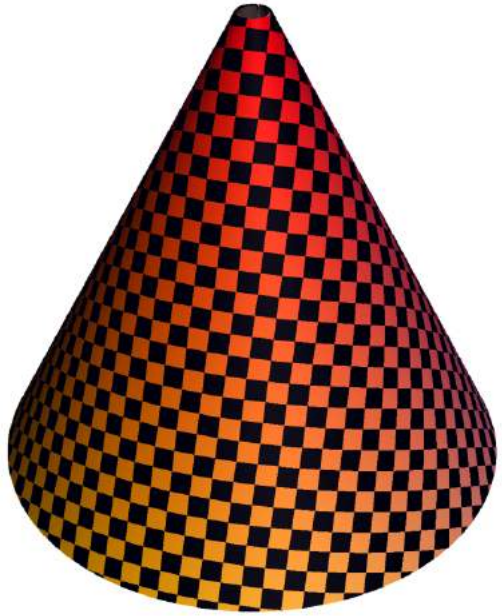
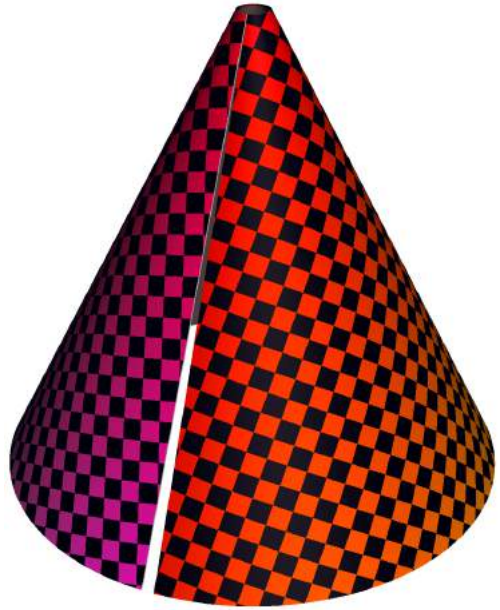
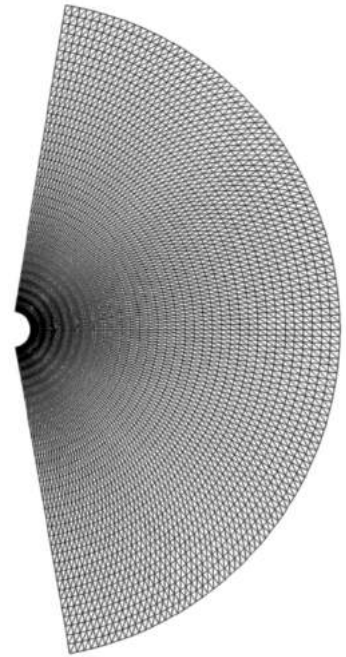
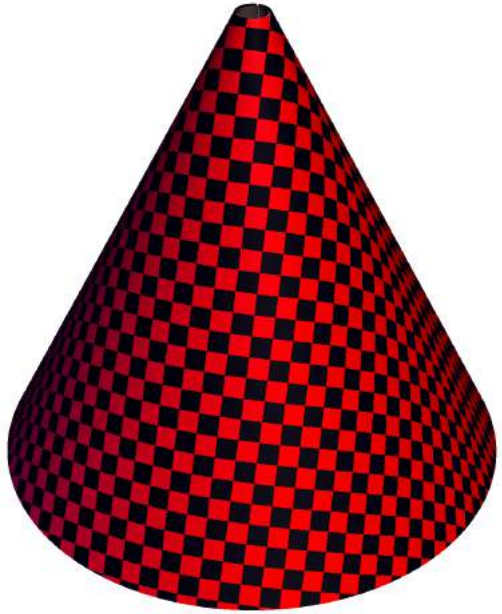
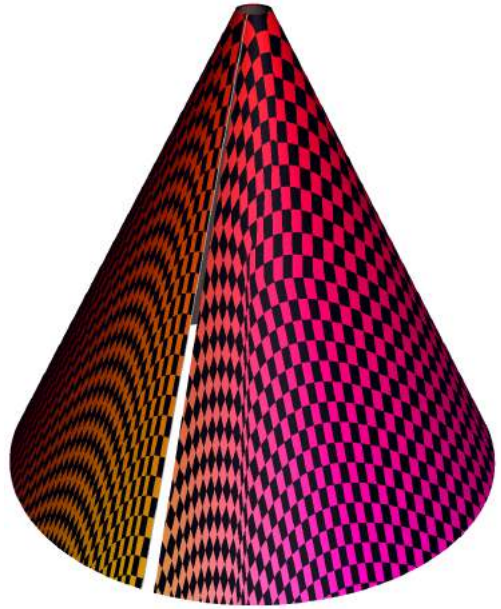
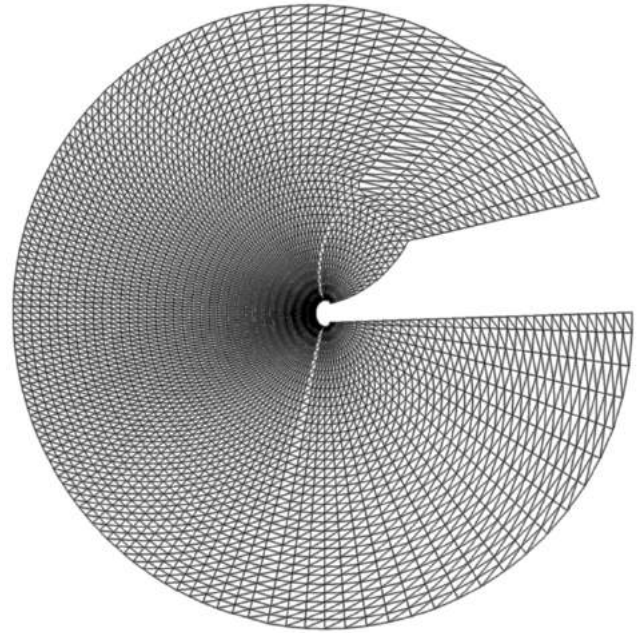
Frame Propagation

- Frame is $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{n})$
- To compute frame at $\mathbf{x}$:

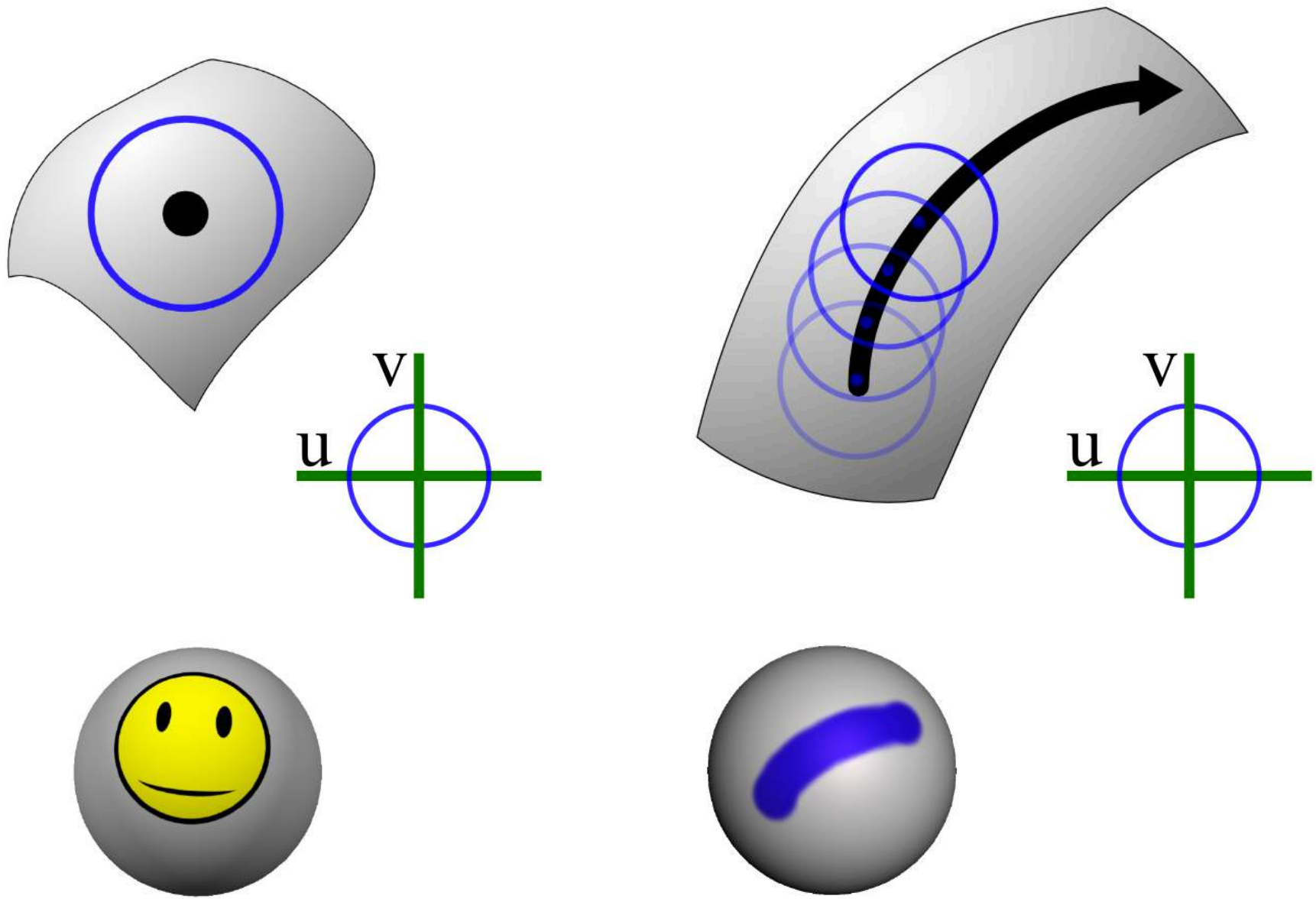
$$\mathbf{e}_1(\mathbf{x}) = \sum_{q_i \in N_u(x)} w(\mathbf{q}_i, \mathbf{x}) R(\mathbf{q}_i, \mathbf{x}) \mathbf{e}_1(\mathbf{q}_i)$$

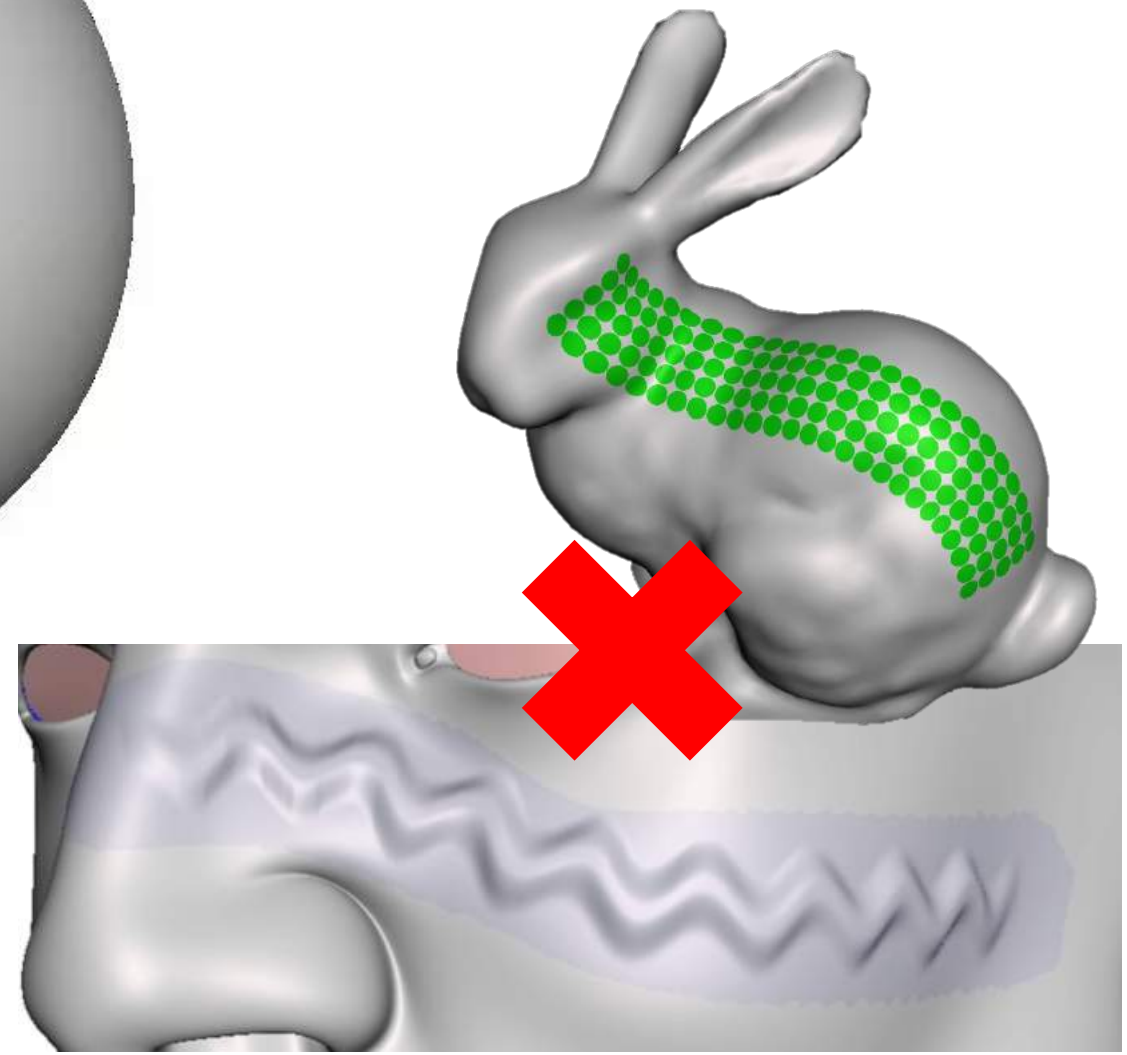
- R is rotation around ($\mathbf{n_q}$ x $\mathbf{n_x}$) that takes $\mathbf{n_q}$ to $\mathbf{n_x}$
  - Levi-Civita Connection
- w is inverse-distance weight
- Find $\mathbf{e_2(x)}$ by orthogonality

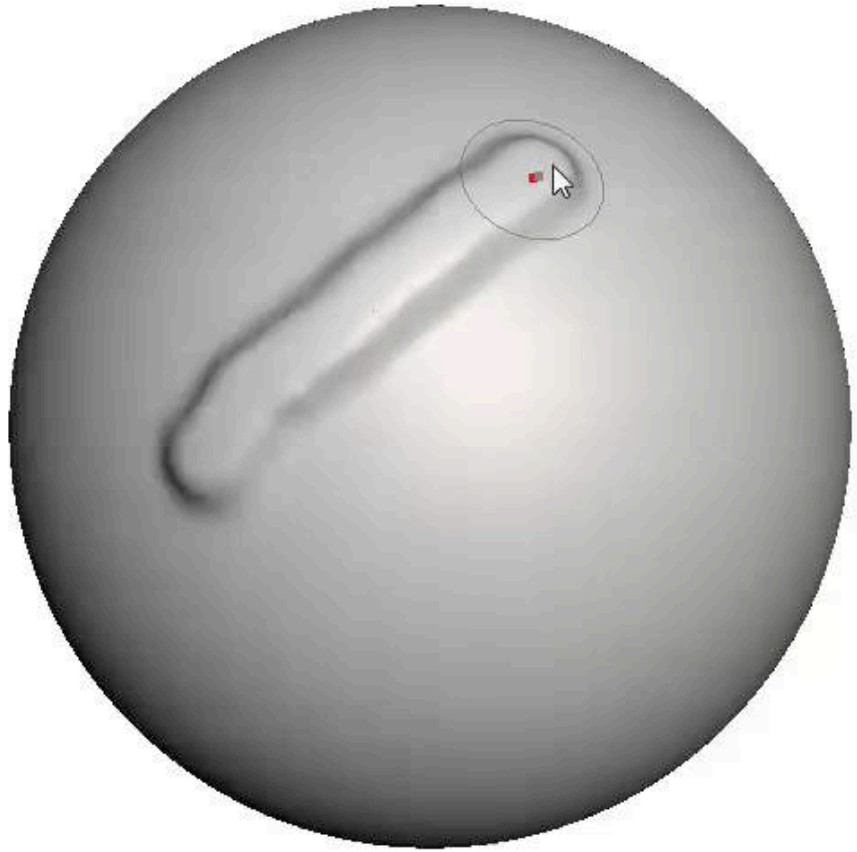


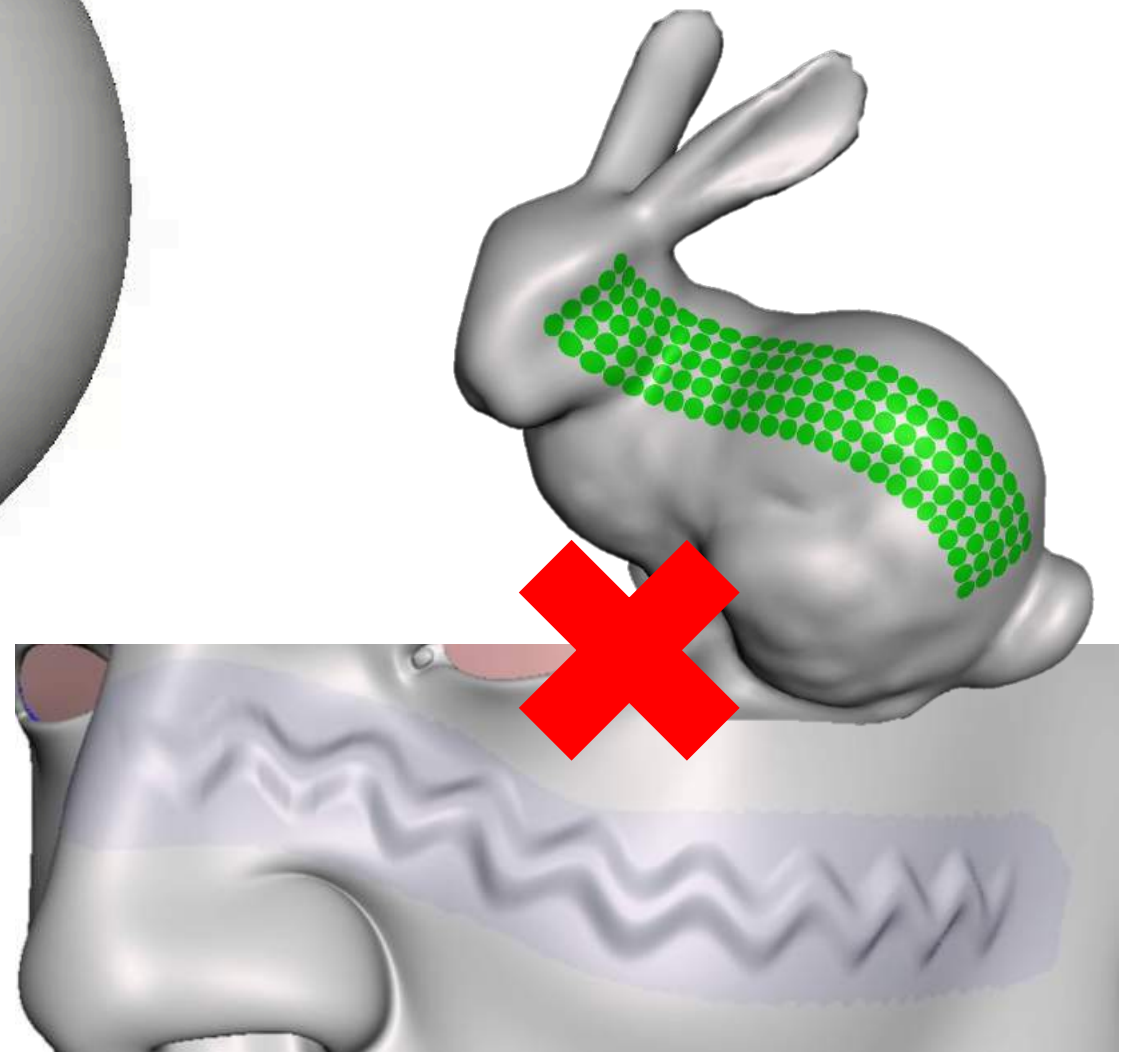







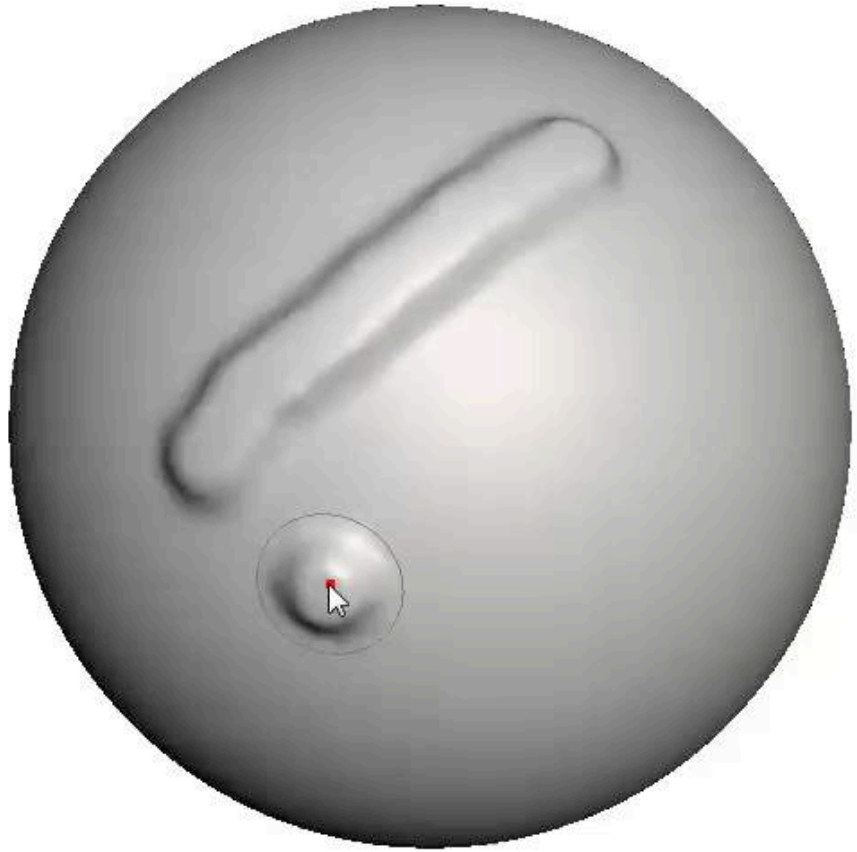
Stroke Parameterization

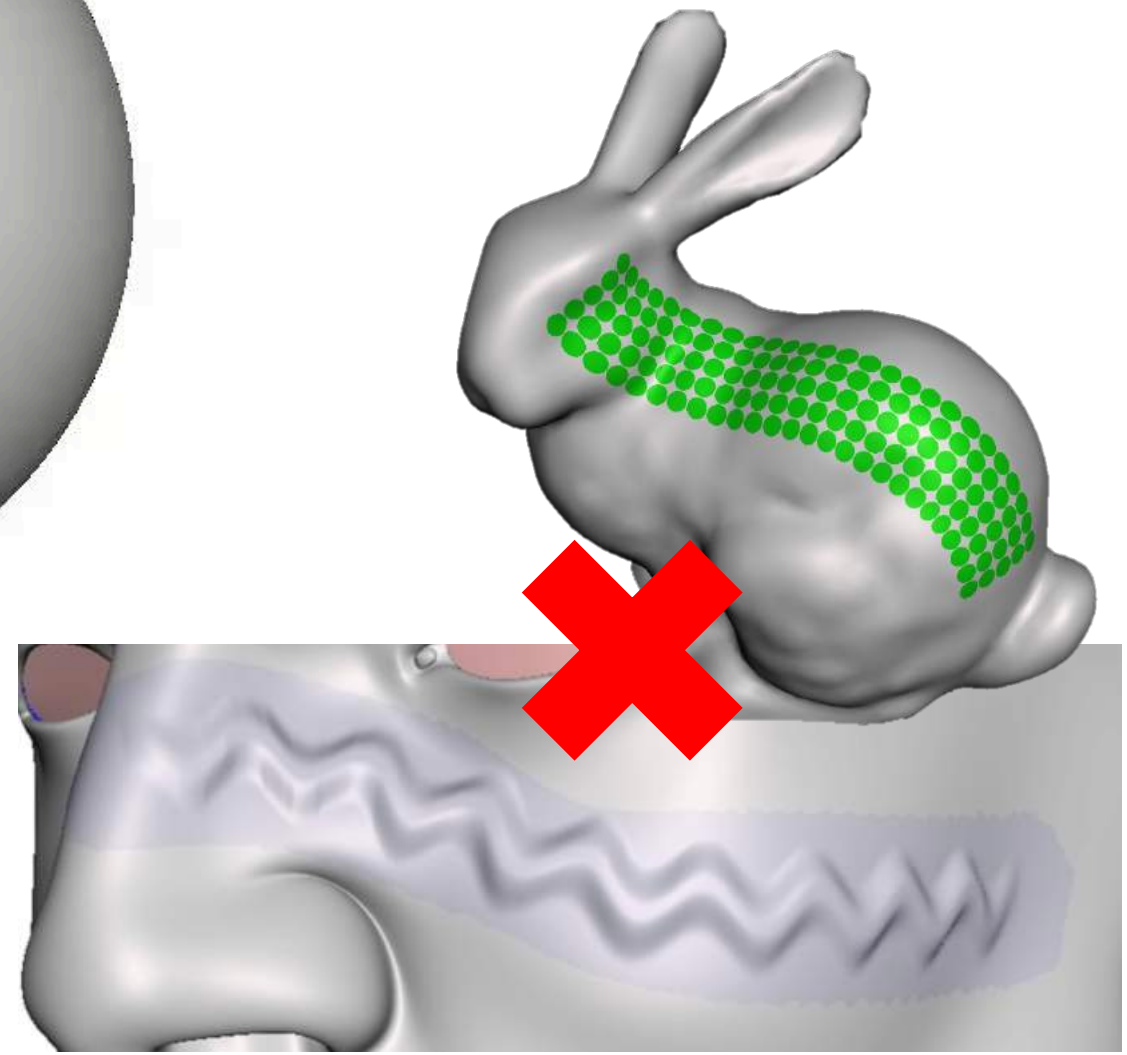

v
u
v
u

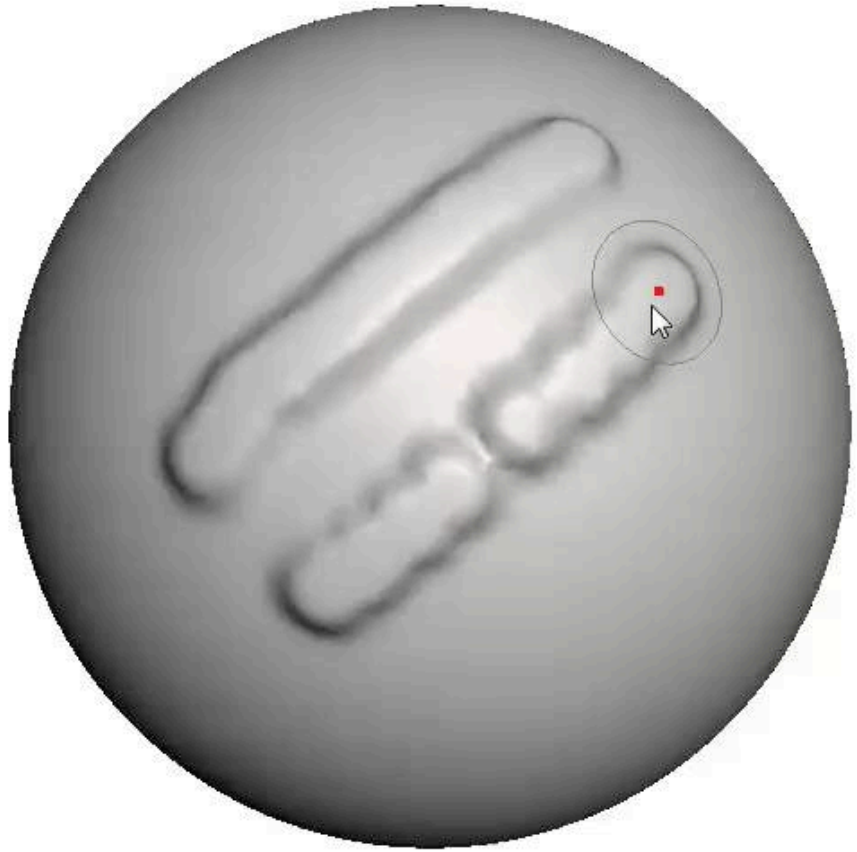


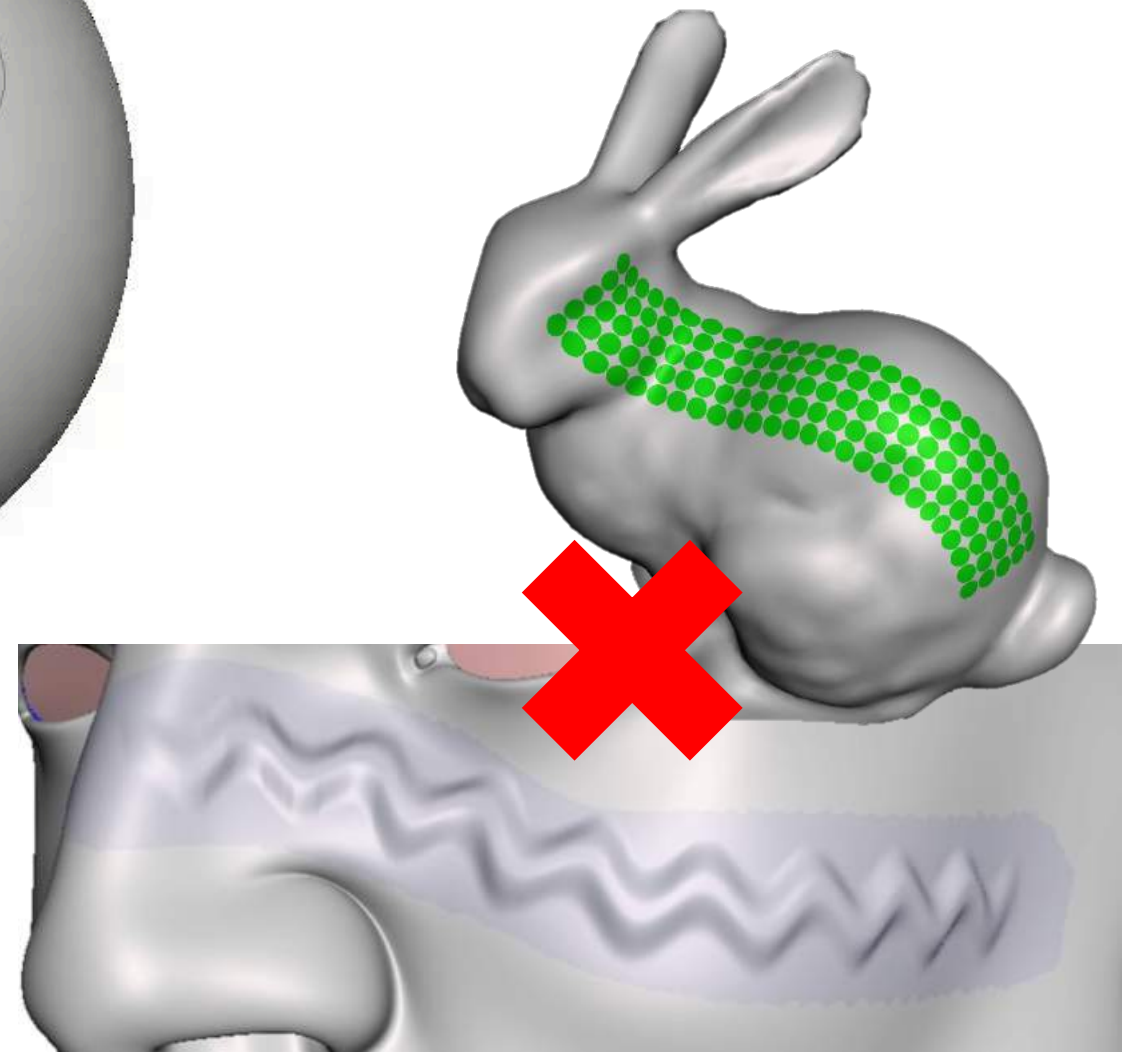


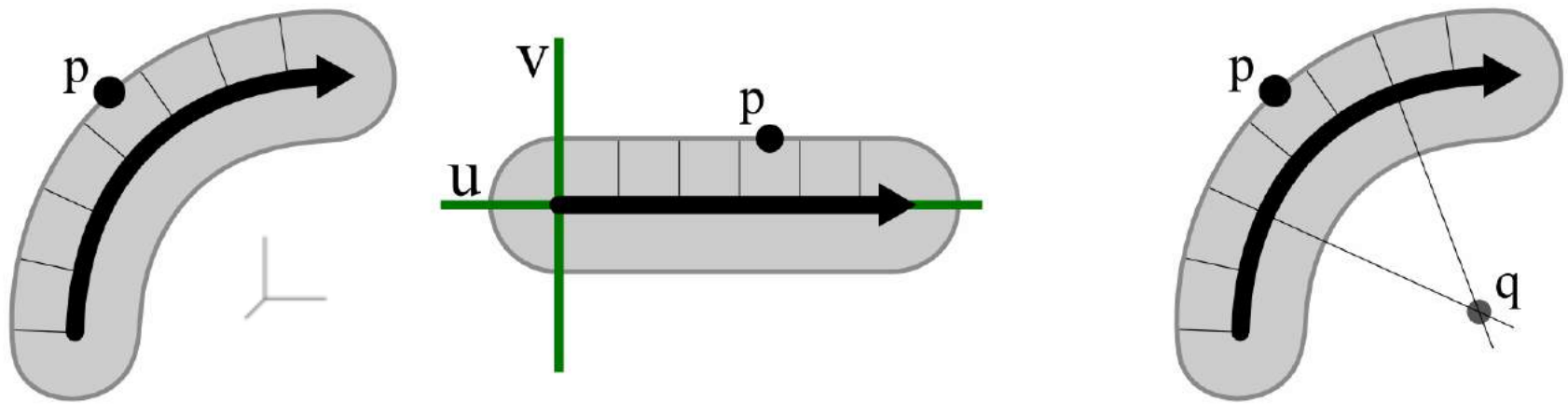


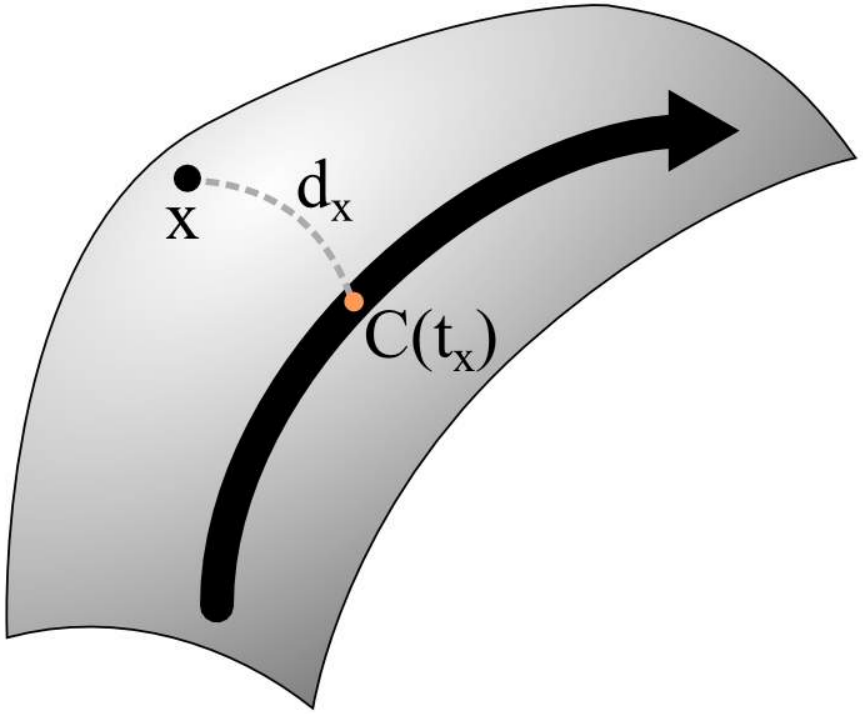


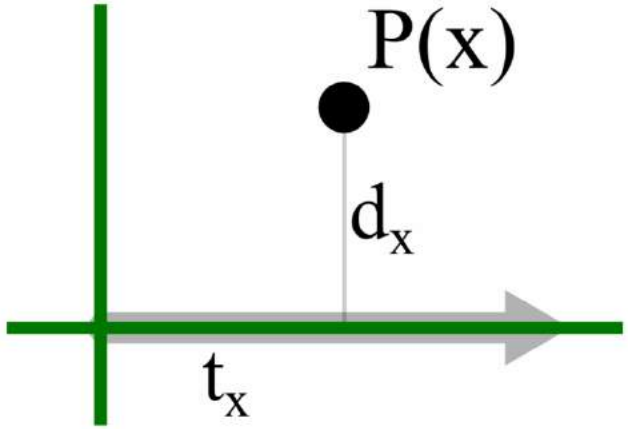




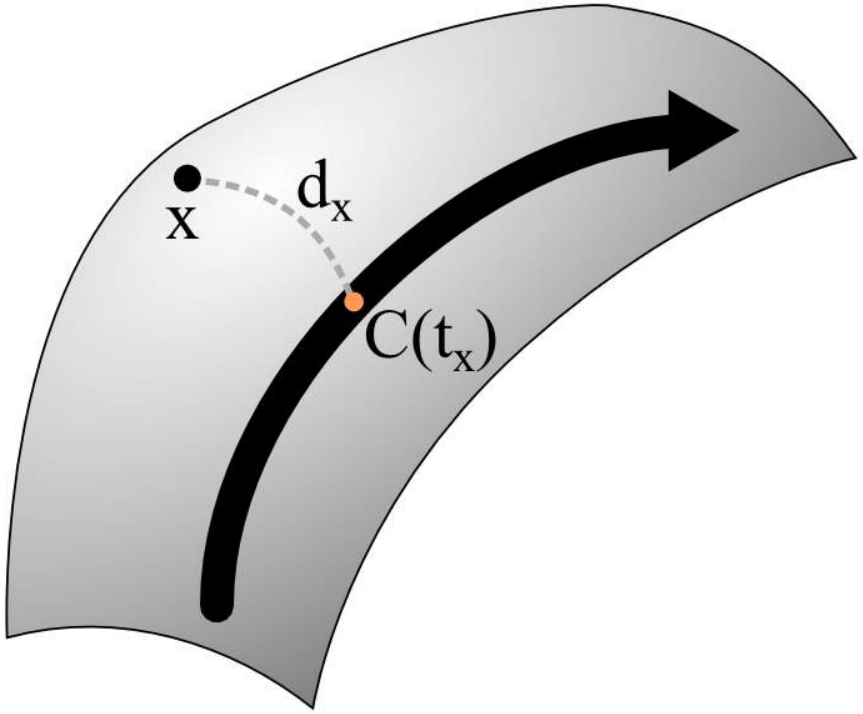


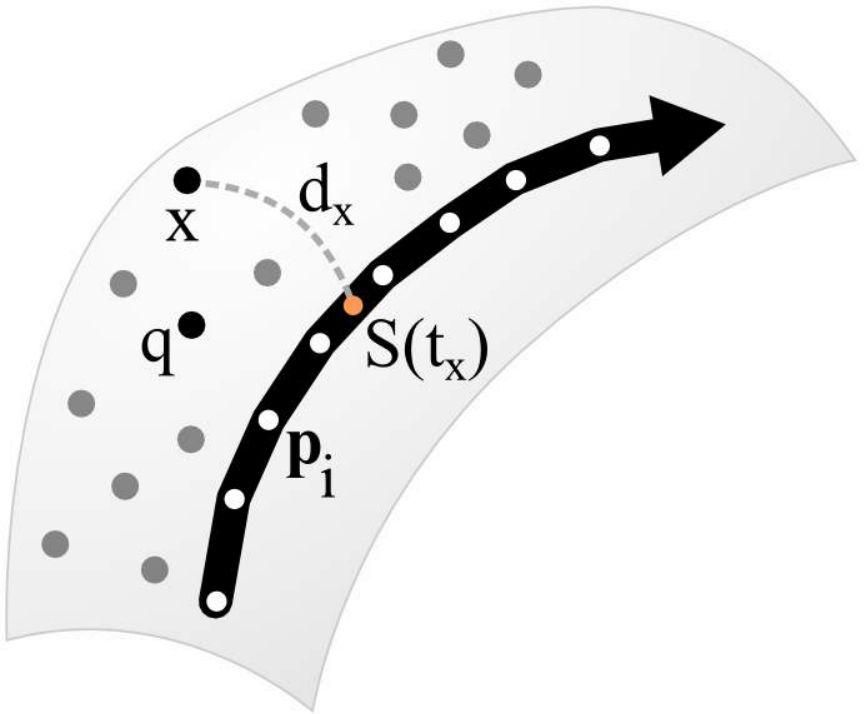

p
v
p
u
p
q



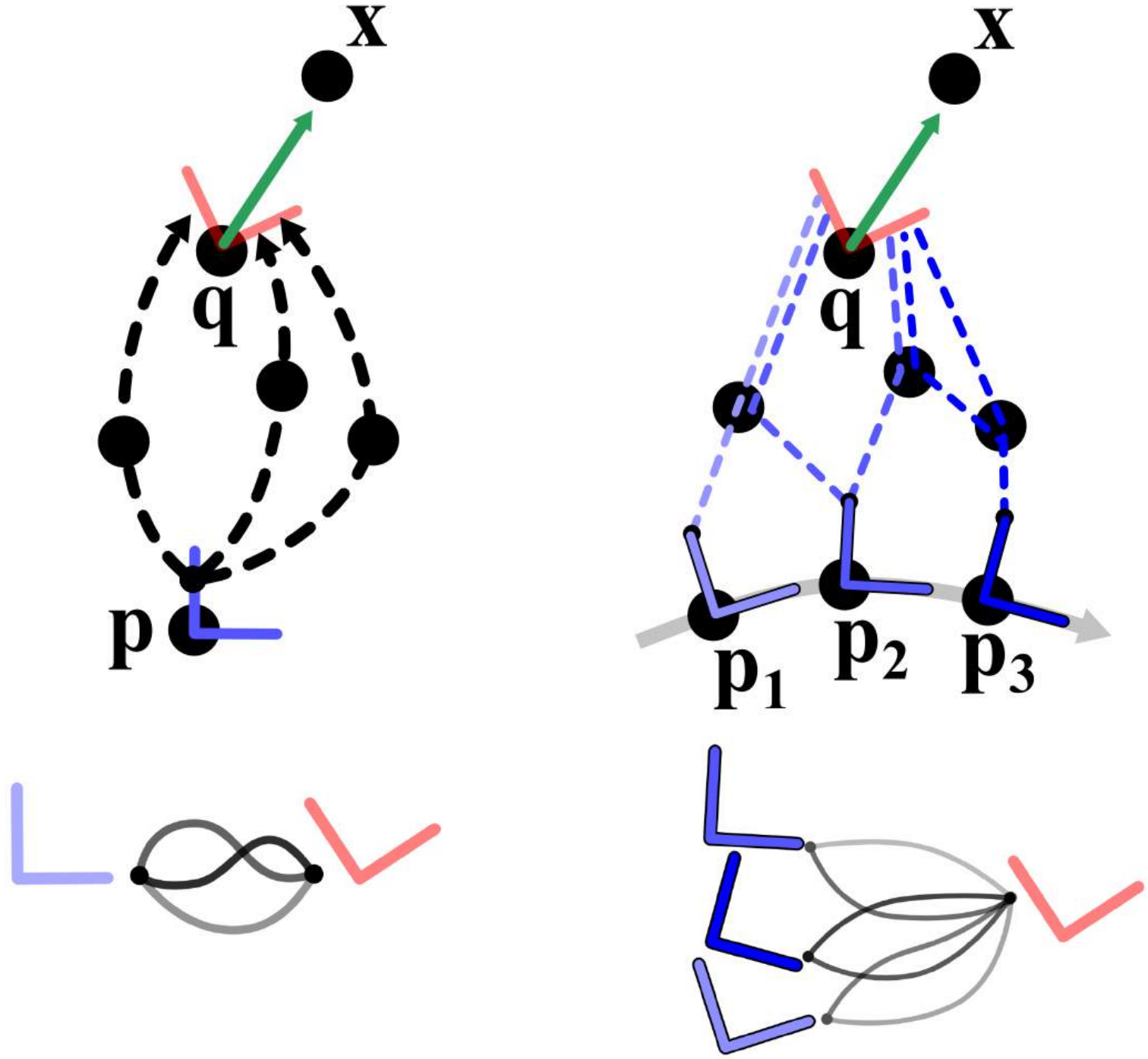


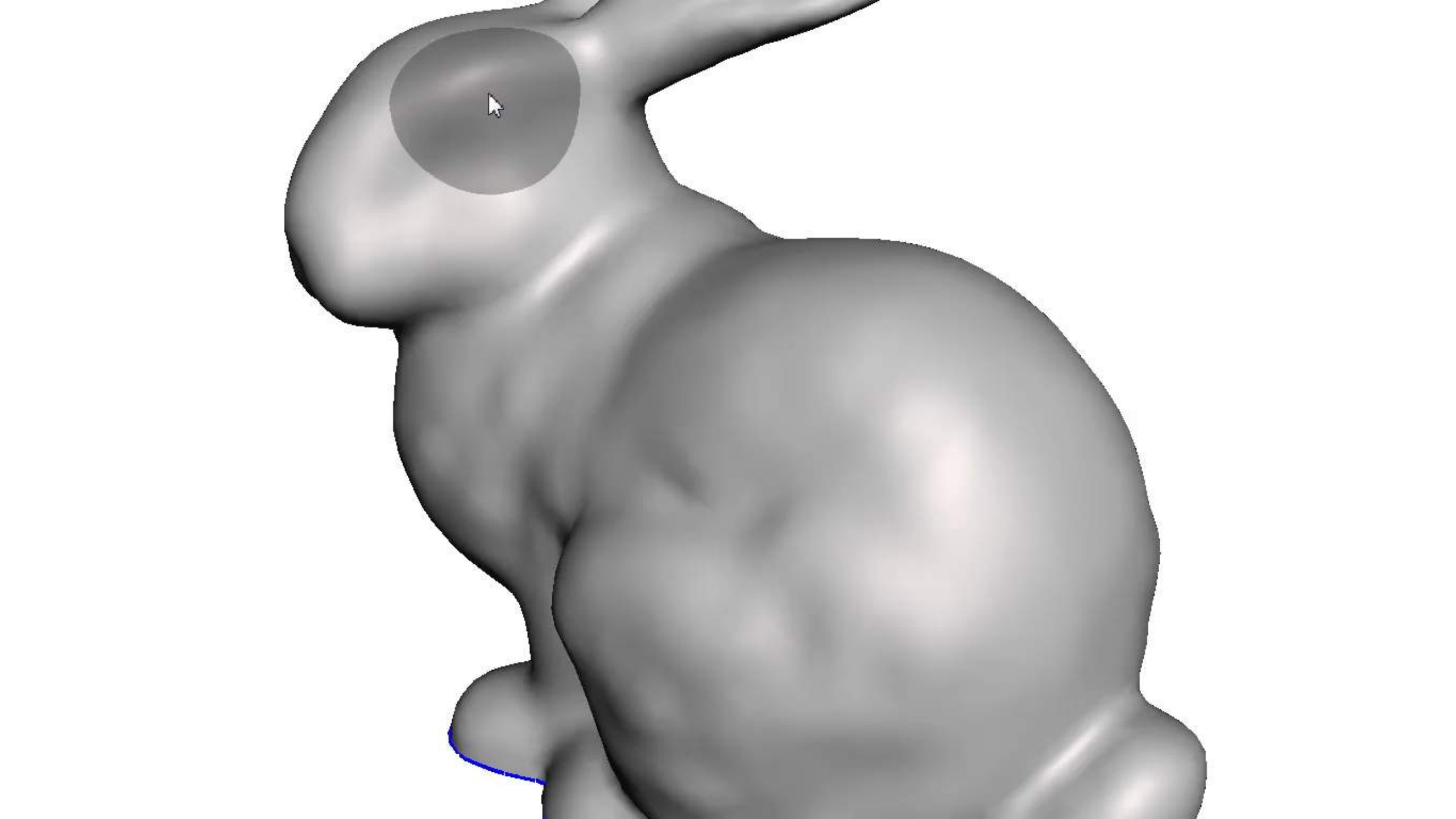
d_x is a ***signed*** distance (see paper)

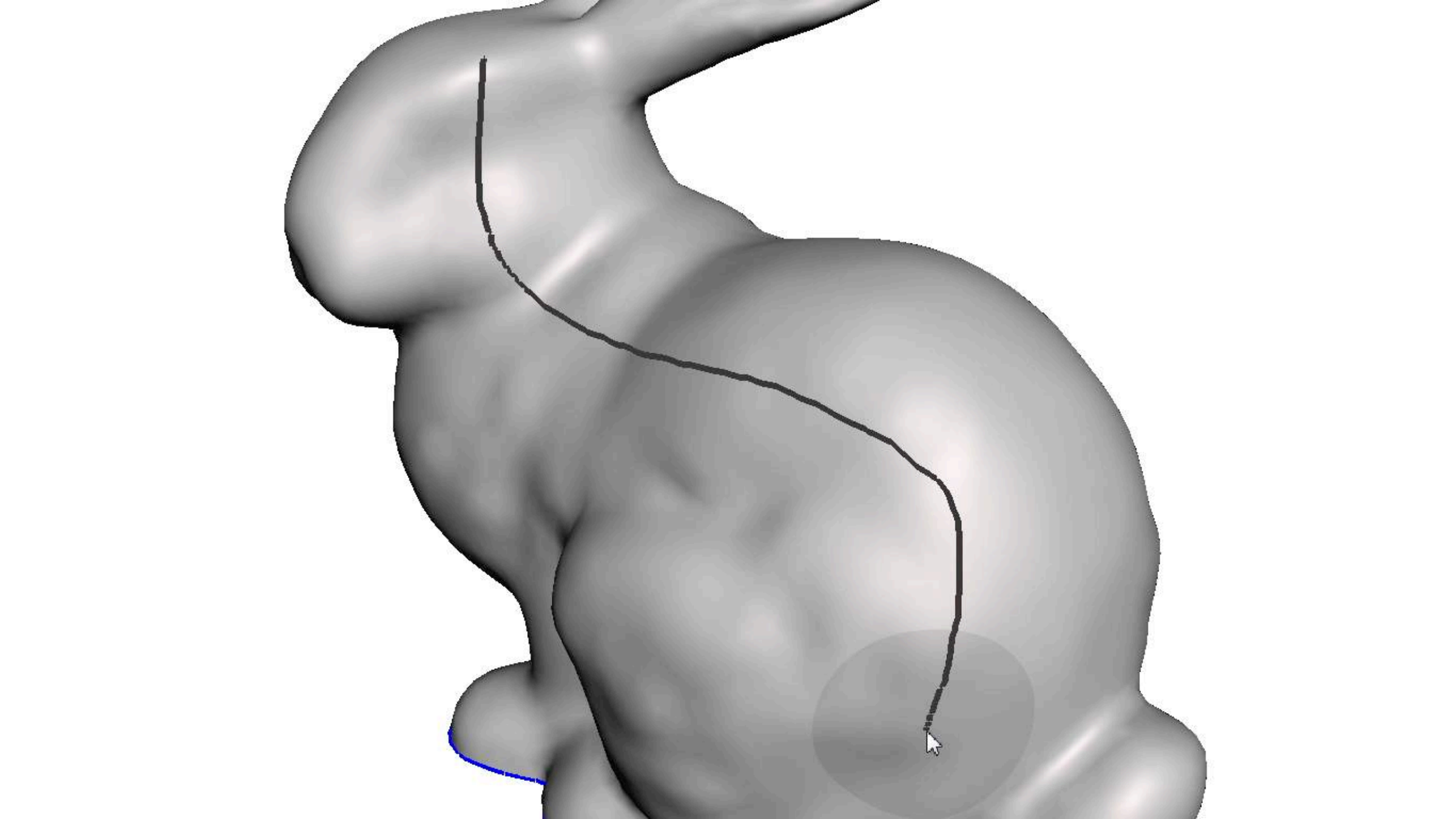


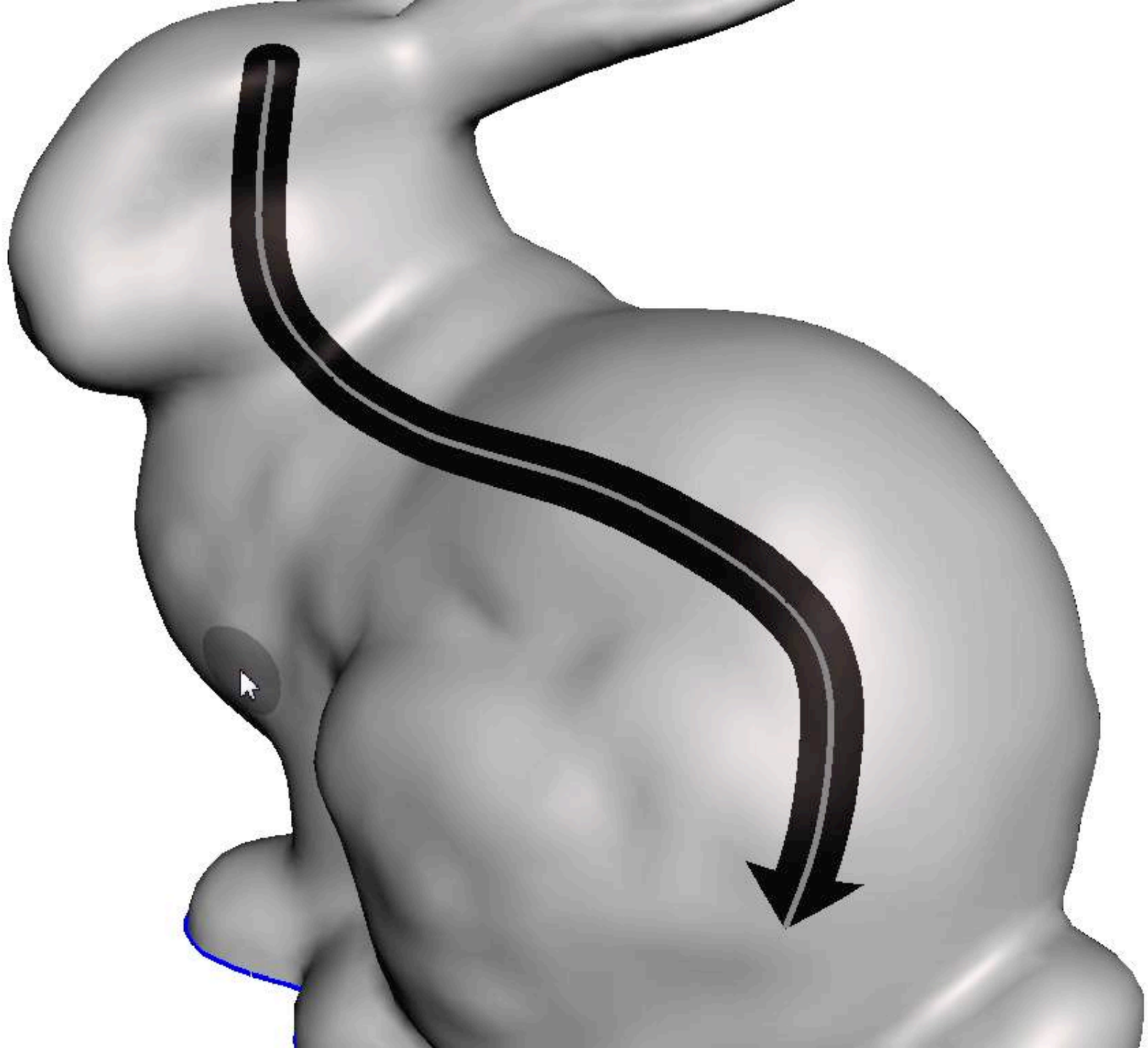


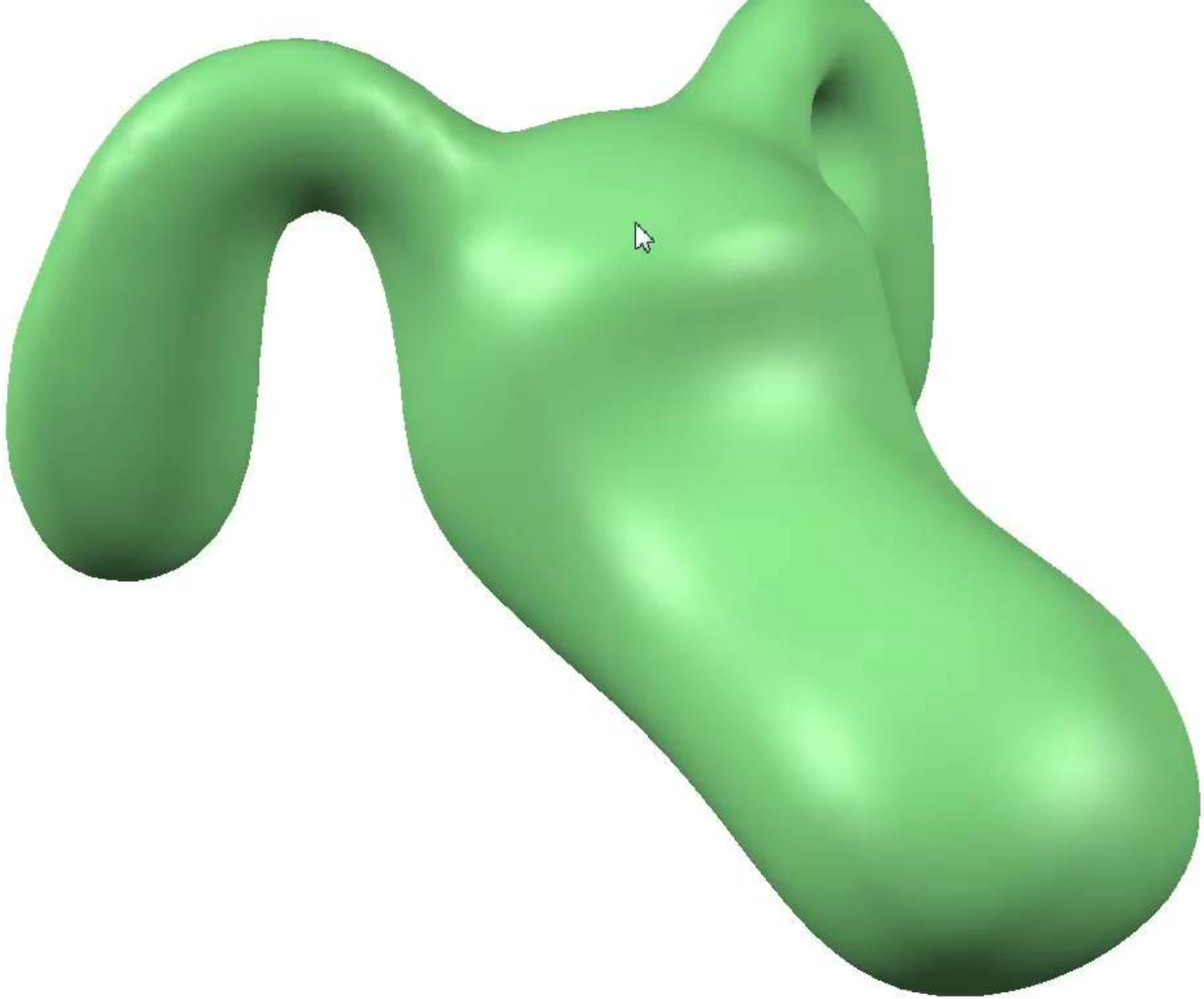
Frames at $\mathbf{p}_i$ are Darboux frames (S', S' x $\mathbf{n}$, $\mathbf{n}$)

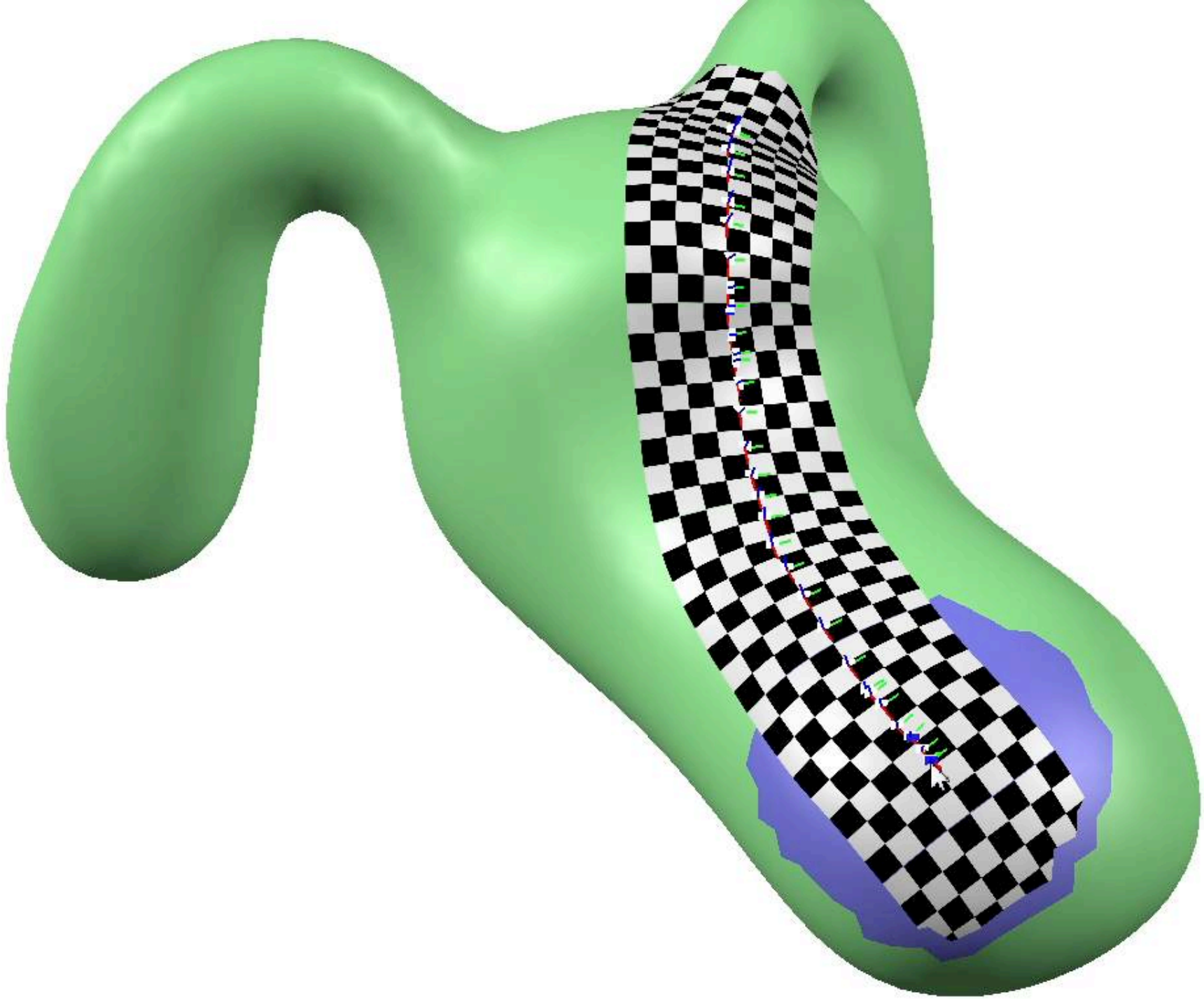

x
q
p
x
q
p1
p2
p3

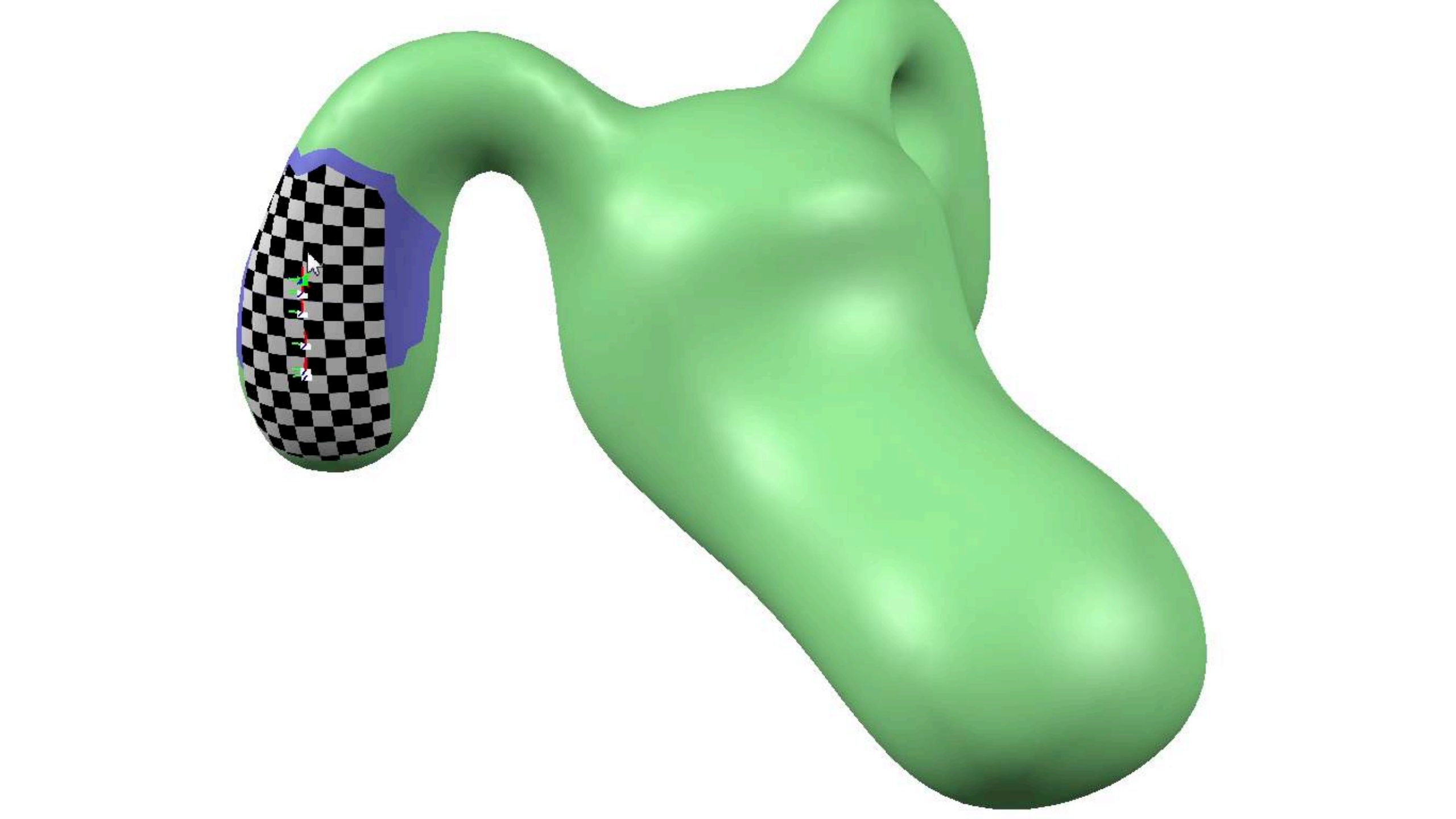


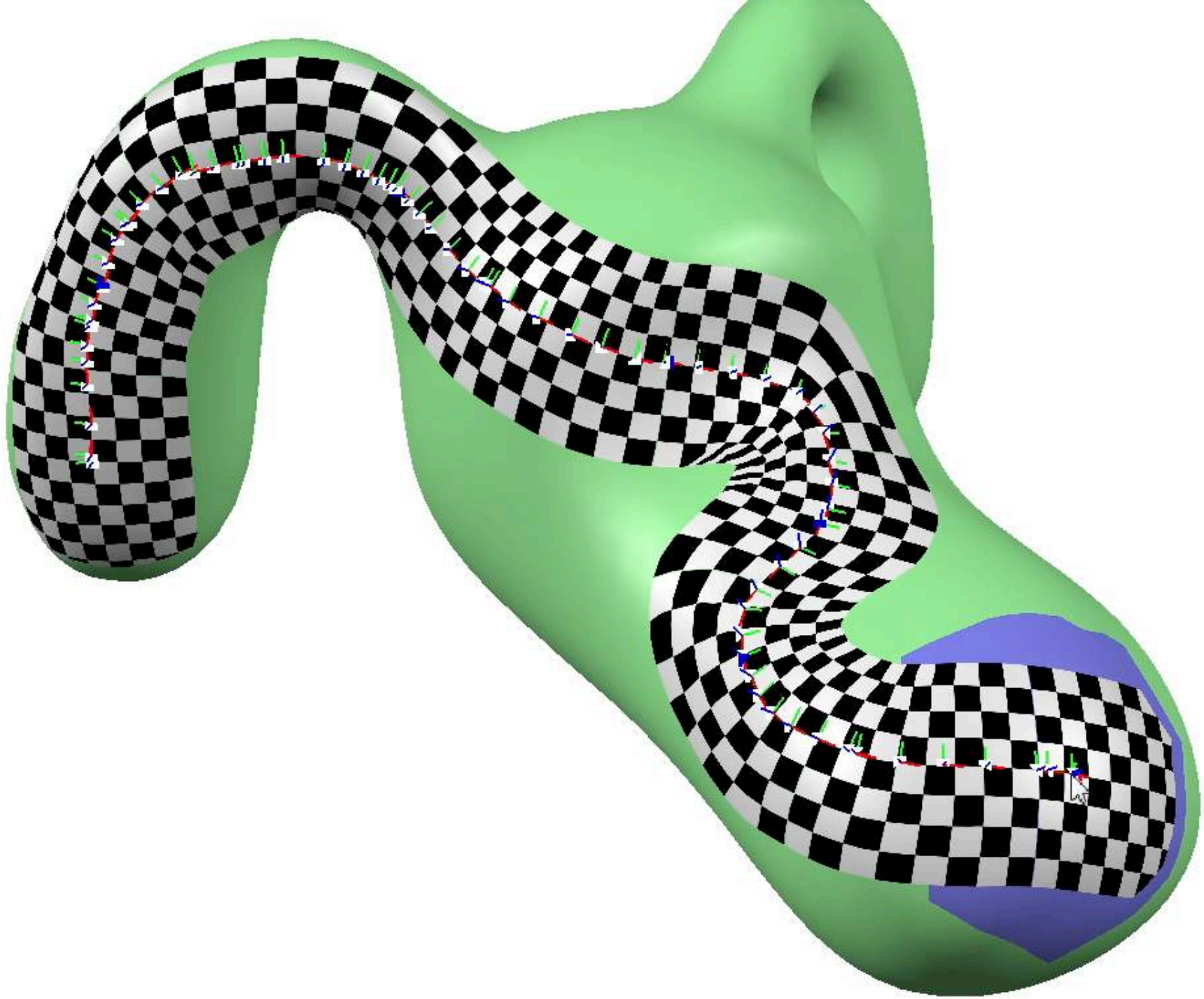


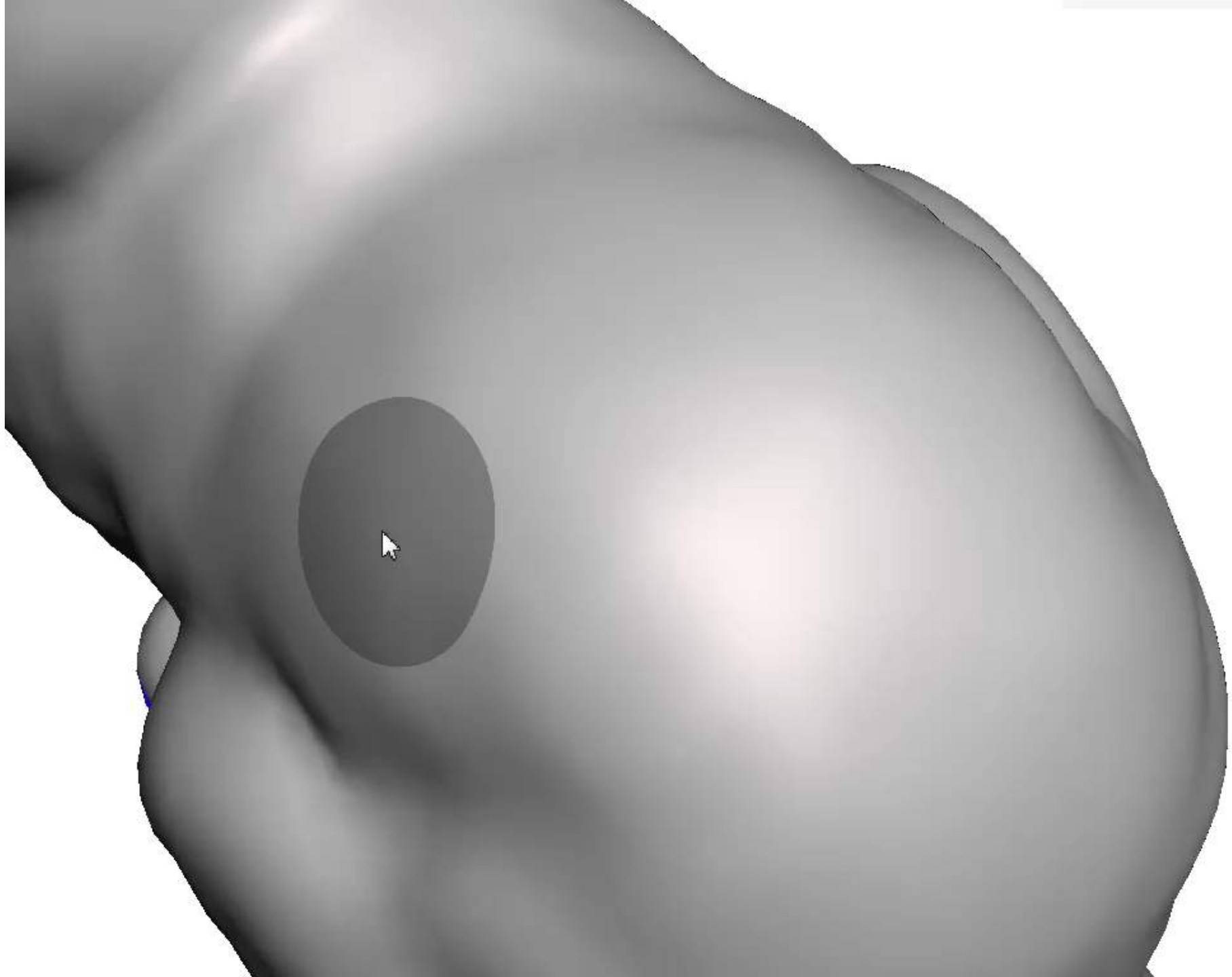


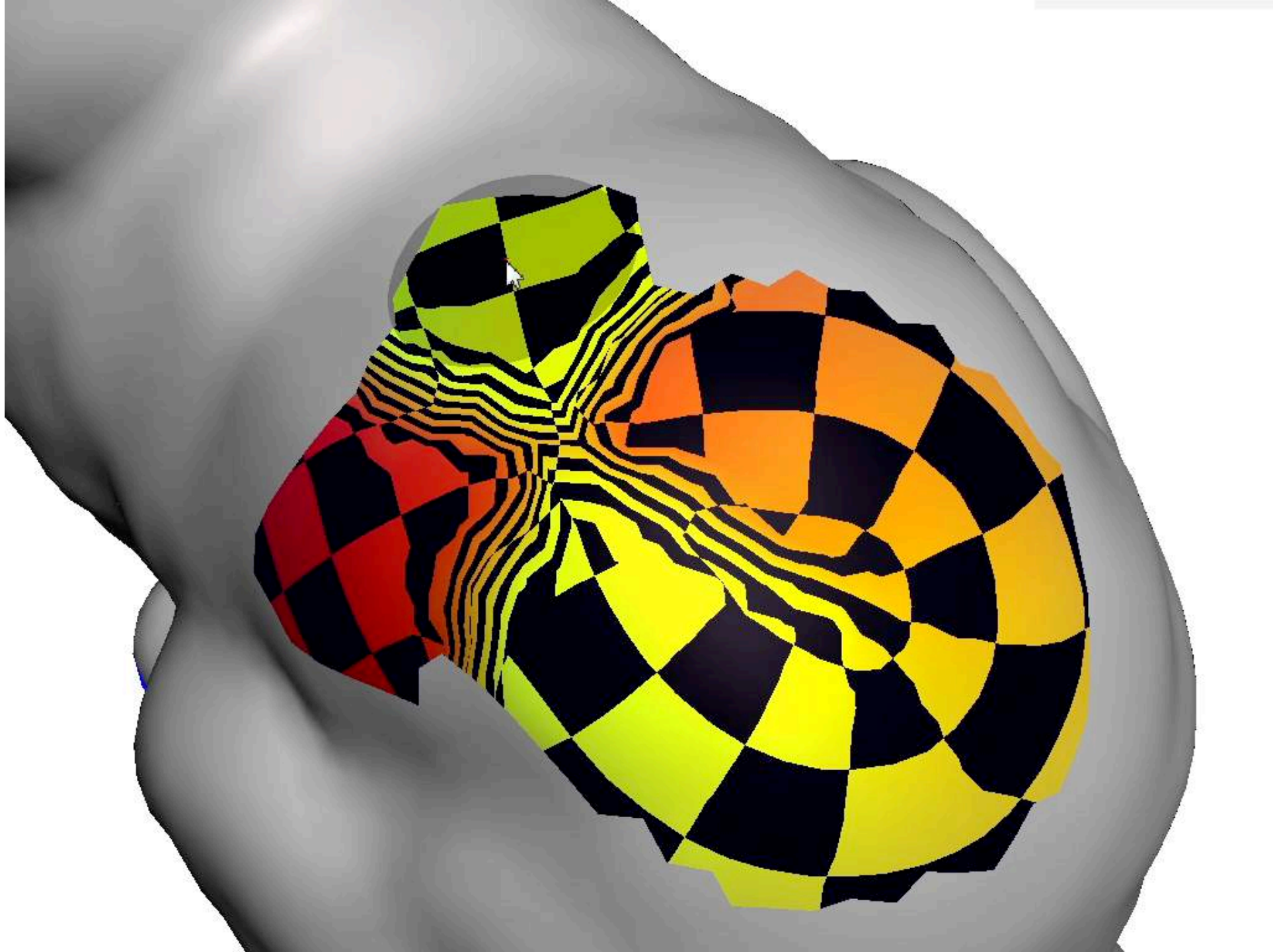


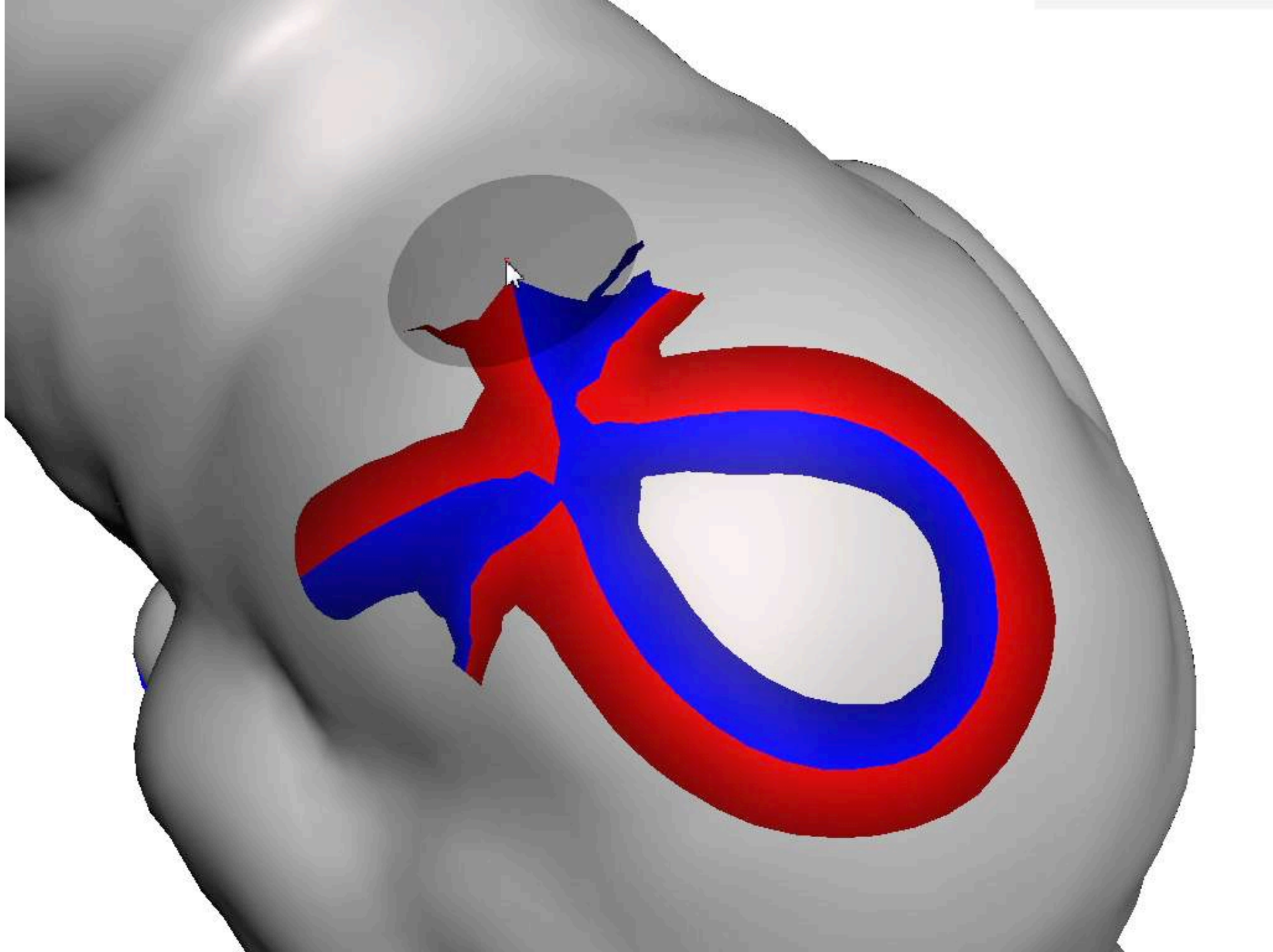


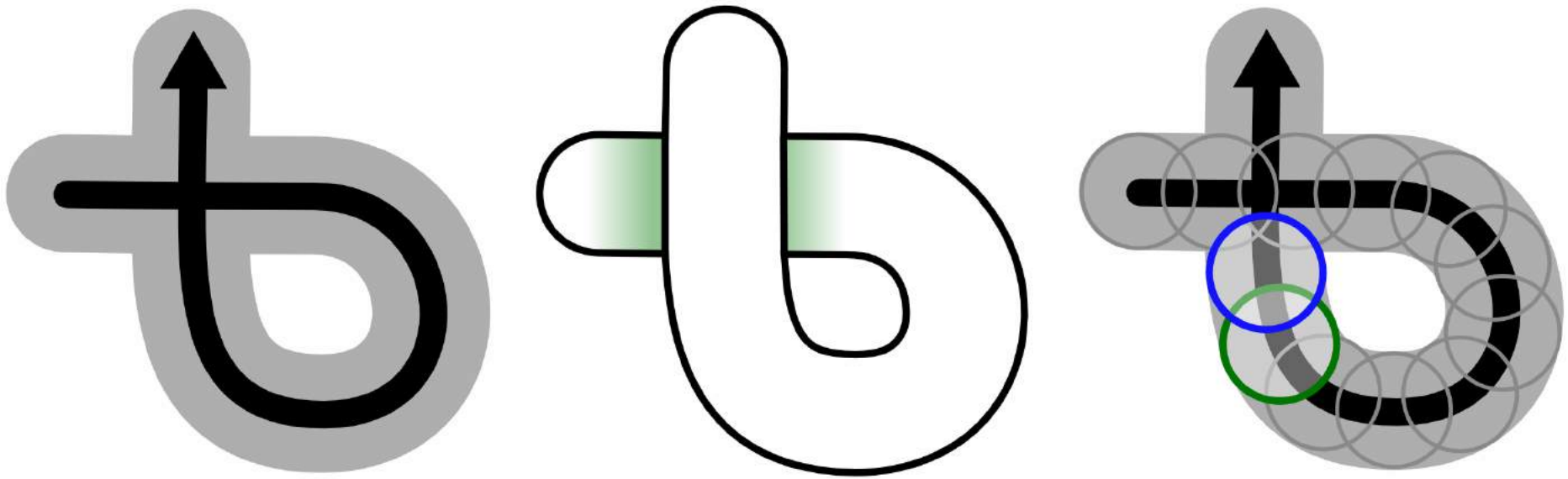


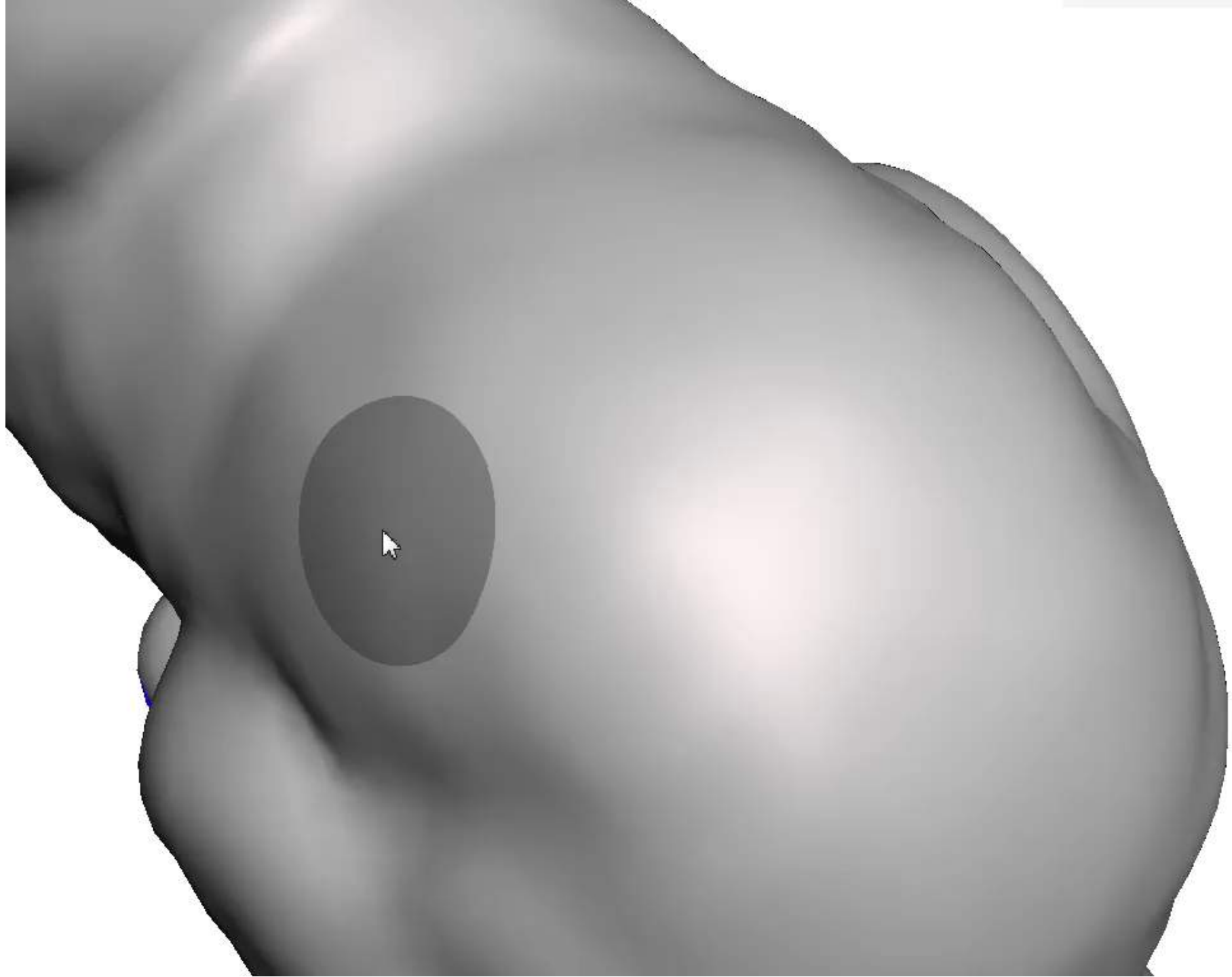


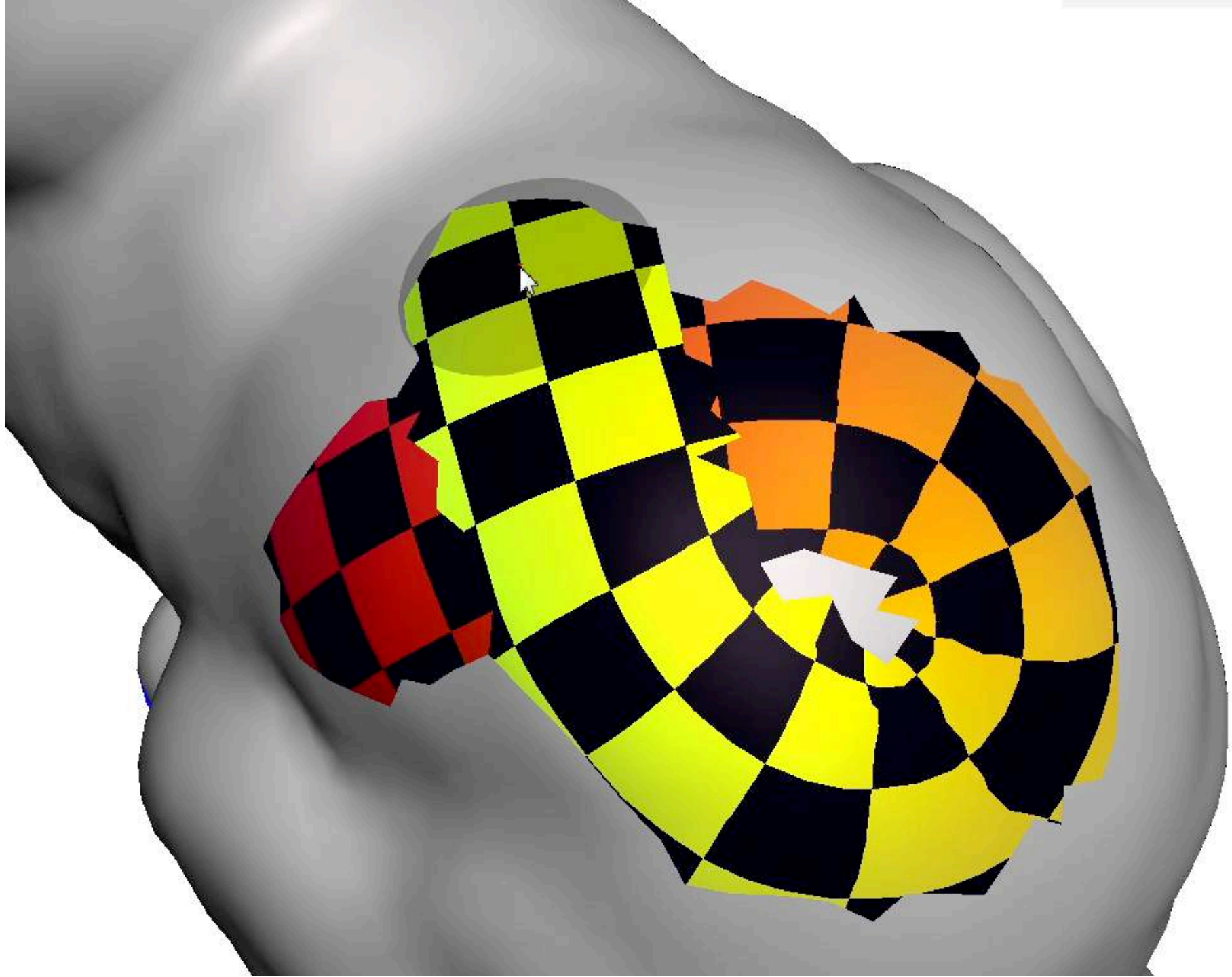




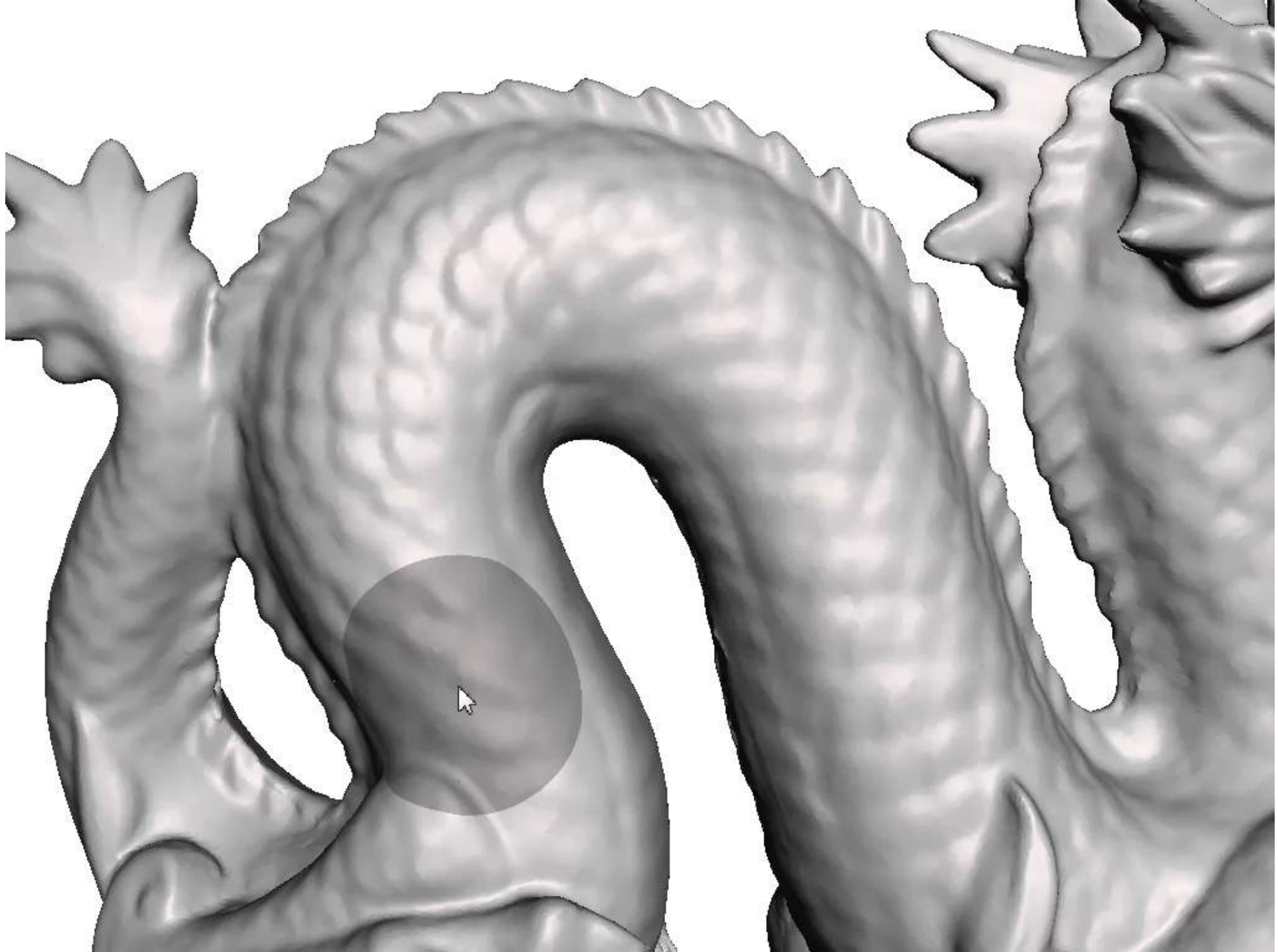


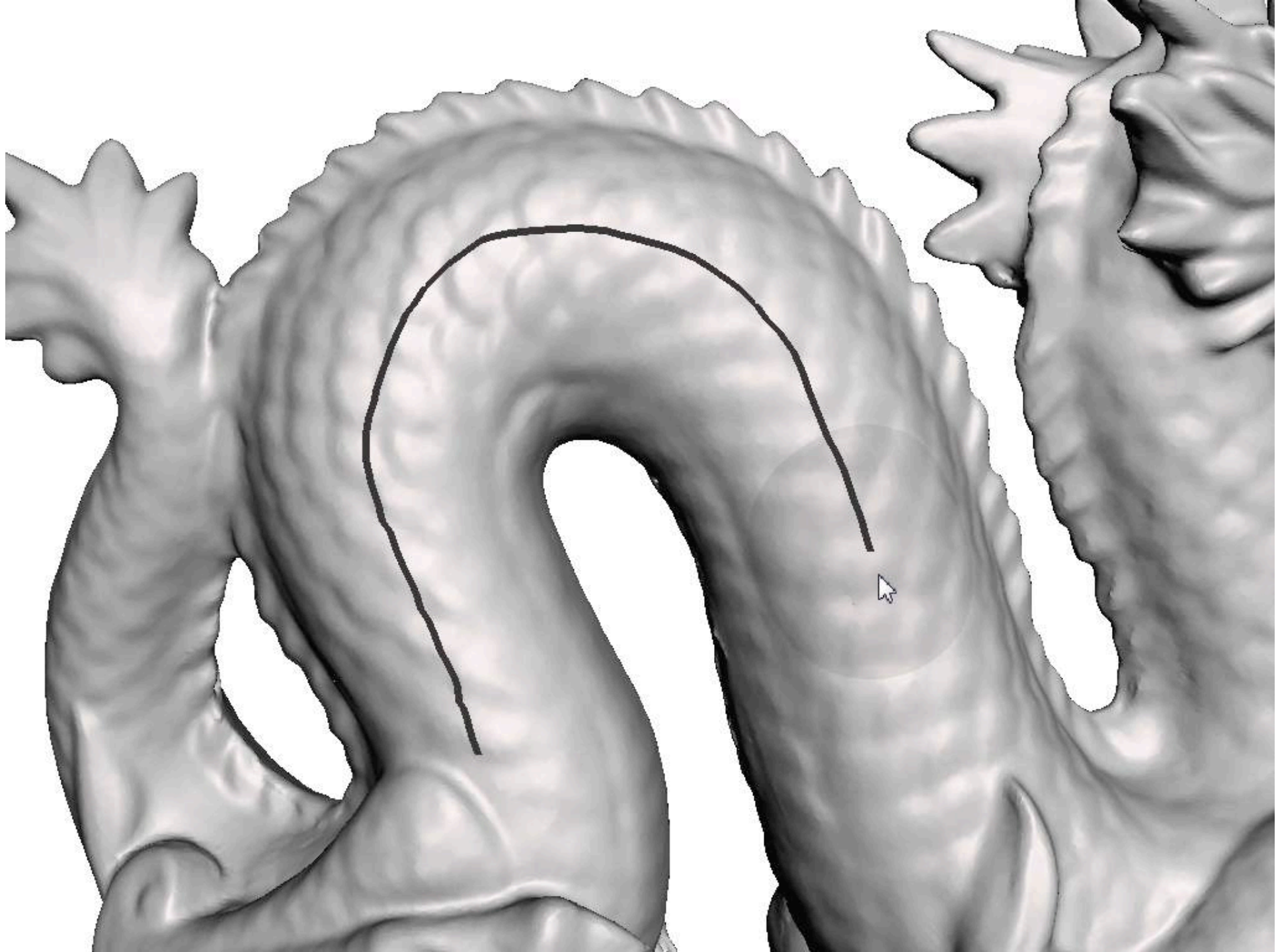


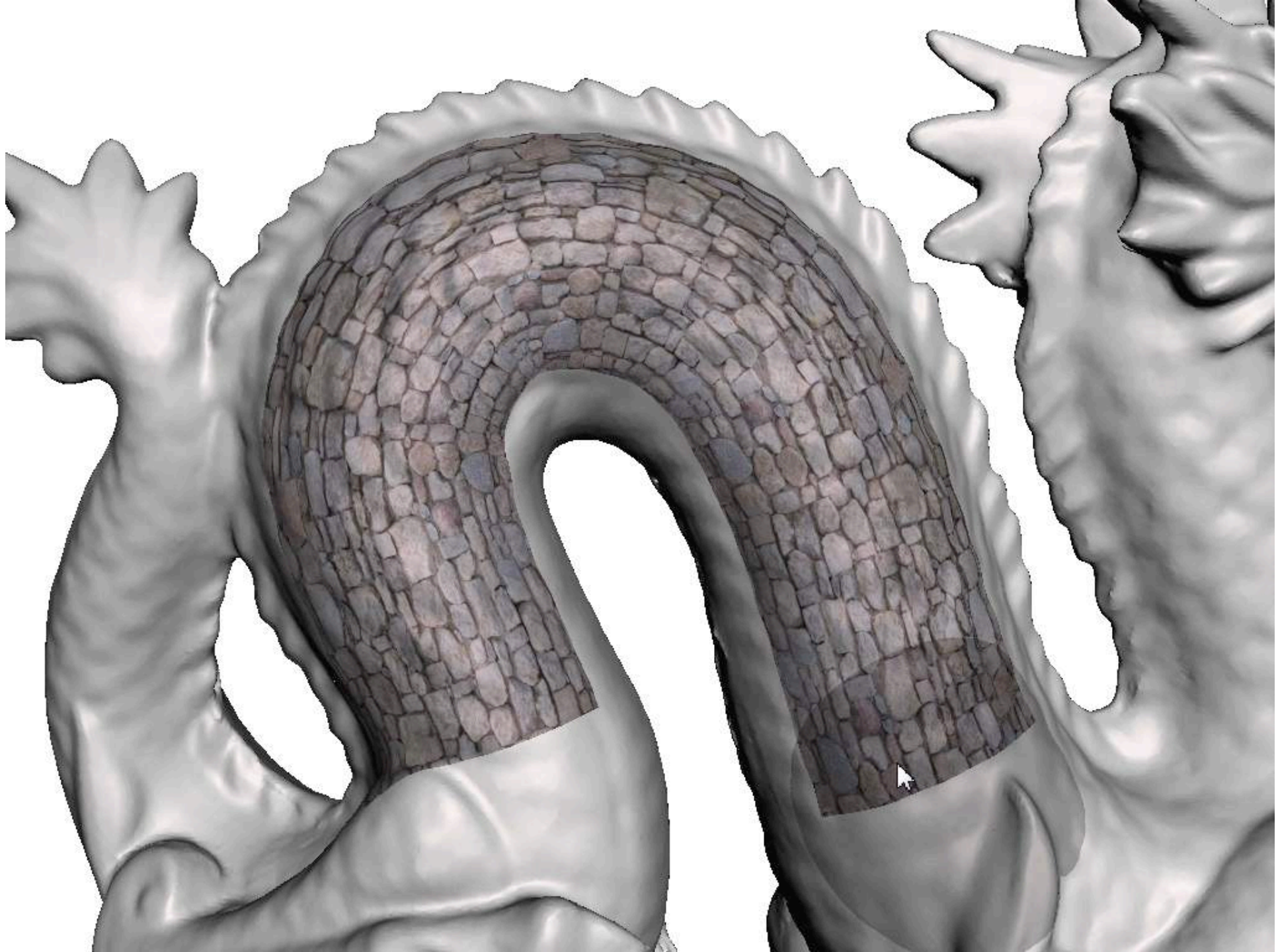


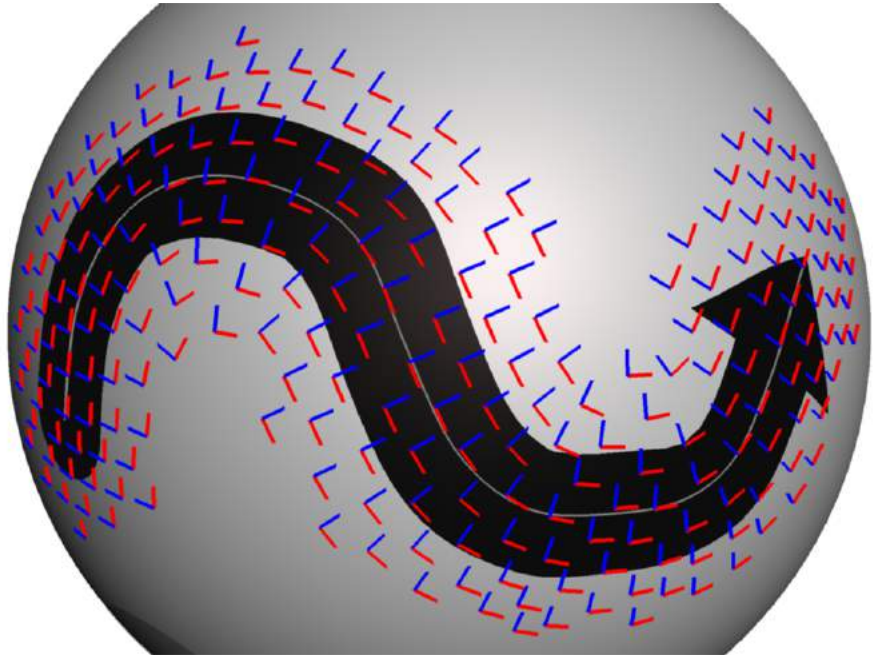


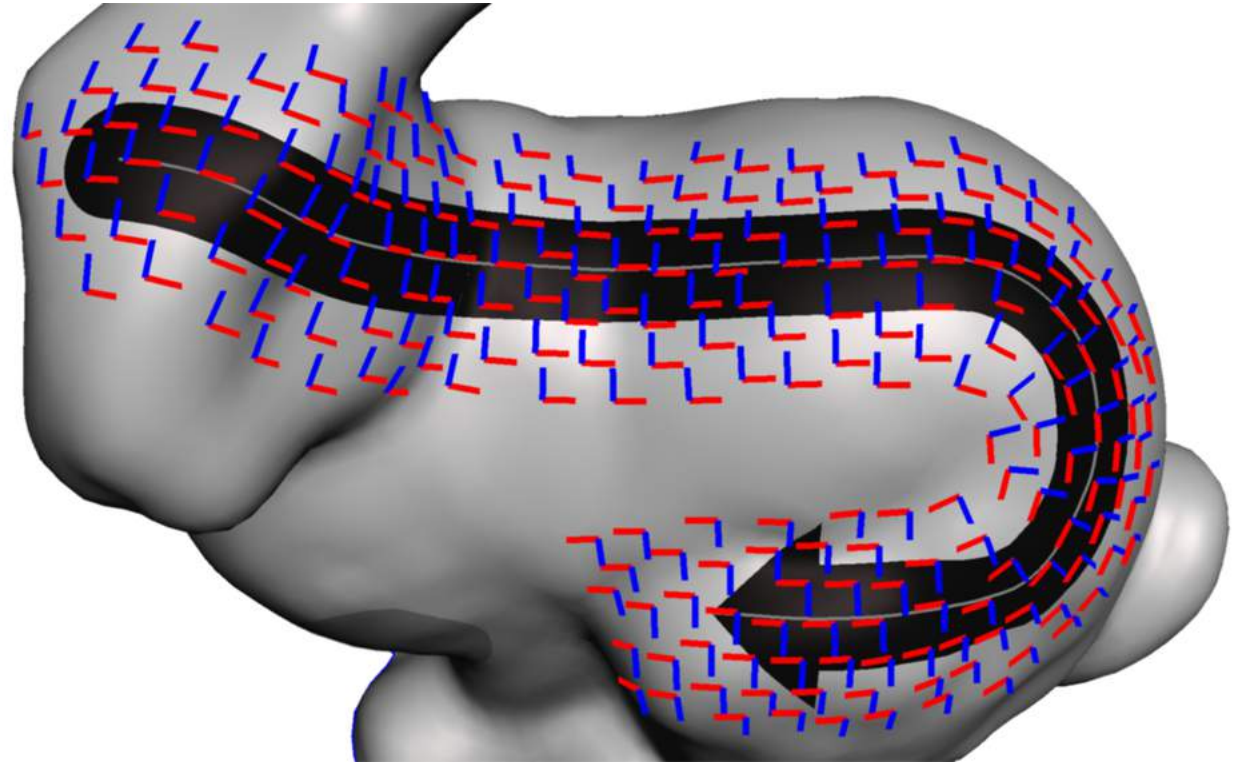
Applications

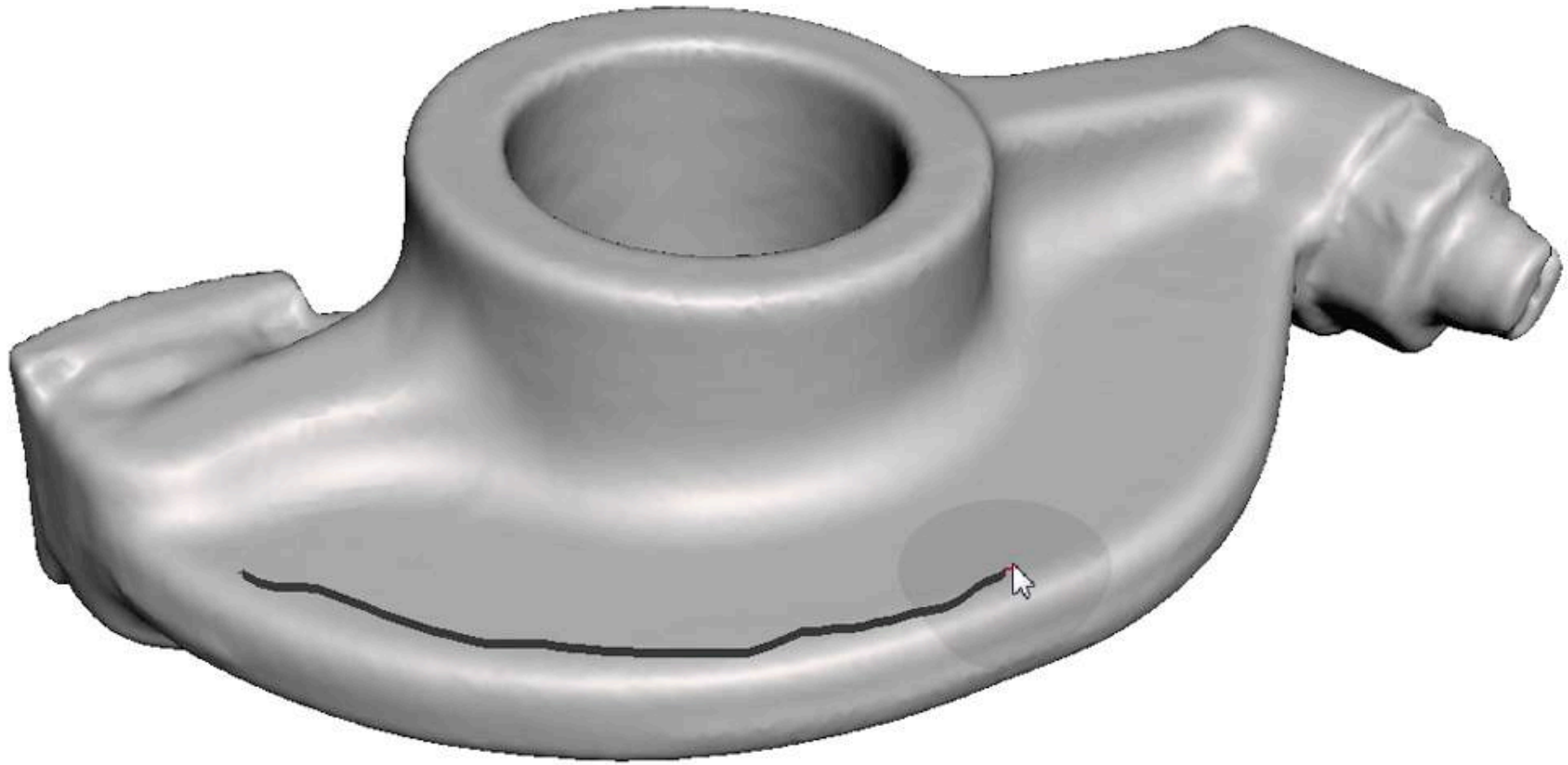


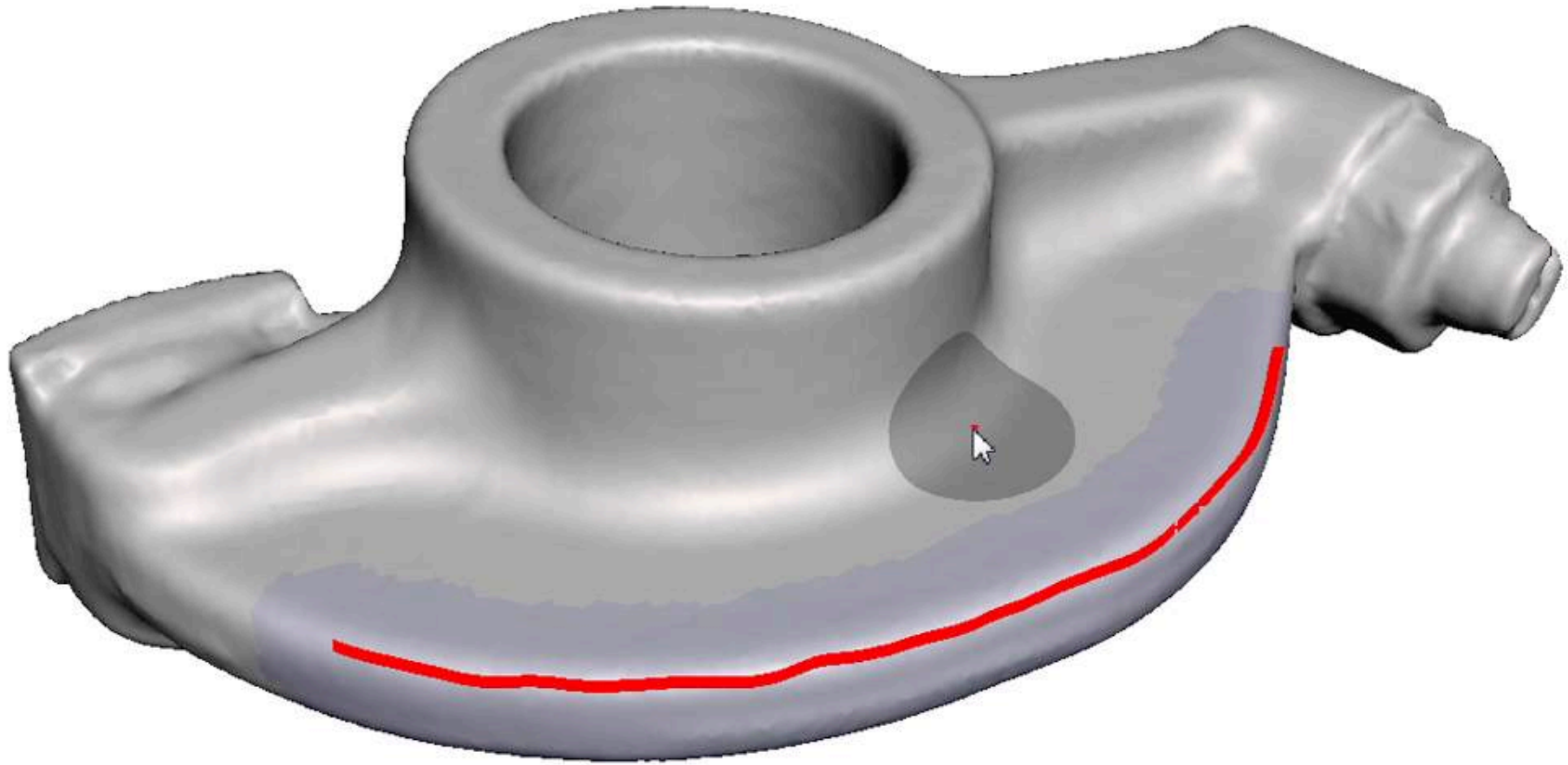


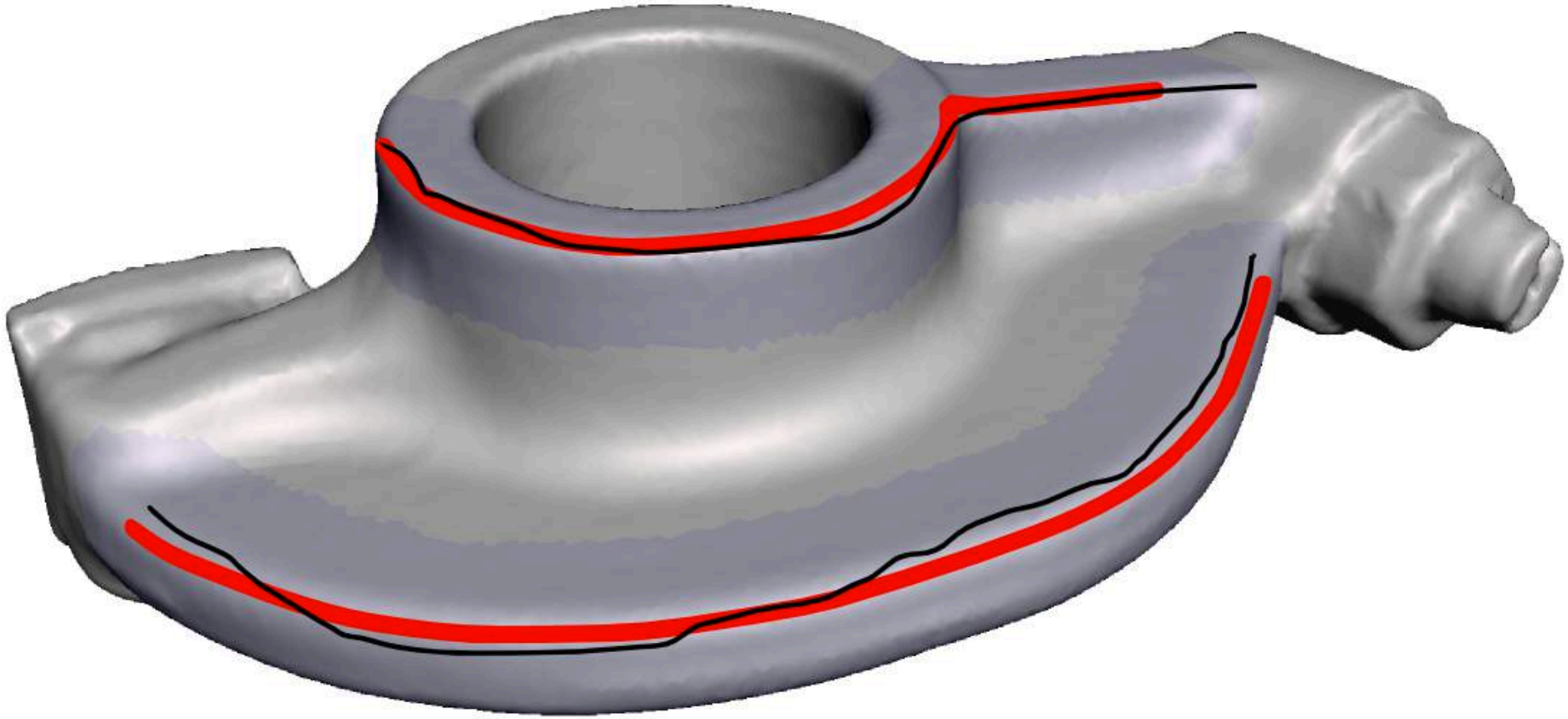


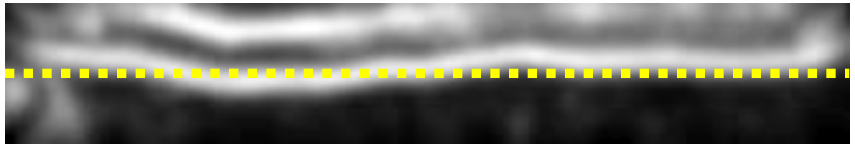


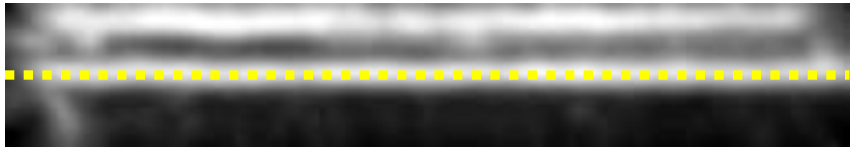






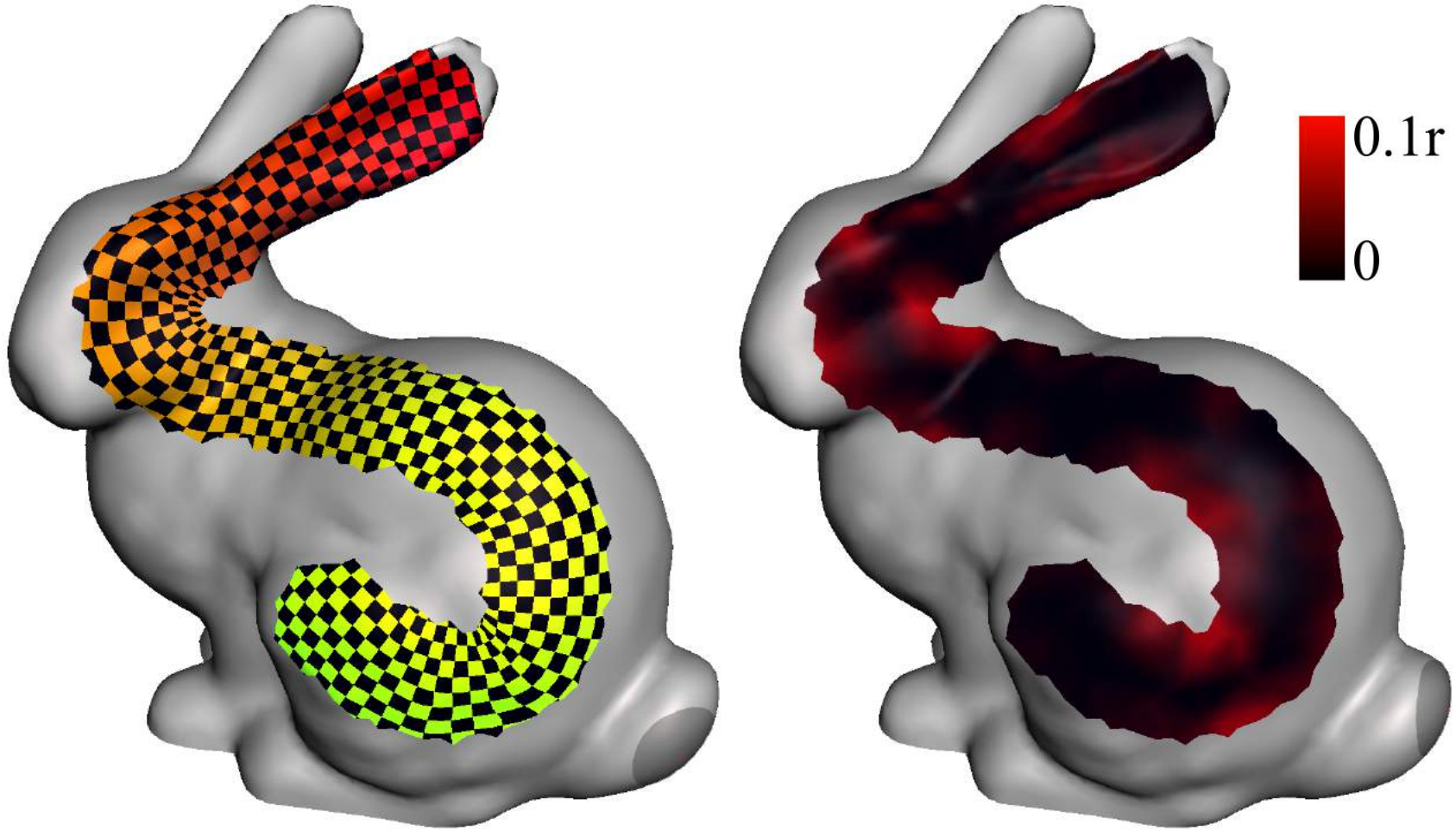




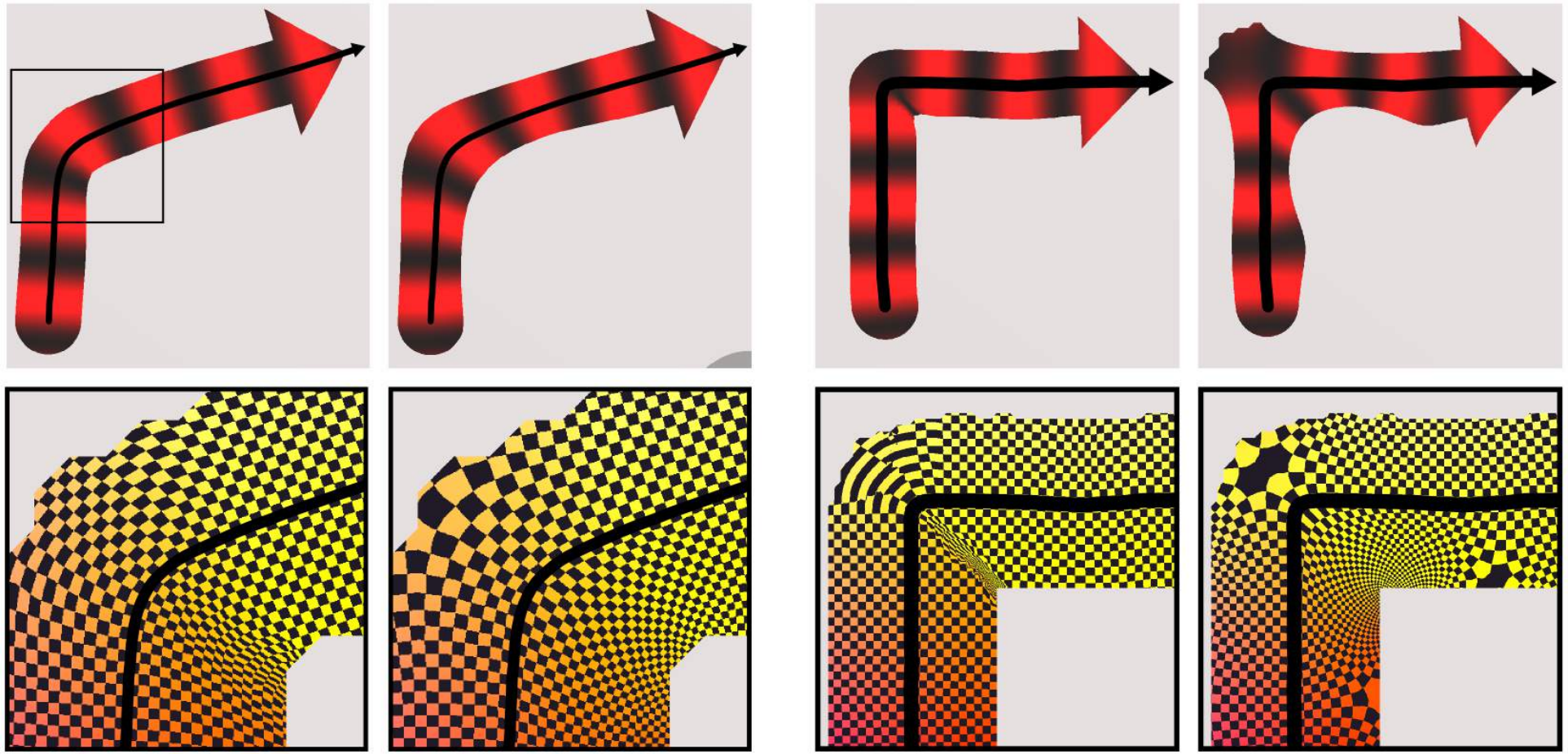


Analysis

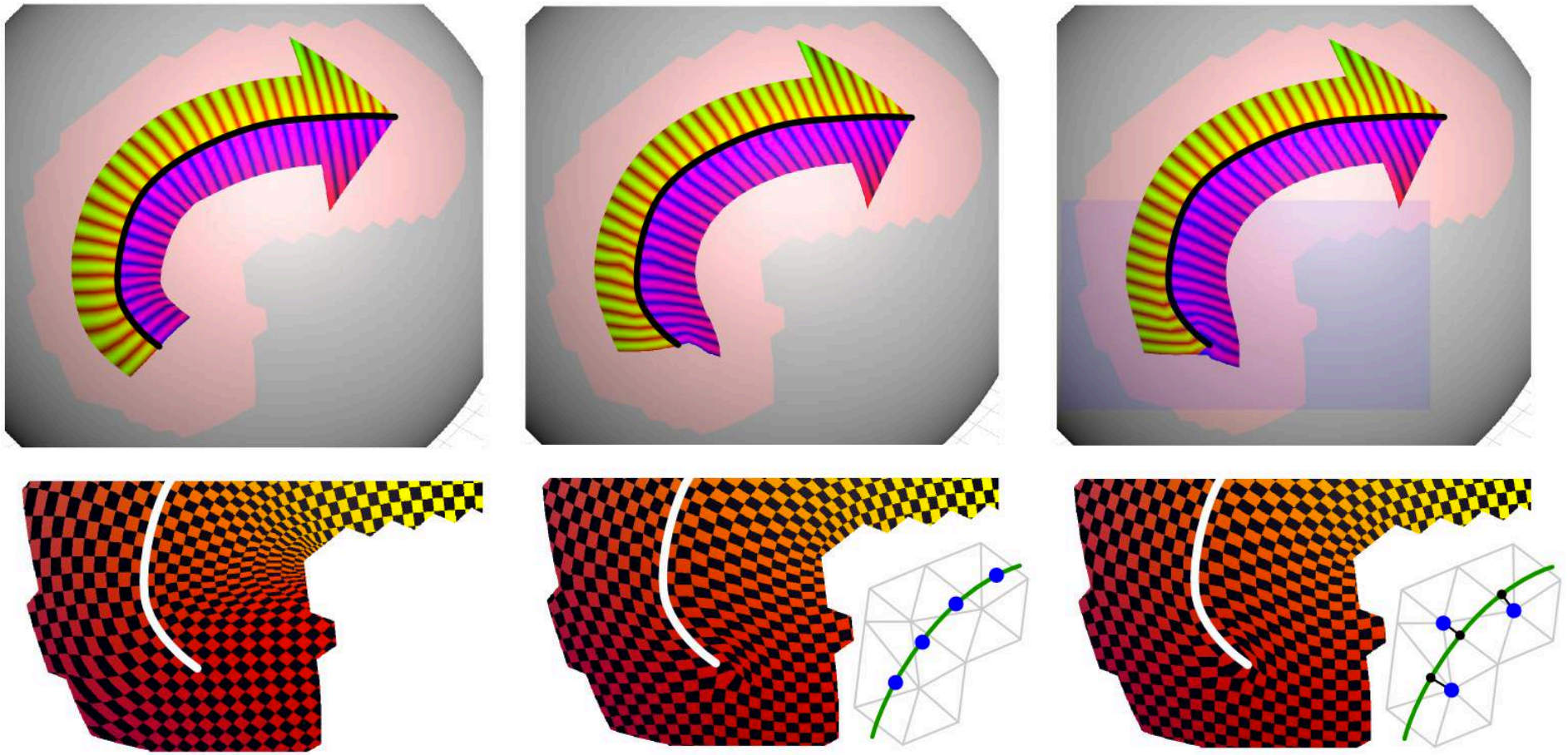
Geodesic Error

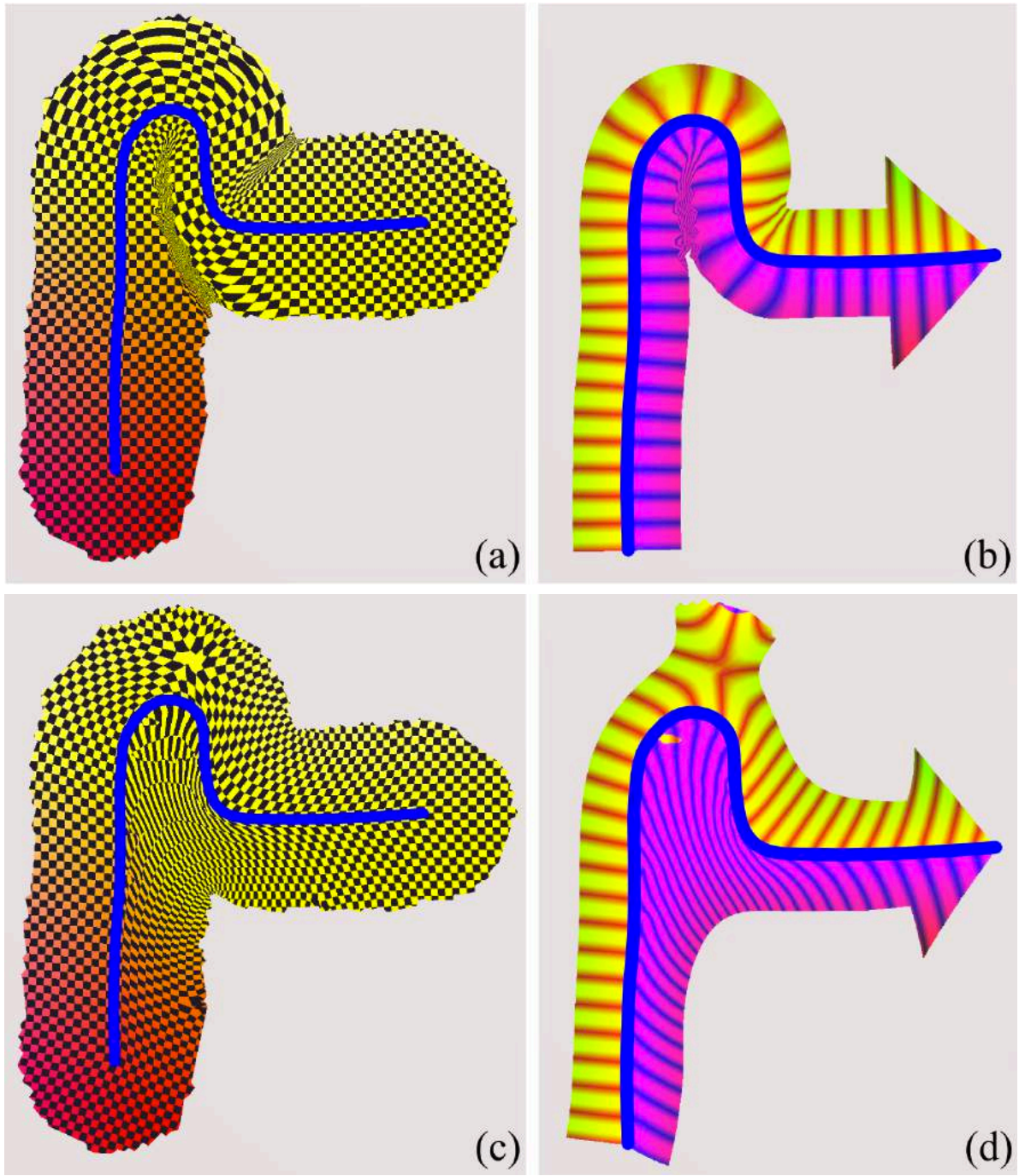


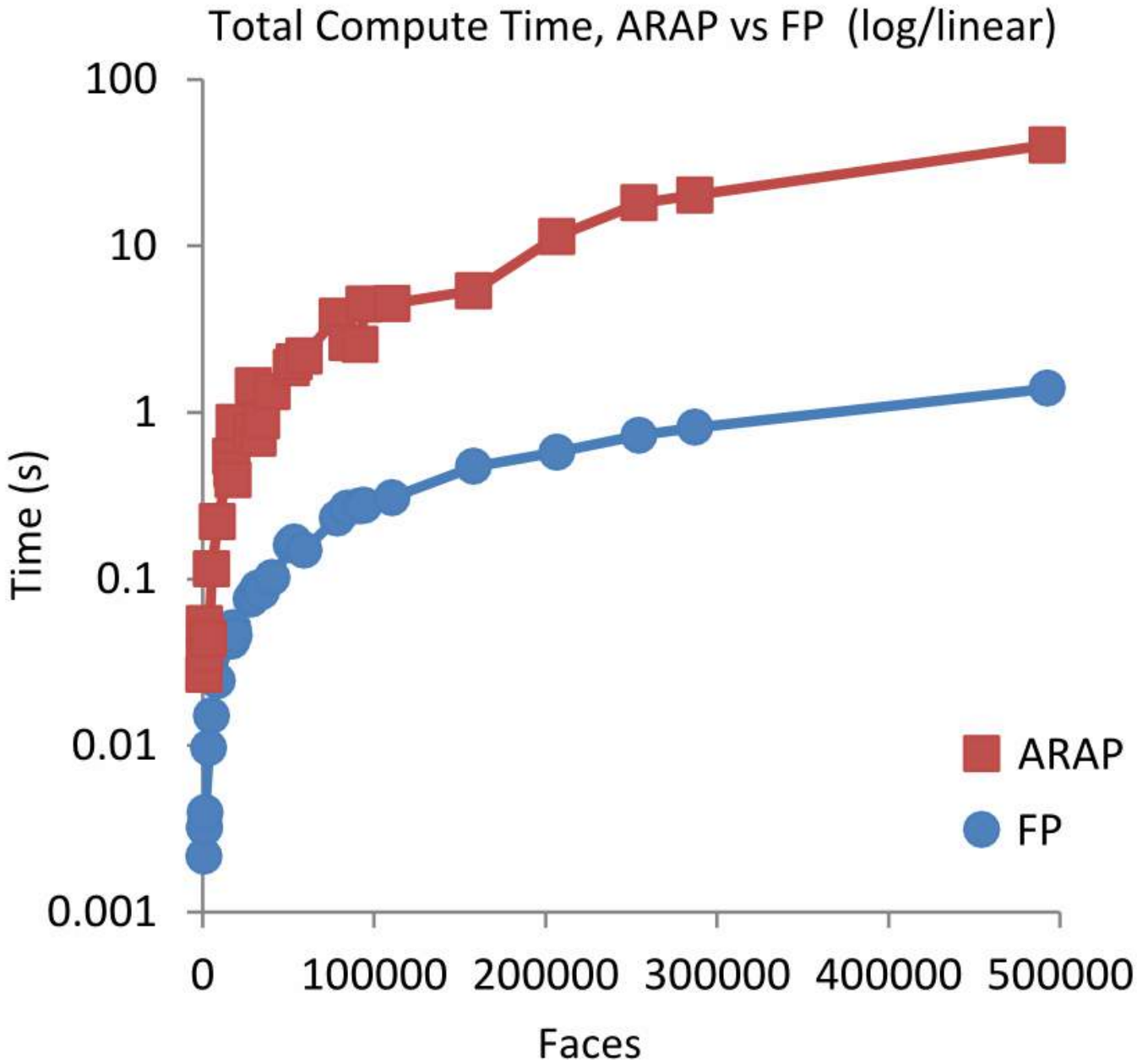
FP vs Conformal Map

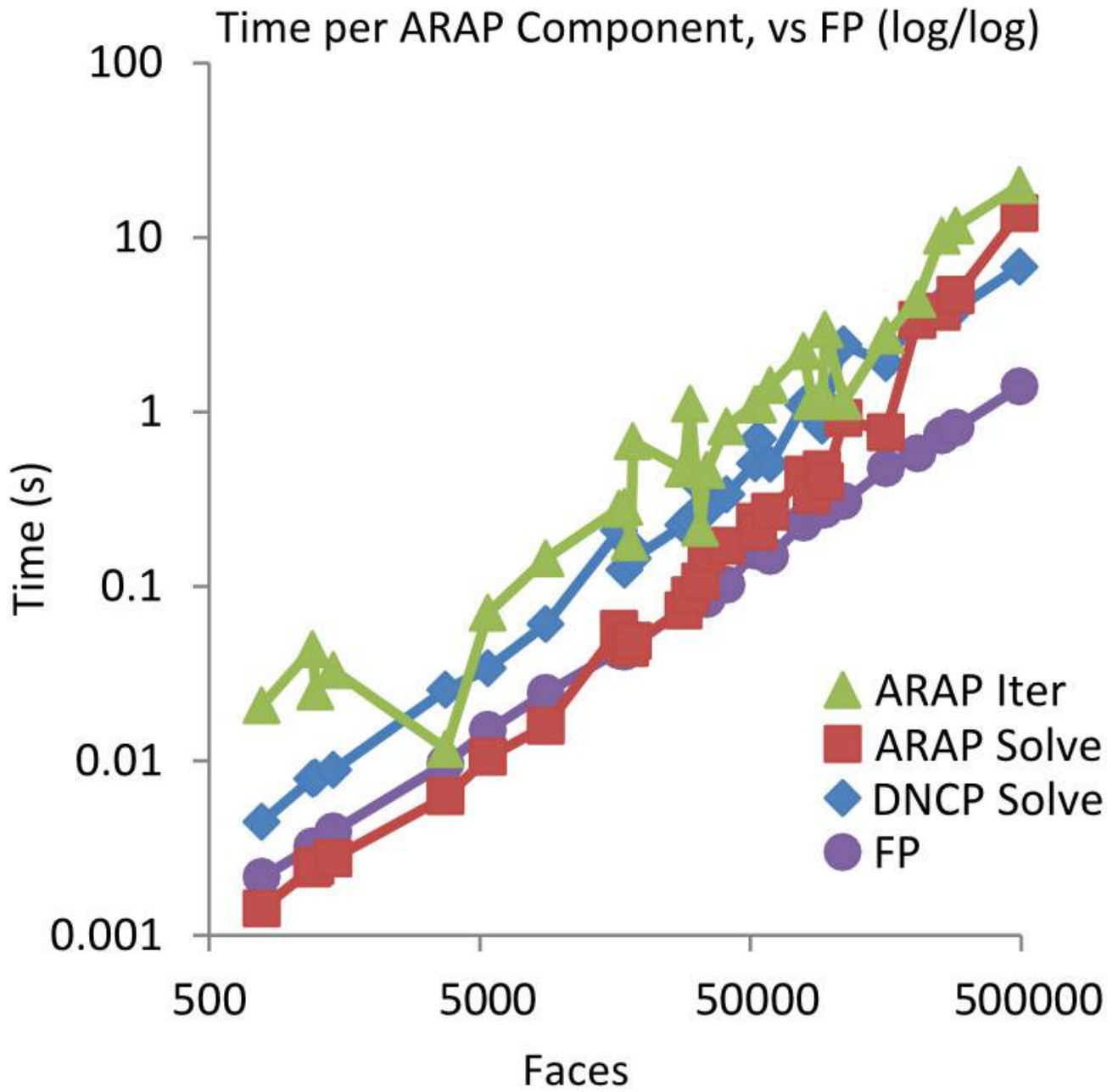


FP vs ARAP



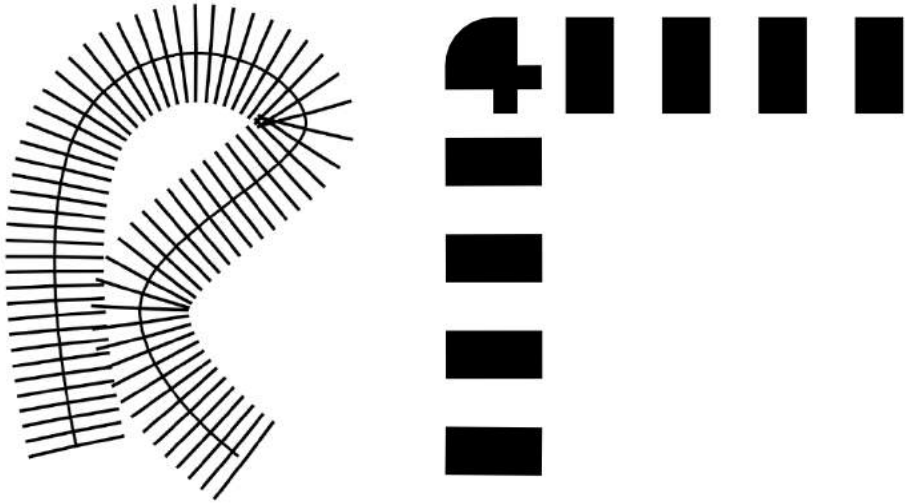

(a)
(b)
(c)
(d)


Total Compute Time, ARAP vs FP (log/linear)
100
10
1
0.1
0.01
0.001
Time (s)
0
100000
200000
300000
400000
500000
Faces
ARAP
FP

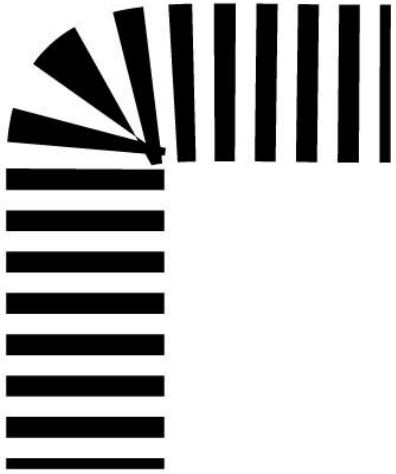

Time per ARAP Component, vs FP (log/log)
100
10
1
0.1
0.01
0.001
Time (s)
500
5000
50000
500000
Faces
ARAP Iter
ARAP Solve
DNCP Solve
FP

Limitations: Self-Intersection

- Not clear that there is a right answer…
- Still not solved in 2D, either



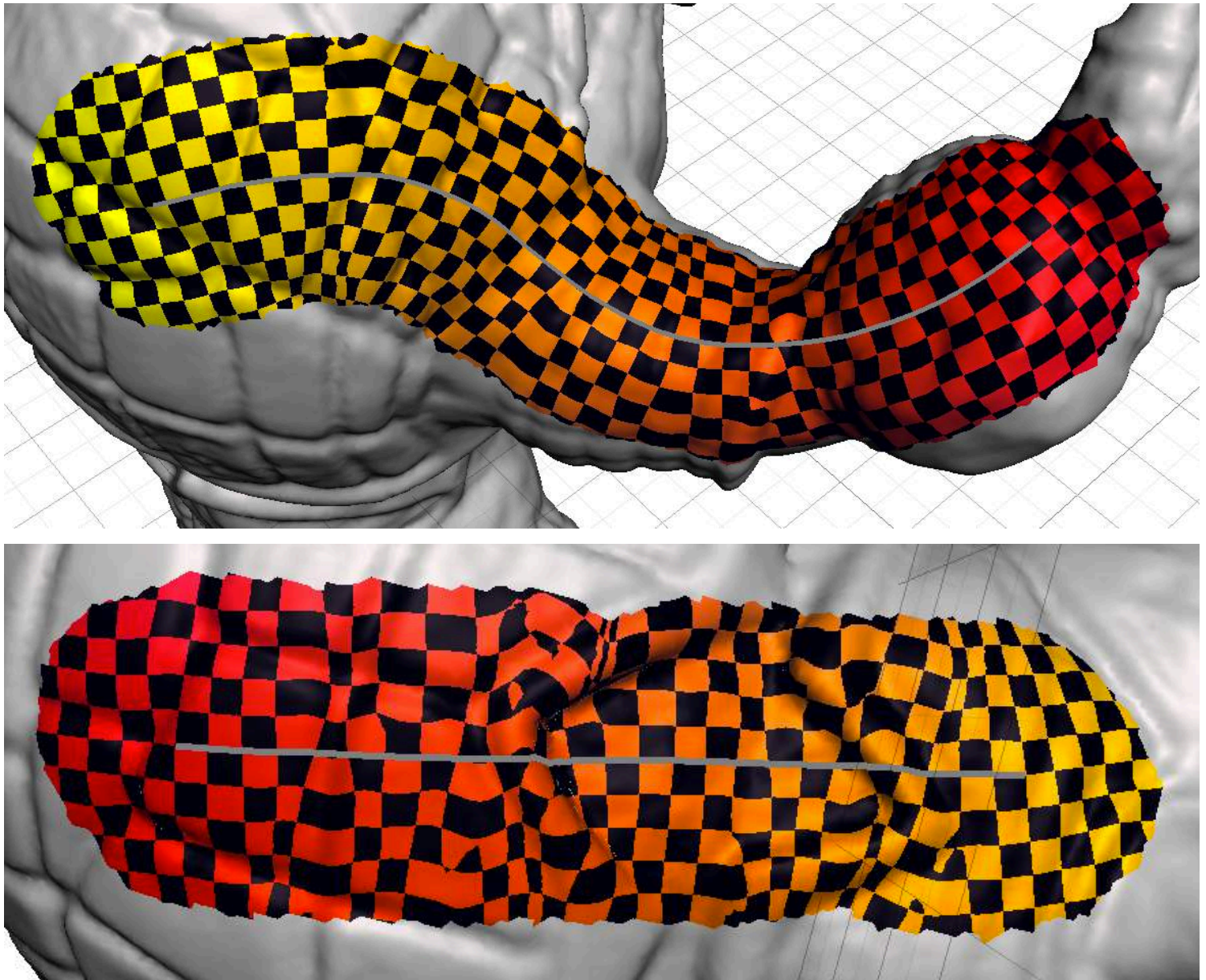
Inkscape



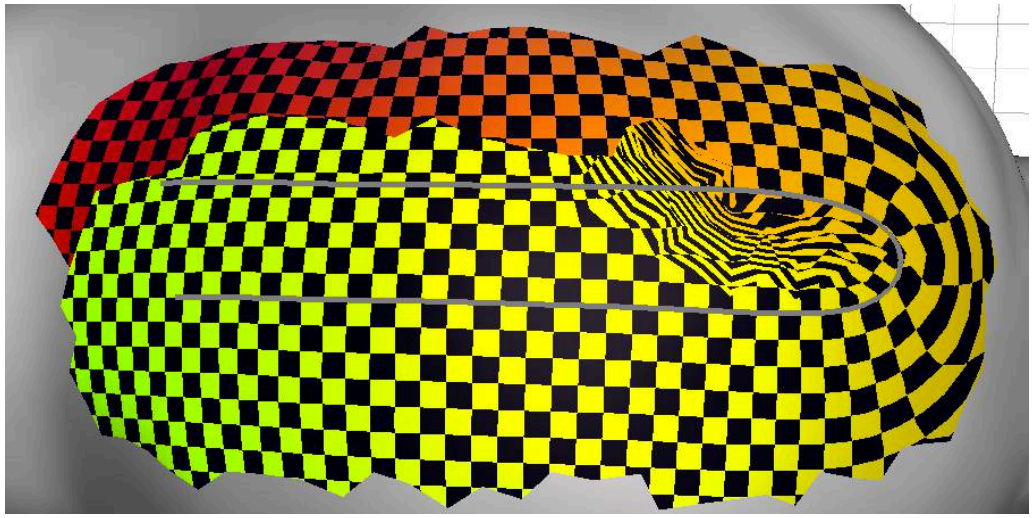
Adobe Illustrator

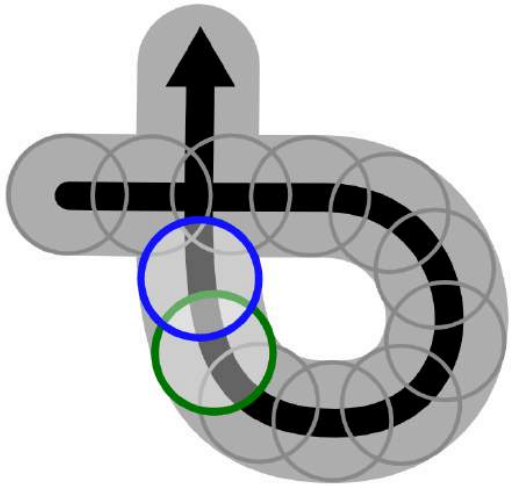
By-Example Synthesis of Curvilinear Structured Patterns, Eurographics 2013
(Next paper session in Room B.1)

Limitations: Distortion / Smoothness



Limitations: Overlap Discrimination





Future Work

- Better constraints for ARAP
- Improved overlap handling
- Closed Loops
  - map to cylinder