252-0538-00L, Spring 2025

Shape Modeling and Geometry Processing

Mesh Editing II

INTERACTIVE GEOMETRY LAB

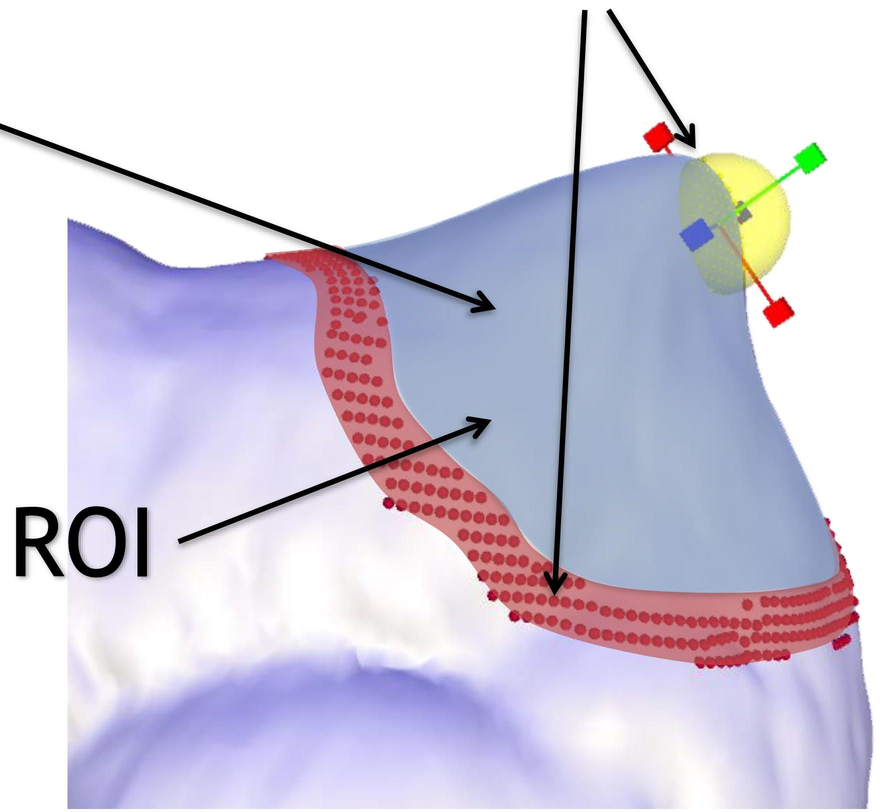
May 7, 2025

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Surface-based Deformation: ROI-Handle Editing Metaphor

$$\mathbf{x}_{\text{def}} = \underset{\mathbf{x}'}{\operatorname{argmin}}\, E(\mathbf{x}') \quad s.t. \quad \mathbf{x}'_i = \mathbf{c}_i \;\; \forall i \in \mathcal{C}$$

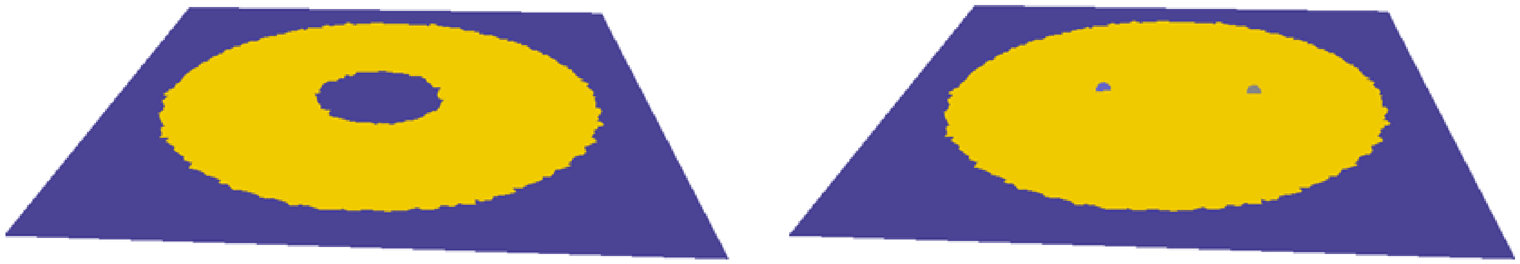
- ROI is bounded by a belt (static anchors)
- Manipulation through handle(s) – affine transformations



Surface-based Deformation: ROI-Handle Editing Metaphor

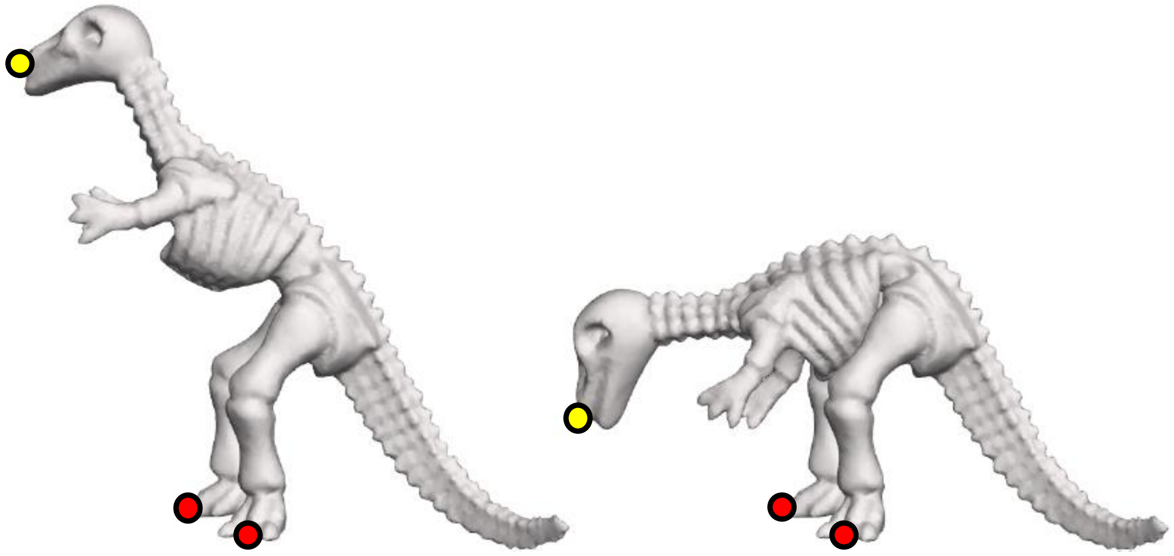
$$\mathbf{x}_{\text{def}} = \underset{\mathbf{x}'}{\operatorname{argmin}}\, E(\mathbf{x}') \quad s.t. \quad \mathbf{x}'_i = \mathbf{c}_i \;\forall i \in \mathcal{C}$$

- ROI is bounded by a belt (static anchors)
- Manipulation through handle(s) – affine transformations



How to Define $E(\mathbf{x}')$?

- Intuitive deformations:
  - Smooth deformation on the global scale
  - Preserve local details (curvatures)
- Invariants: $E(\mathbf{x}')$ should be zero if $\mathbf{x}'$ is a rigid transformation of original geometry $\mathbf{x}$



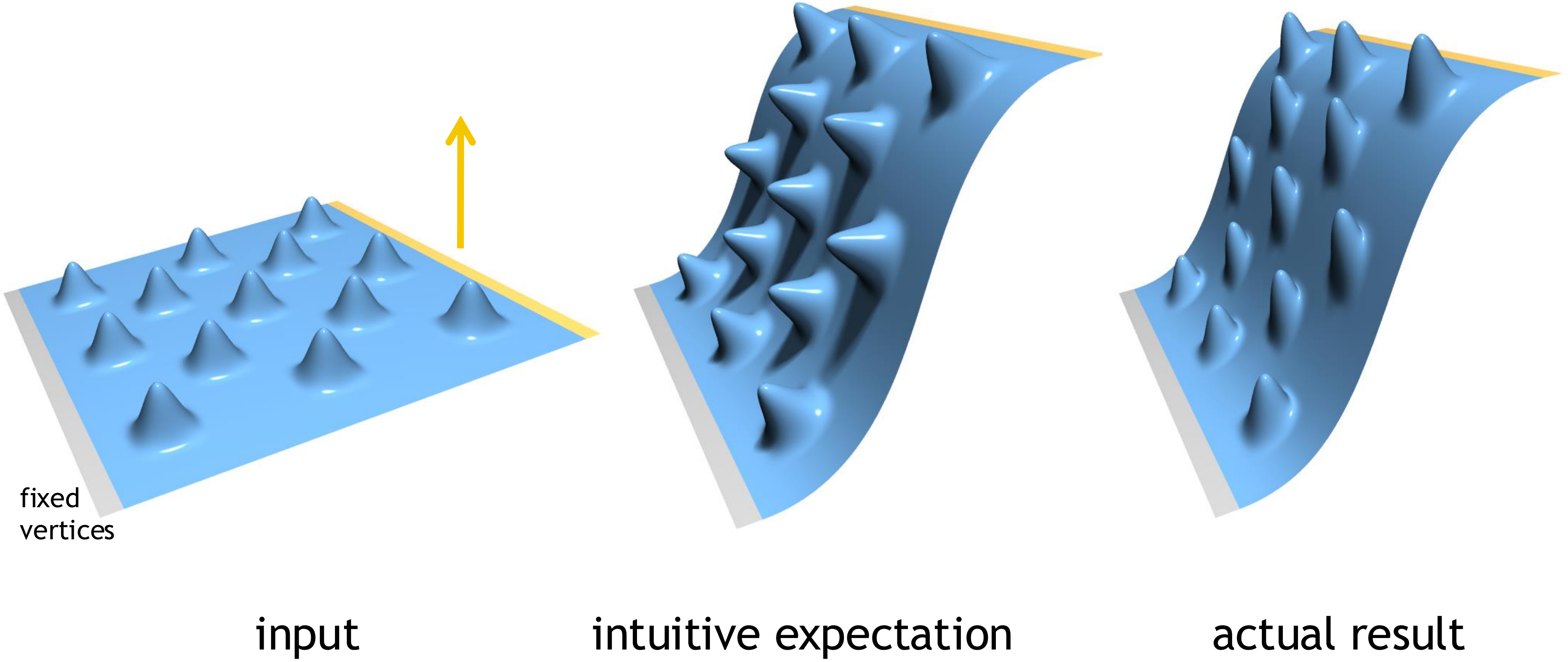
Simple Laplacian Editing

- Preserve mean curvature normal [≈ differential coordinates] at every point in the ROI [≈ every vertex of the ROI]

continuous: $$E(\mathcal{S}') = \int_{\mathcal{S}'} \|\Delta \mathbf{x}' - \delta\|^2 d\mathbf{x}'$$

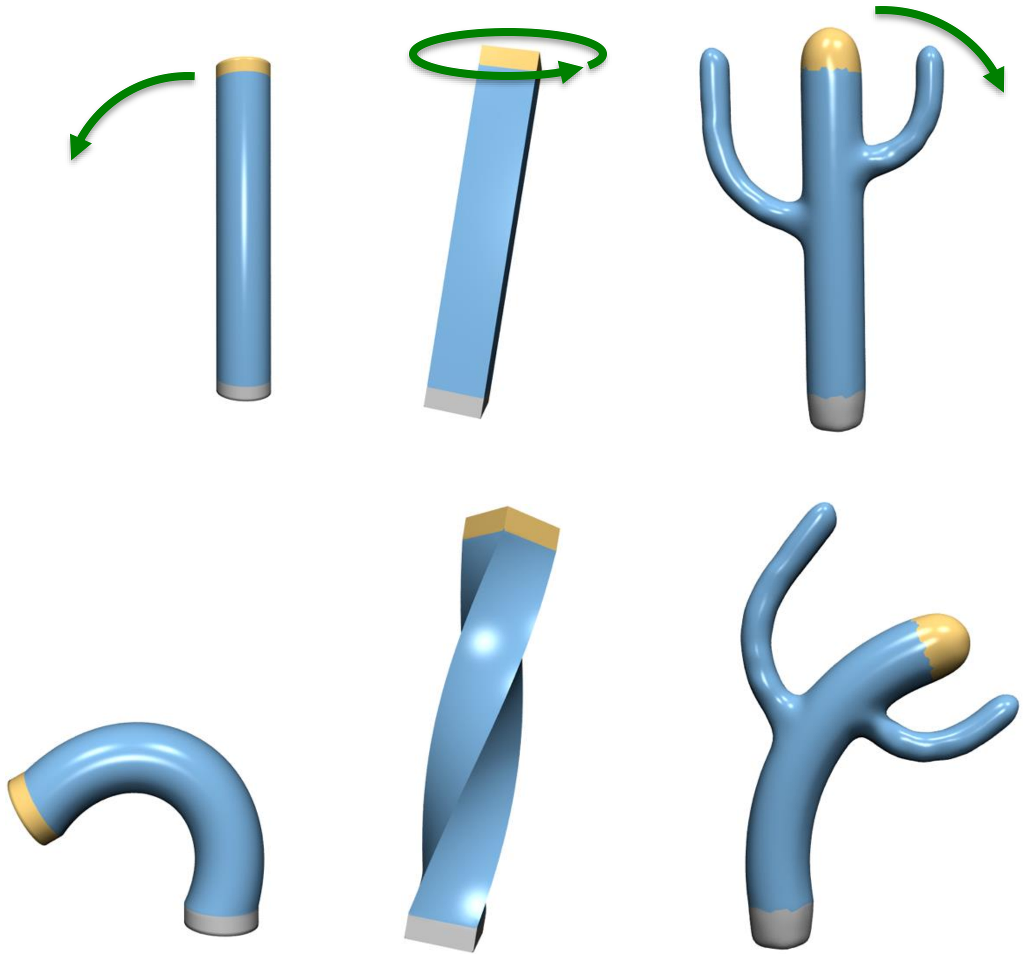
discrete: $$E(\mathbf{x}') = \sum_{i=1}^{n} A_i \|\Delta(\mathbf{x}_i') - \delta_i\|^2$$

Fundamental Problem:
Invariance to Transformations



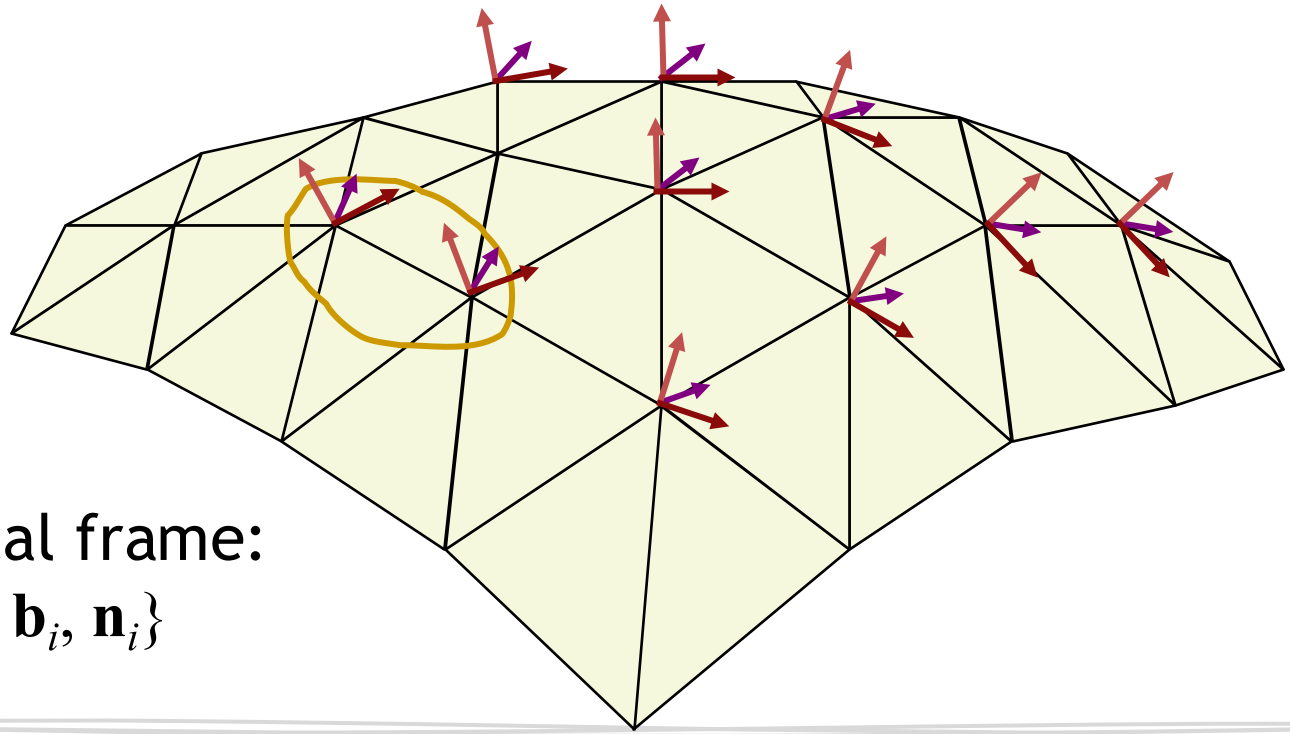
Rotation Propagation

- Propagate using a harmonic scalar field from handle to fixed region
- Works for single handle, user prescribes same rotation for all vertices in the handle



Optimization of Rotation Propagation
Lipman et al. 2005

- Keep a local orthonormal frame at each vertex
- Prescribe changes to some selected frames (rotation/scaling)



Local frame:
$\{\mathbf{a}_i, \mathbf{b}_i, \mathbf{n}_i\}$

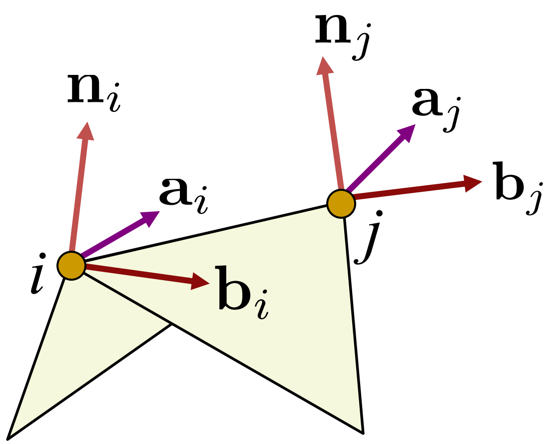
Optimization of Rotation Propagation

Lipman et al. 2005

- Encode the differences between adjacent frames in the original mesh – the numbers $\alpha_{i,j,*}$ $\beta_{i,j,*}$ $\gamma_{i,j,*}$ – for each edge (i,j)
  - Represented in the **local frame** coordinate system!

- Want the deformed surface to have the same local differences
  - Rotation-invariant representation!

$$\mathbf{a}_i - \mathbf{a}_j = \alpha_{i,j,1}\mathbf{a}_i + \alpha_{i,j,2}\mathbf{b}_i + \alpha_{i,j,3}\mathbf{n}_i$$
$$\mathbf{b}_i - \mathbf{b}_j = \beta_{i,j,1}\mathbf{a}_i + \beta_{i,j,2}\mathbf{b}_i + \beta_{i,j,3}\mathbf{n}_i$$
$$\mathbf{n}_i - \mathbf{n}_j = \gamma_{i,j,1}\mathbf{a}_i + \gamma_{i,j,2}\mathbf{b}_i + \gamma_{i,j,3}\mathbf{n}_i$$



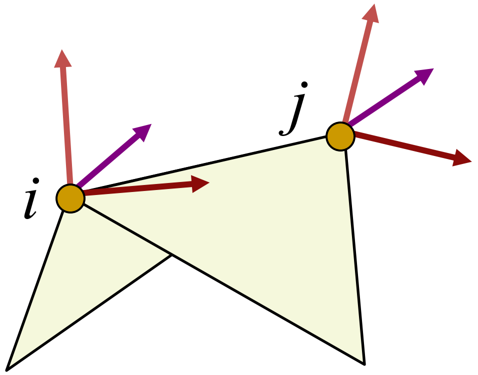
Optimization of Rotation Propagation

Lipman et al. 2005

- Solve for the new frames in least-squares sense
- Need to orthogonalize (and normalize) post-hoc

$$\min_{\mathbf{a}',\mathbf{b}',\mathbf{n}'} \sum_{i=1}^{n} \|(\mathbf{a}'_i - \mathbf{a}'_j) - (\alpha_{i,1}\mathbf{a}'_i + \alpha_{i,2}\mathbf{b}'_i + \alpha_{i,3}\mathbf{n}'_i)\|^2 + \|(\mathbf{b}'_i - \mathbf{b}'_j) - (\beta_{i,1}\mathbf{a}'_i + \beta_{i,2}\mathbf{b}'_i + \beta_{i,3}\mathbf{n}'_i)\|^2 + \|(\mathbf{n}'_i - \mathbf{n}'_j) - (\gamma_{i,1}\mathbf{a}'_i + \gamma_{i,2}\mathbf{b}'_i + \gamma_{i,3}\mathbf{n}'_i)\|^2$$

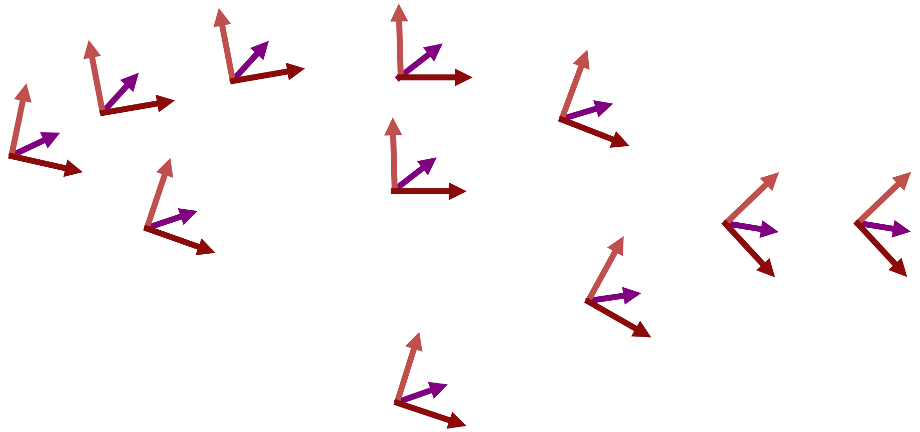
$$s.t. \quad (\mathbf{a}'_k, \mathbf{b}'_k, \mathbf{n}'_k) = \mathbf{M}_k, \; k \in \mathcal{C}$$



Optimization of Rotation Propagation

Lipman et al. 2005

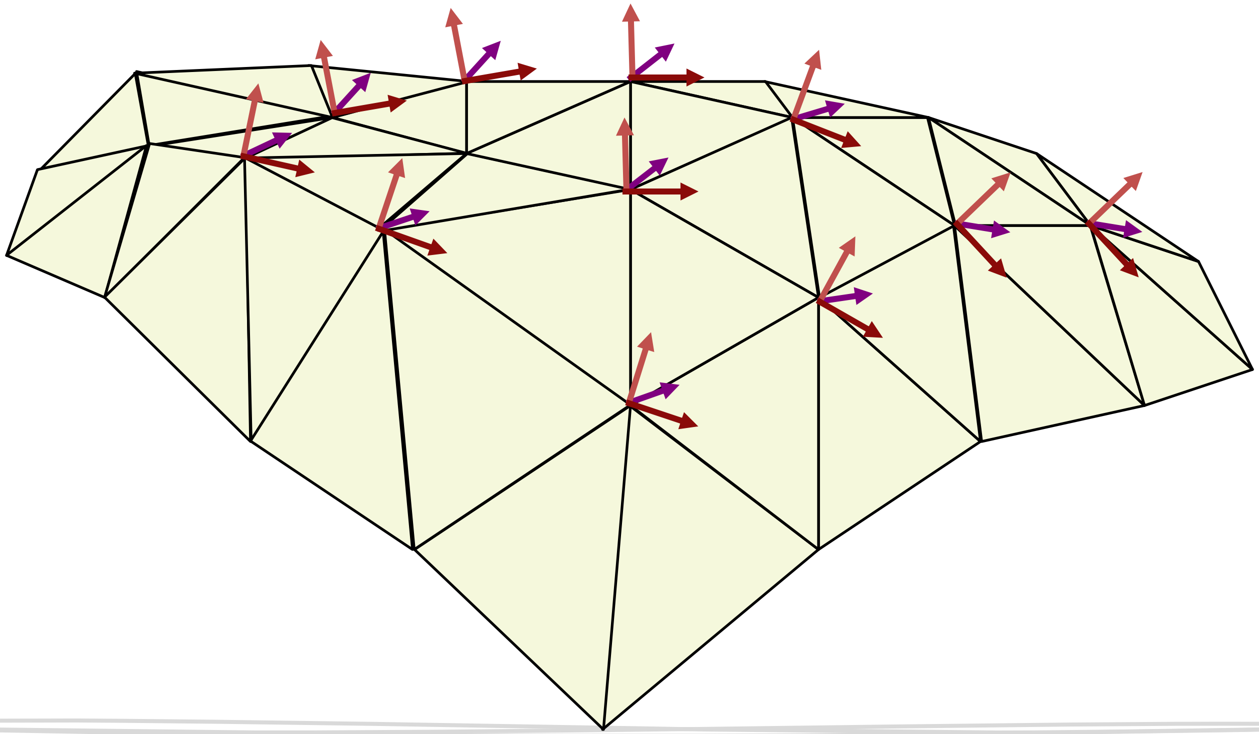
- After solving the frames, solve for positions using e.g. naïve Laplacian editing (rotate each delta-vector…)



Optimization of Rotation Propagation

Lipman et al. 2005

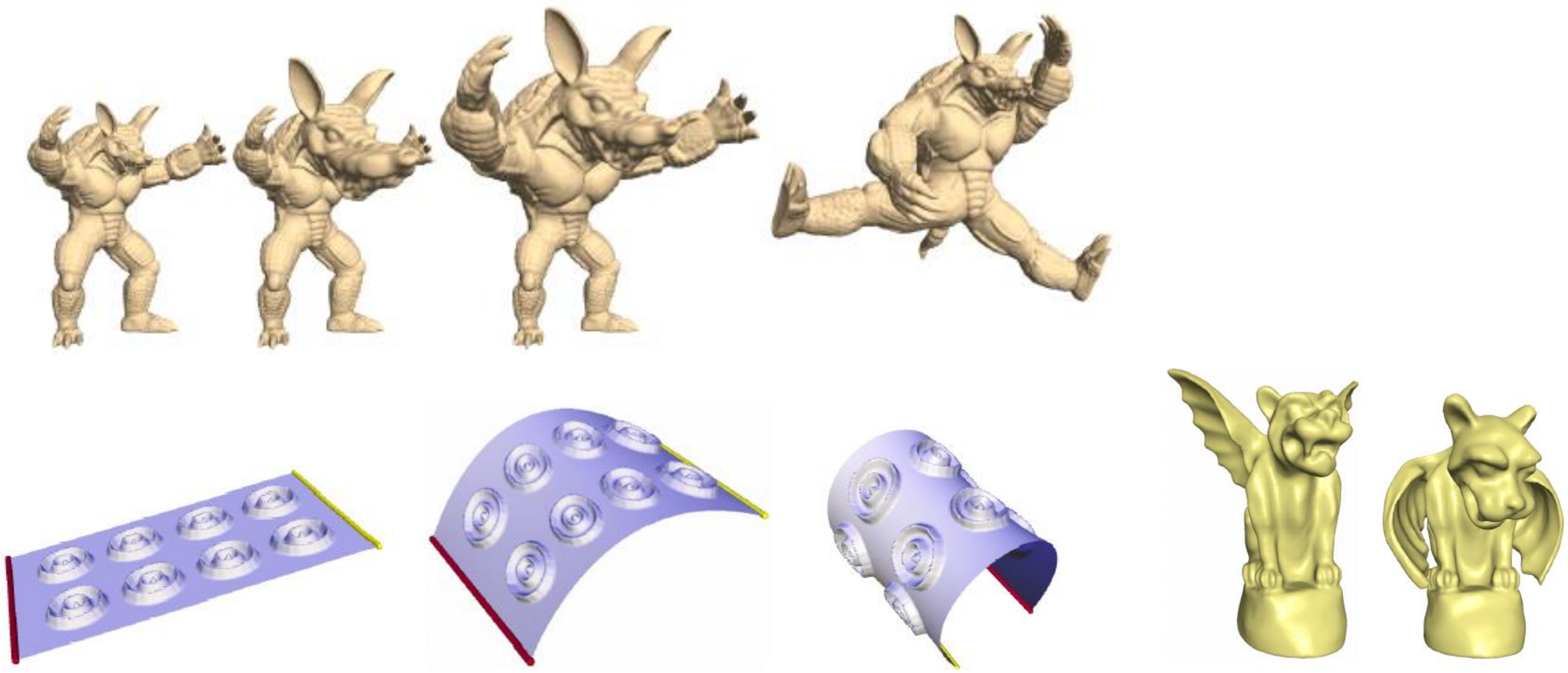
- After solving the frames, solve for positions using e.g. naïve Laplacian editing (rotate each delta-vector…)



Optimization of Rotation Propagation

Lipman et al. 2005

- Some results



Optimization of Rotation Propagation

Lipman et al. 2005

- Can use this representation for shape interpolation

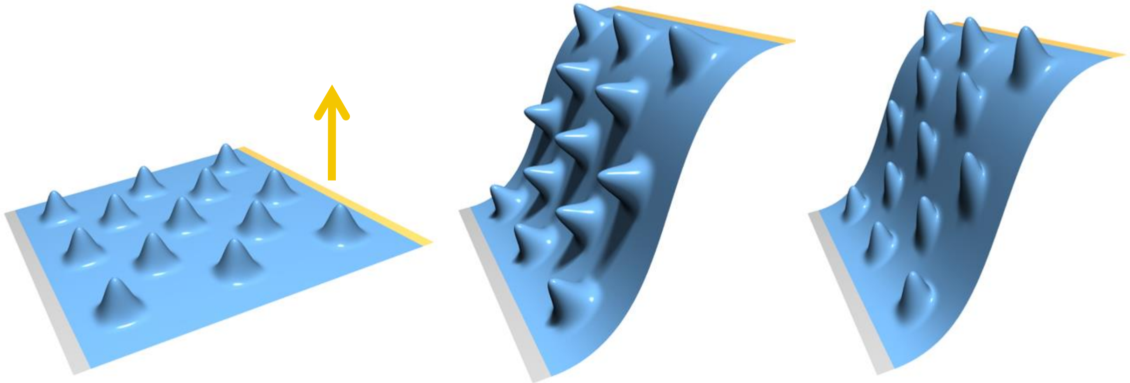
Linear Rotation-Invariant Coordinates
for Meshes

Yaron Lipman
Olga Sorkine
David Levin
Daniel Cohen-Or

Tel Aviv University

Rotation Propagation – Summary

- Linear optimization to find the local frames of the deformed surface
- Works well even for large rotations of the handle
- Does not work if there is no rotation to propagate – translation insensitivity



Implicit Definition of Transformations

Sorkine et al. 2004

- The idea: solve for local transformations AND the edited surface simultaneously!
- Estimate the local transformations $\mathbf{T}_i$ from the eventual, unknown solution

$$E(\mathbf{x}') = \sum_{i=1}^{n} A_i \|\Delta(\mathbf{x}'_i) - \mathbf{T}_i \delta_i\|^2$$

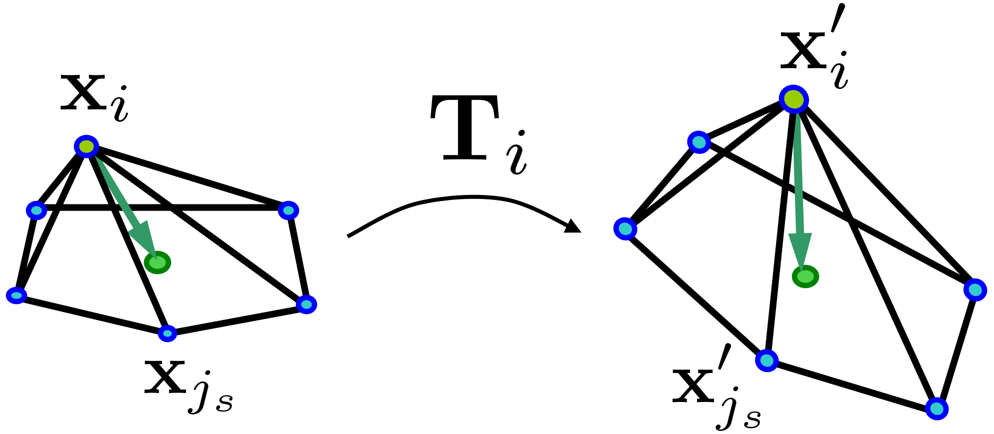
Linear transformation of the local frame

Defining $\mathbf{T}_i$

$$E(\mathbf{x}') = \sum_{i=1}^{n} A_i \|\Delta(\mathbf{x}'_i) - \mathbf{T}_i \delta_i\|^2$$

- How to formulate $\mathbf{T}_i$?
  - Based on the local (1-ring) neighborhood
  - Linear combination of the unknown $\mathbf{x}'_i$s

$$\mathbf{x}'_{j1} - \mathbf{x}'_i = \mathbf{T}_i (\mathbf{x}_{j1} - \mathbf{x}_i)$$
$$\vdots$$
$$\mathbf{x}'_{jk} - \mathbf{x}'_i = \mathbf{T}_i (\mathbf{x}_{jk} - \mathbf{x}_i)$$



Defining $\mathbf{T}_i$

- First attempt: define $\mathbf{T}_i$ simply by solving

$$\mathbf{T}_i = \underset{\mathbf{T}_i}{\operatorname{argmin}} \sum_{s=1}^{k} \left\| (\mathbf{x}'_{j_s} - \mathbf{x}'_i) - \mathbf{T}_i \left(\mathbf{x}_{j_s} - \mathbf{x}_i\right) \right\|^2$$

$$\Downarrow$$

$$\mathbf{T}_i = \begin{pmatrix} | & | & & | \\ (\mathbf{x}'_{j_1} - \mathbf{x}'_i) & (\mathbf{x}'_{j_2} - \mathbf{x}'_i) & \cdots & (\mathbf{x}'_{j_k} - \mathbf{x}'_i) \\ | & | & & | \end{pmatrix} \begin{pmatrix} | & | & & | \\ (\mathbf{x}_{j_1} - \mathbf{x}_i) & (\mathbf{x}_{j_2} - \mathbf{x}_i) & \cdots & (\mathbf{x}_{j_k} - \mathbf{x}_i) \\ | & | & & | \end{pmatrix}^+$$

pseudoinverse

Defining $\mathbf{T}_i$

- Plug the expressions for $\mathbf{T}_i$ into the energy formula:

$$E(\mathbf{x}') = \sum_{i=1}^{n} A_i \| \Delta(\mathbf{x}'_i) - \mathbf{T}_i \delta_i \|^2$$

Linear combination of the unknown $\mathbf{x}'$

But: we didn't solve anything since $\mathbf{T}_i$ is an arbitrary linear transformation, i.e. it admits distorting shears.

Constraining $\mathbf{T}_i$

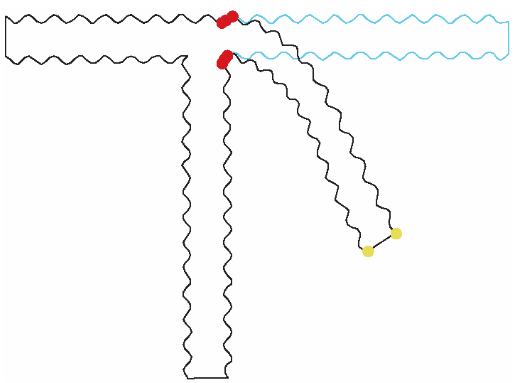
- Rotation + scale (i.e., similarity) is easy in 2D:

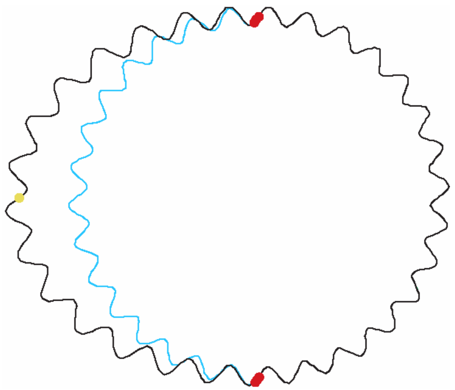
$$\mathbf{T}_i = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$$

  - Rotation alone is nonlinear (bounds on a and b)

Recall LSCM mesh parameterization

- Can edit 2D curves:





Constraining $\mathbf{T}_i$

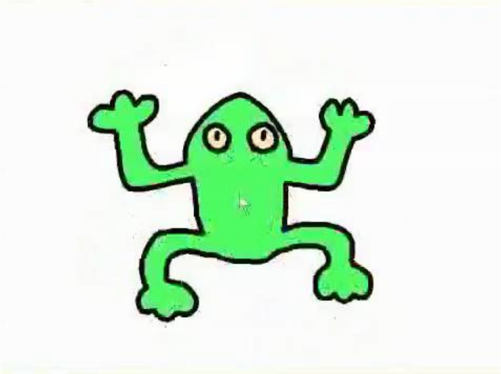
Recall LSCM mesh parameterization

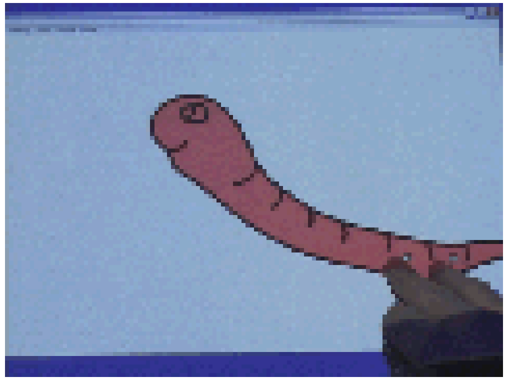
- Rotation + scale (i.e., similarity) is easy in 2D:

$$\mathbf{T}_i = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$$

  - Rotation alone is nonlinear (bounds on a and b)

- Similar idea applied in [Igarashi et al. 2005] for 2D shape manipulation:



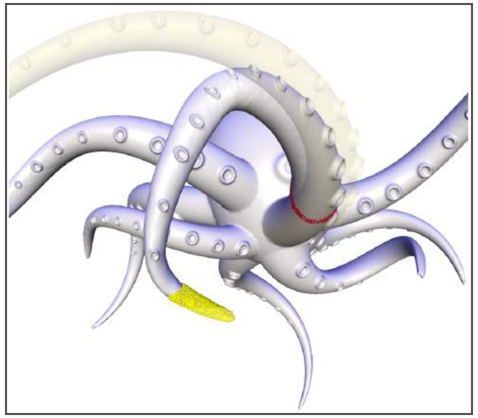


Constraining $\mathbf{T}_i$

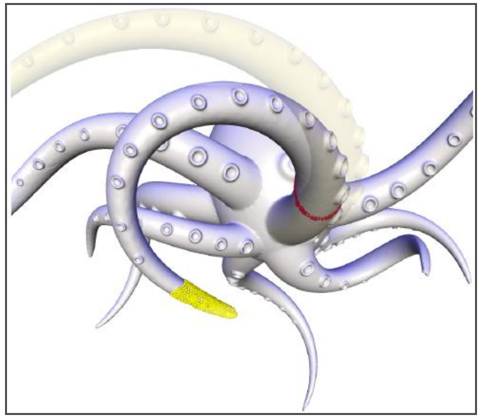
- In 3D: even similarity has nonlinear form.
- Linearization of rotations – first order Taylor approximation: identity + skew-symmetric matrix

$$\mathbf{T}_i = \begin{pmatrix} 1 & -h_3 & h_2 \\ h_3 & 1 & -h_1 \\ -h_2 & h_1 & 1 \end{pmatrix}$$

- Works well for moderate rotations, problems with large rotation angles

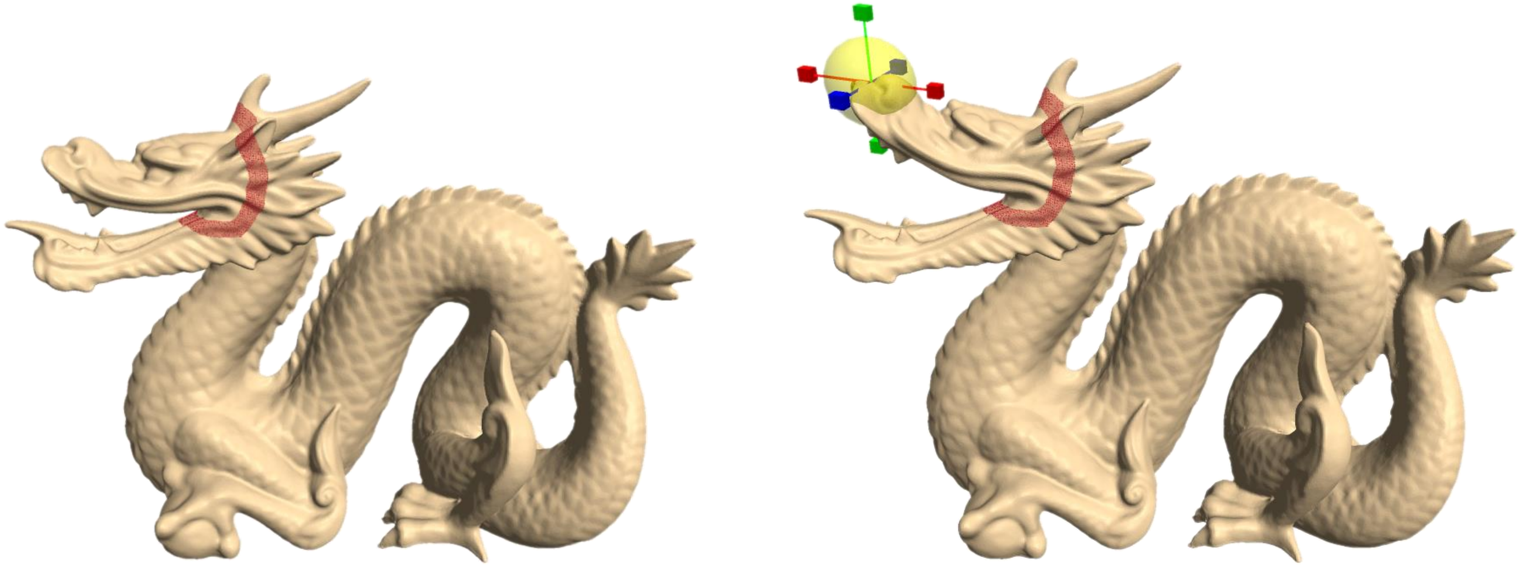


Result of rotation linearization [Sorkine et al. 2004]

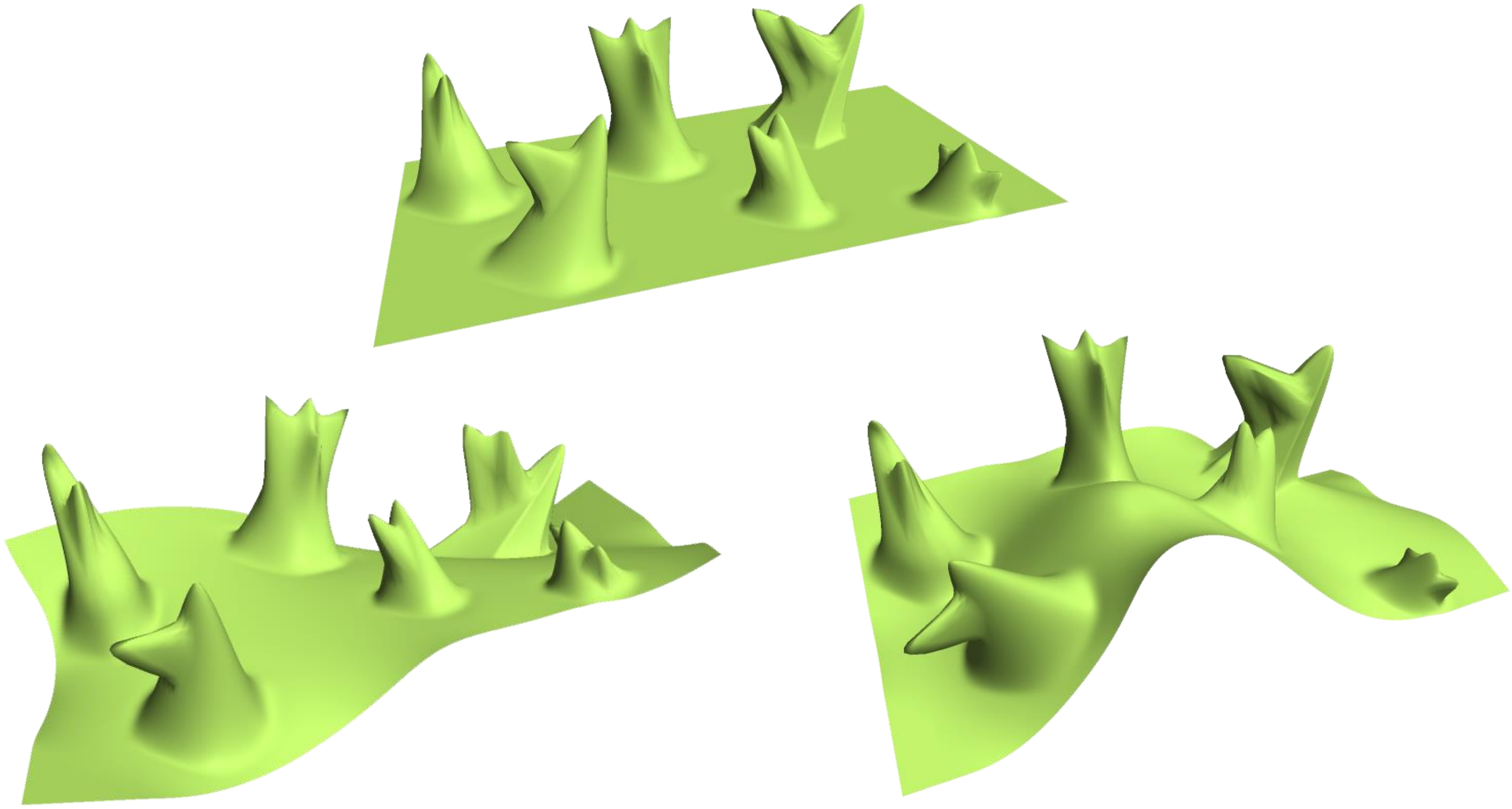


Result of rotation propagation [Lipman et al. 2005]

Laplacian Editing Results

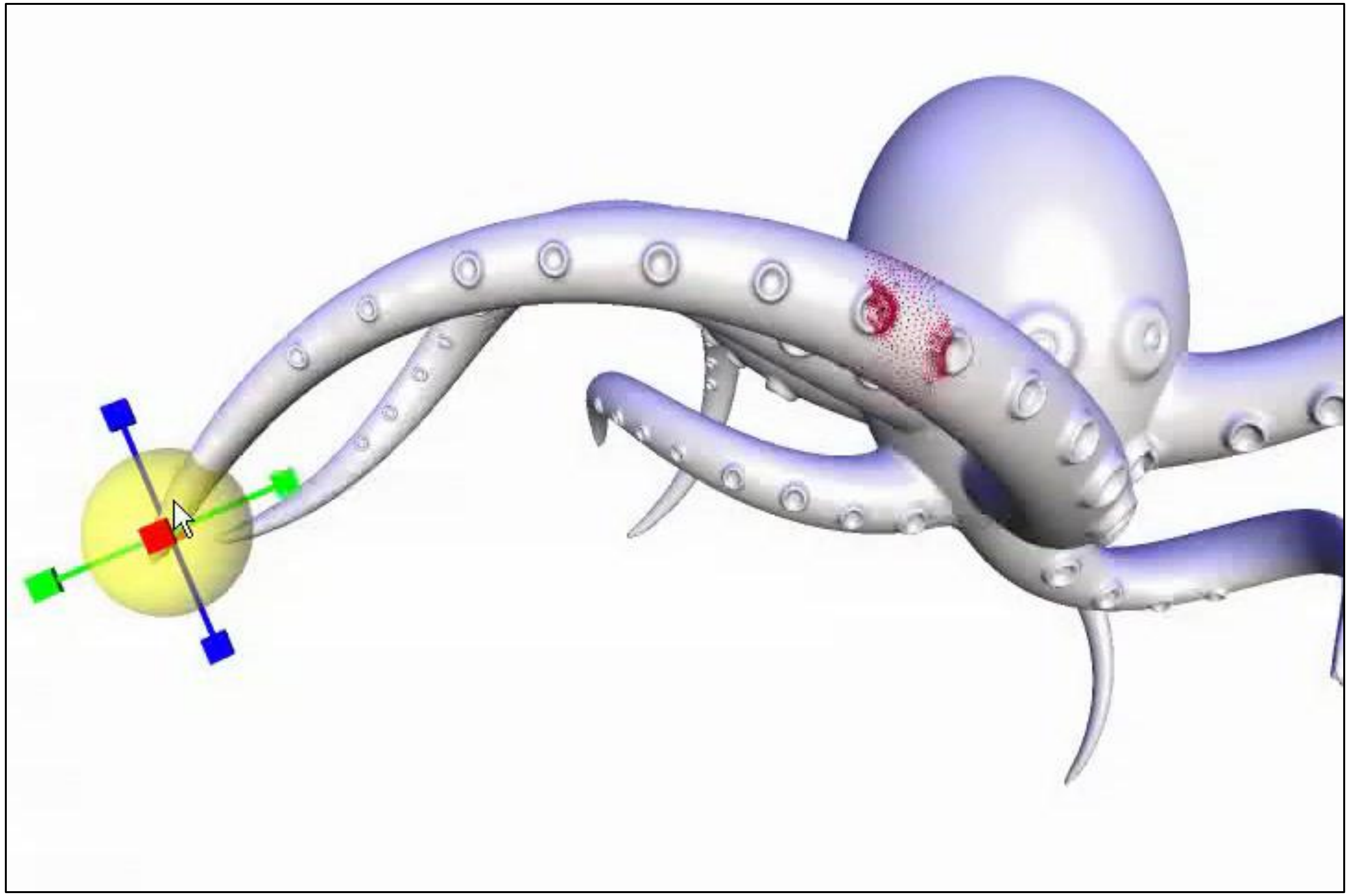


Laplacian Editing Results



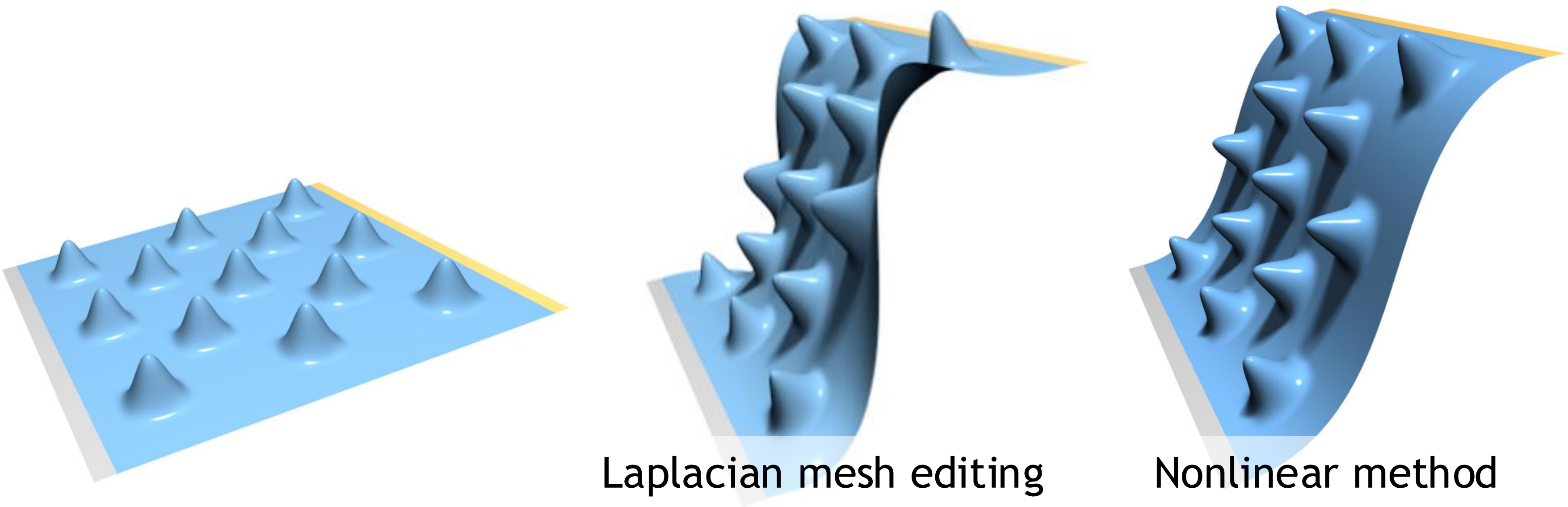
Laplacian Editing Results

Laplacian Editing Results



Linear Deformation Methods: Summary

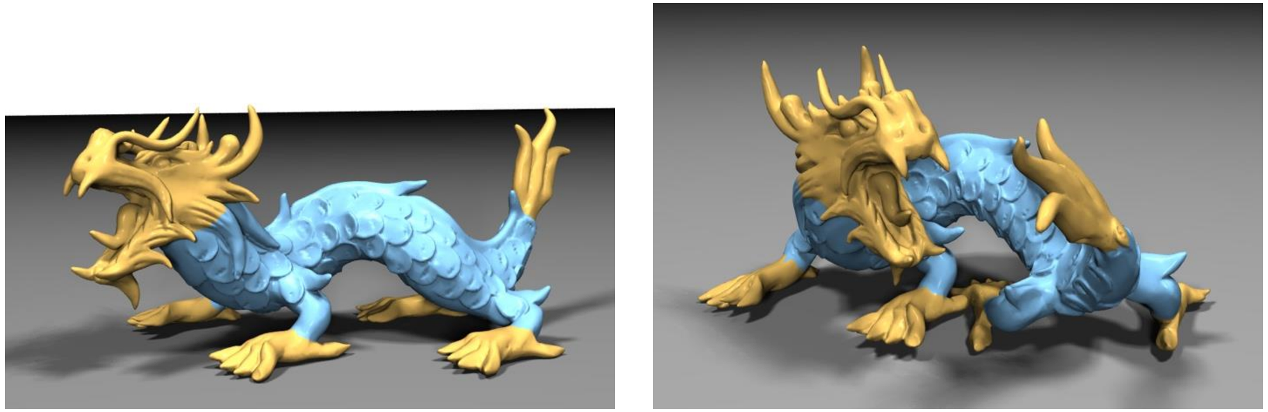
- Involve **linear** global optimization (efficient)
- Suffer from artifacts because of **local rotations**
- The relationship between the **translation** of a handle and the local **rotation** is inherently **nonlinear**



Laplacian mesh editing   Nonlinear method

Nonlinear Surface-based Deformations

- Formulate a nonlinear (non-quadratic) functional $E(\mathbf{x}')$ that handles local rotations properly
- Still need an efficient minimization method…



[PriMo, Botsch et al. 2006]

Literature list

- **Igarashi et al. 2005**
  As-Rigid-As-Possible Shape Manipulation
  https://dl.acm.org/doi/10.1145/1073204.1073323

- **Lipman et al. 2005**
  Linear Rotation-Invariant Coordinates for Meshes
  https://dl.acm.org/doi/10.1145/1073204.1073217

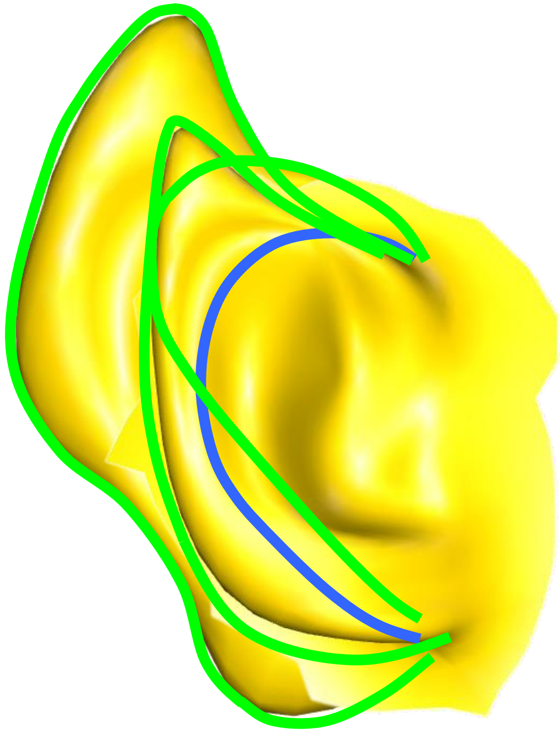
- **Sorkine et al. 2004**
  Laplacian Surface Editing
  https://igl.ethz.ch/projects/Laplacian-mesh-processing/Laplacian-mesh-editing/

Sketch based handles

Sketch-based Editing

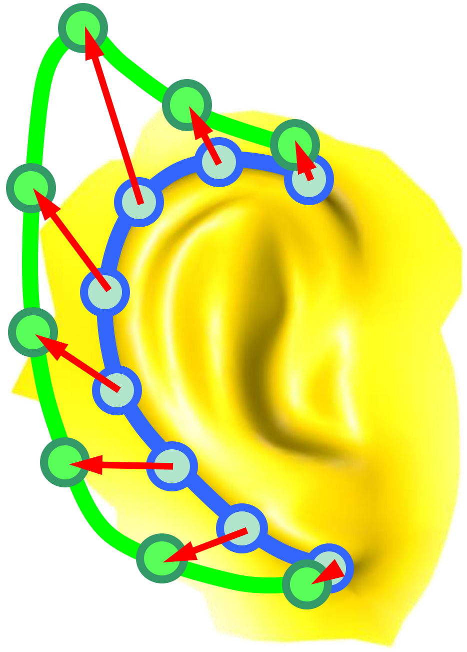
Nealen et al. 2005

- Can use curves as handles in e.g. Laplacian Editing
- Silhouettes are intuitive curves to reshape



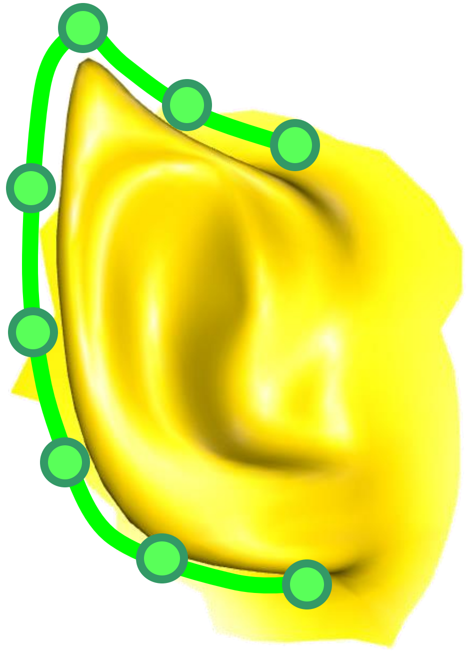
Silhouette Sketching

- Using silhouettes as handles
  - Detect object space silhouette
  - Project to screen space and parameterize [0,1]
  - Parametrize sketch [0,1]
  - Find correspondences



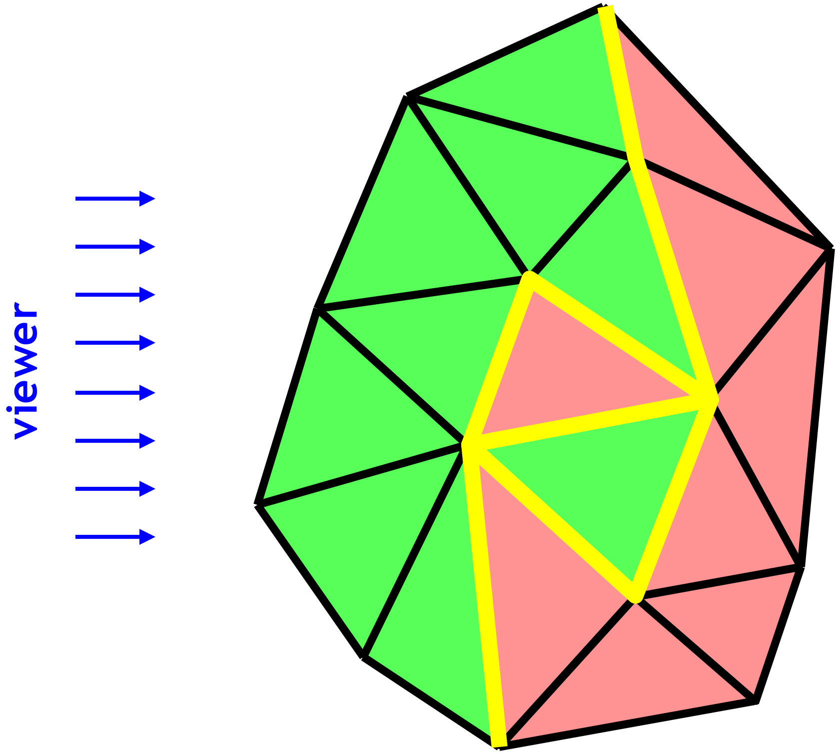
Silhouette Sketching

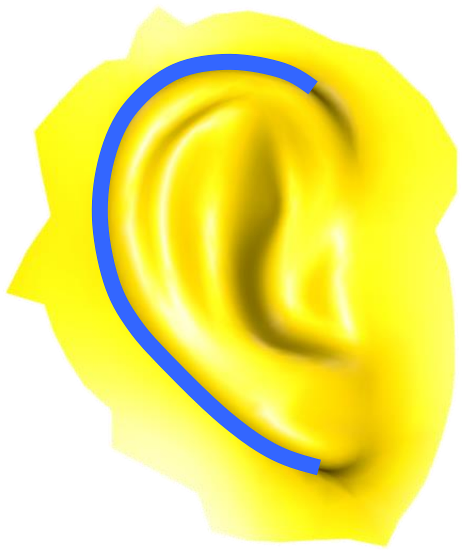
- Using silhouettes as handles
  - Detect object space silhouette
  - Project to screen space and parameterize [0,1]
  - Parametrize sketch [0,1]
  - Find correspondences
  - Use as positional constraints while retaining depth value



Silhouette Sketching

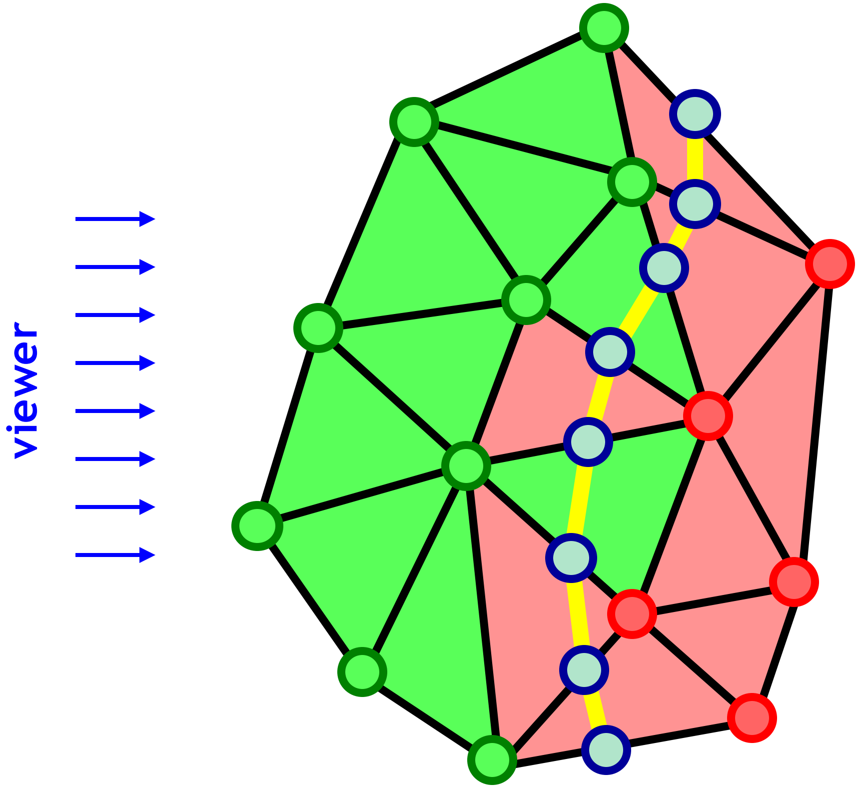
- What is a good silhouette?

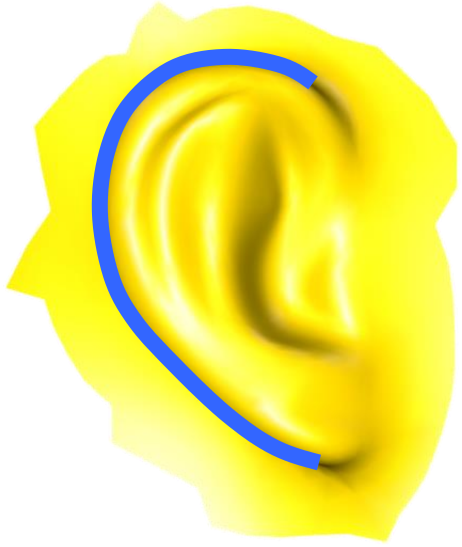




Silhouette Sketching

- Positional constraints on linear combinations of mesh vertices

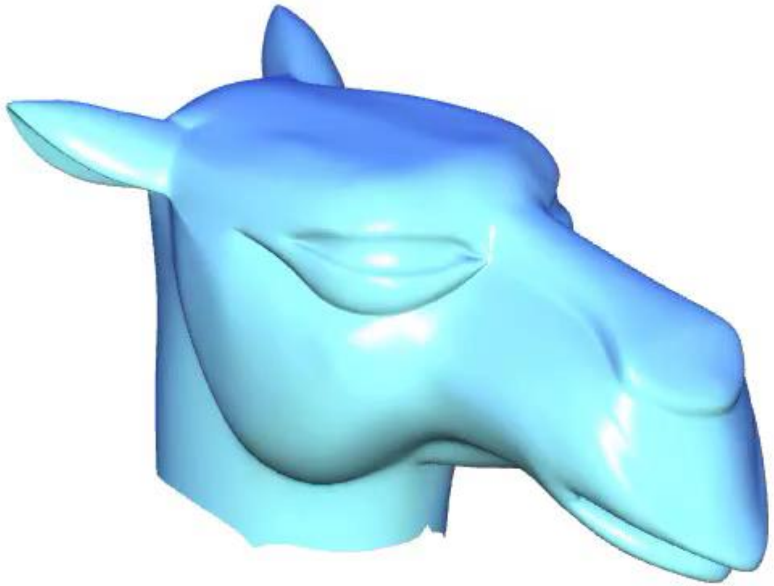




Illustrating Smooth Surfaces [Hertzmann and Zorin 00]

Approximate Sketching

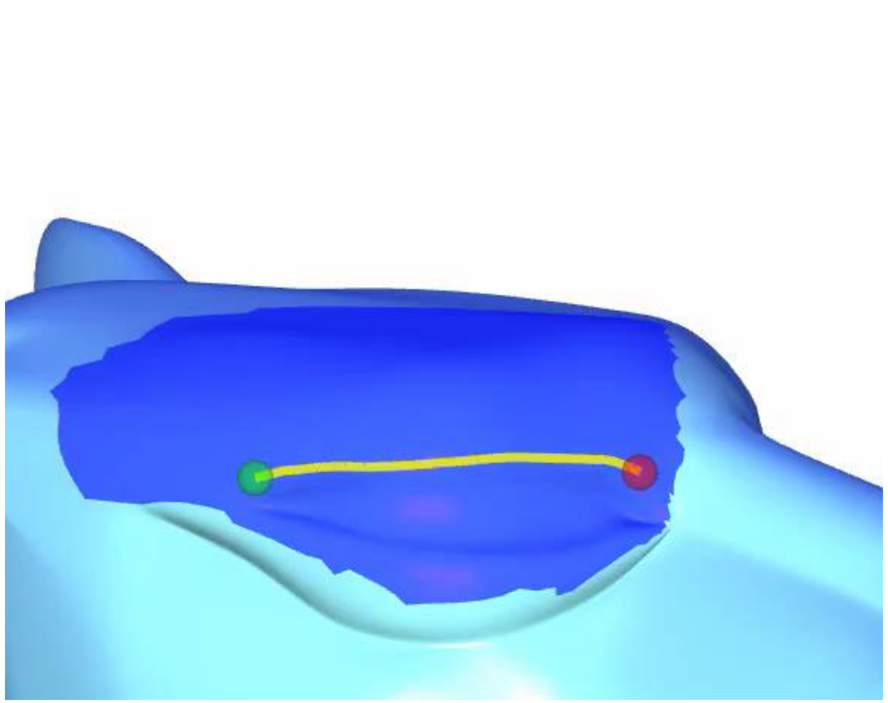
- Use *soft* positional constraints



http://youtu.be/EMx6yNe23ug

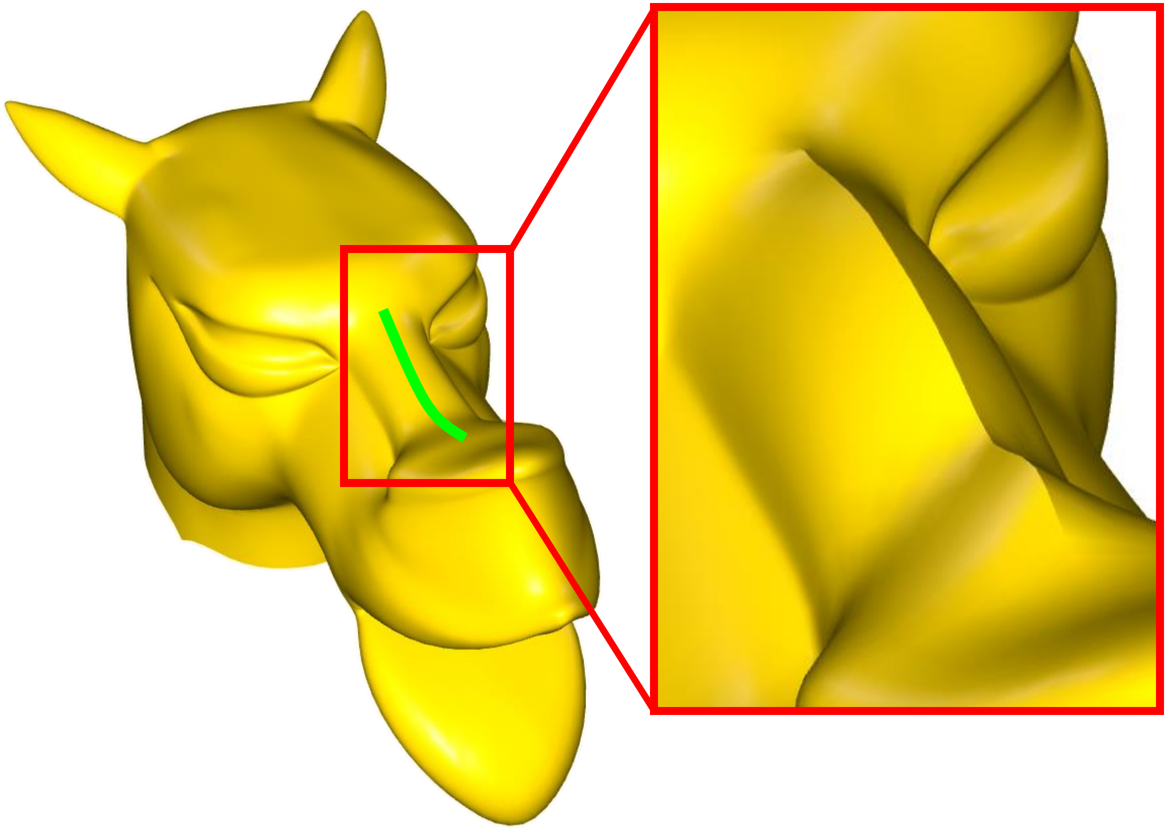
Approximate Sketching

- Use *soft* positional constraints



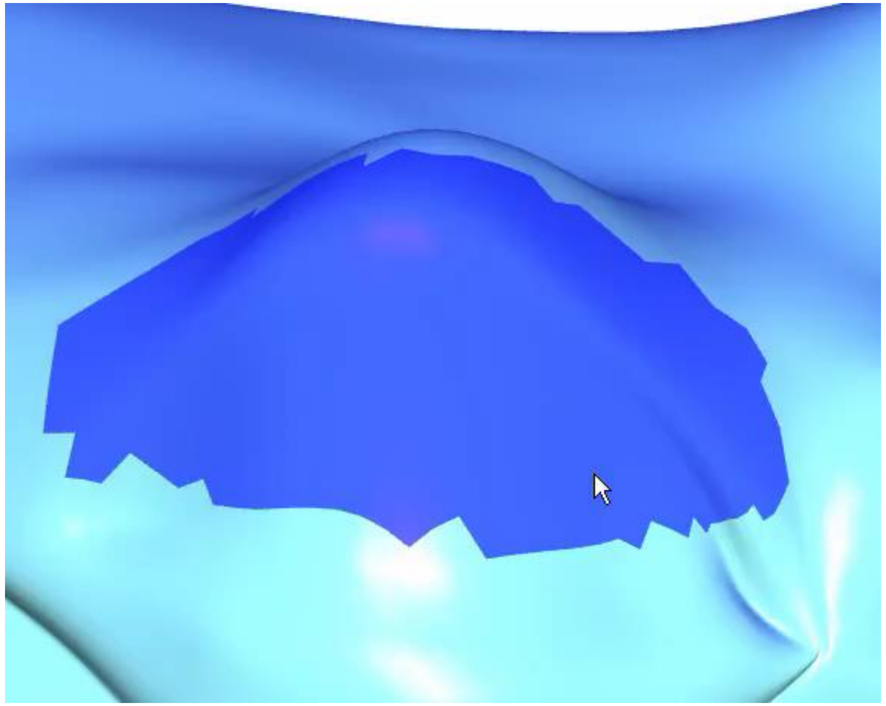
Feature Sketching

- Length of Laplacians is proportional to mean curvature
  - Curvature can be prescribed
  - Application: sketching ridges and ravines



Feature Sketching

- Length of Laplacians is proportional to mean curvature
  - Curvature can be prescribed
  - Application: sketching ridges and ravines



252-0538-00L, Spring 2025

Shape Modeling and Geometry Processing

As-Rigid-As-Possible Surface Modeling

https://igl.ethz.ch/projects/ARAP/

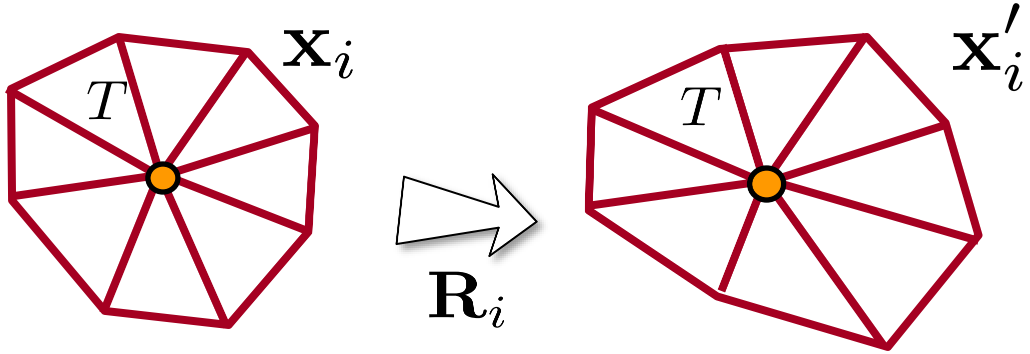
INTERACTIVE GEOMETRY LAB

May 7, 2025

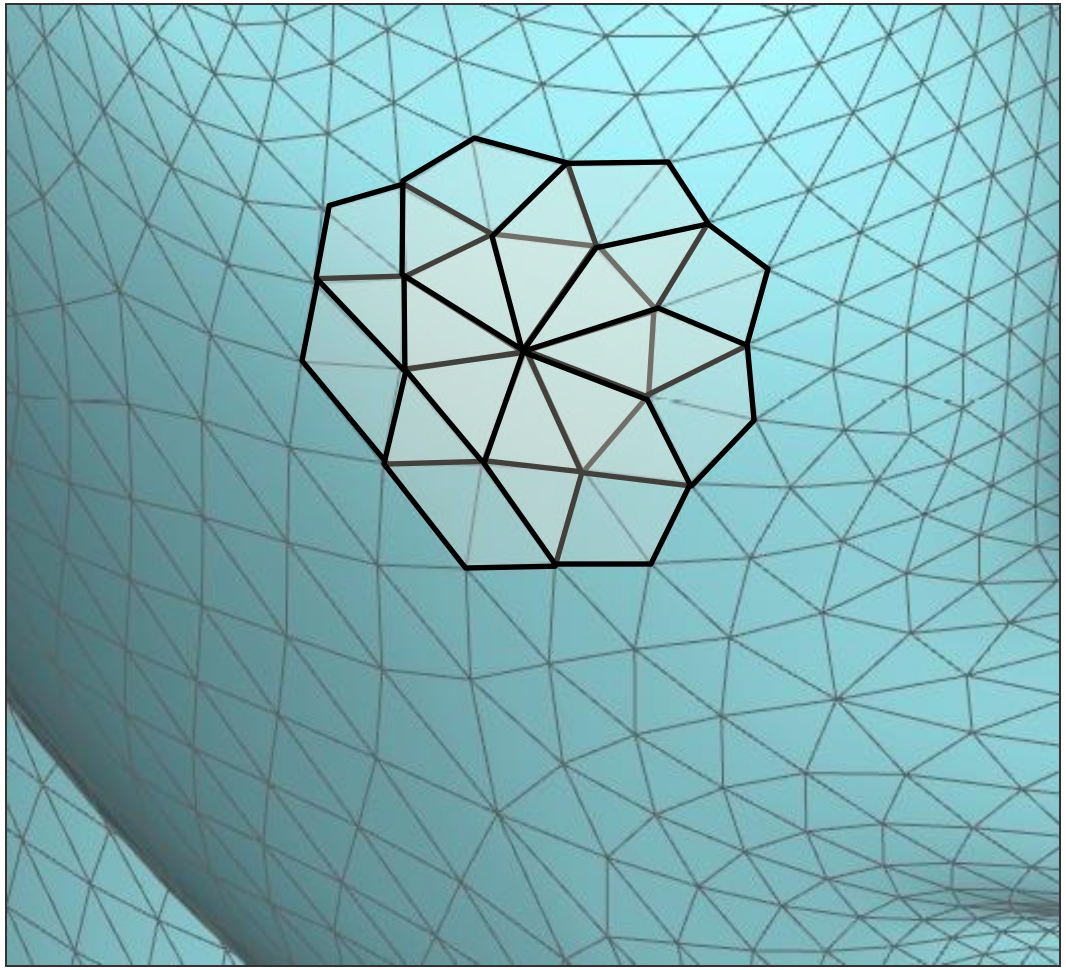
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

As-Rigid-As-Possible Deformation

- Preserve shape of cells covering the surface
- Ask each cell i to transform **rigidly** by best-fitting rotation $\mathbf{R}_i$

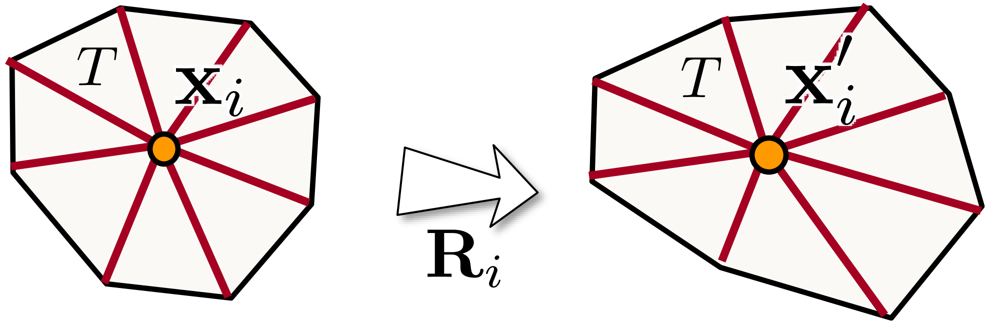


$$\min \sum_{T \in \text{Cell}_i} \sum_{(j,k) \in T} \|(\mathbf{x}'_j - \mathbf{x}'_k) - \mathbf{R}_i(\mathbf{x}_j - \mathbf{x}_k)\|^2$$



As-Rigid-As-Possible Deformation

- Optimal $\mathbf{R}_i$ is uniquely defined by the cells of $\mathbf{x}_i,\ \mathbf{x}_i'$

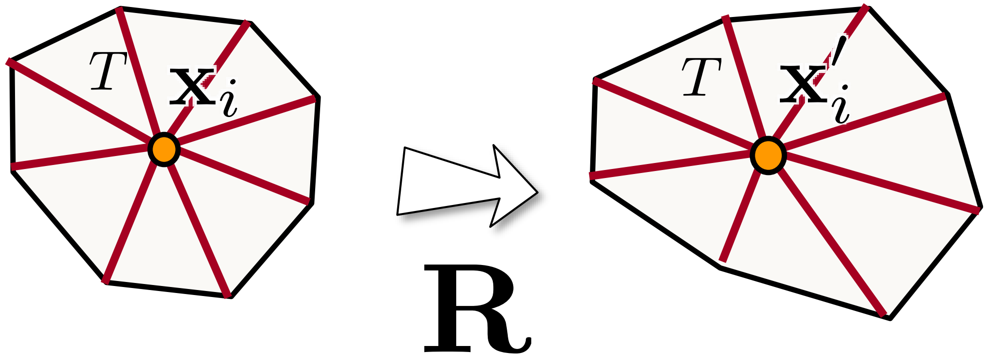


$$\mathbf{R}_i = \operatorname*{argmin}_{\mathbf{R}_i \in SO(3)} \sum_{T \in \mathrm{Cell}_i} \sum_{(j,k) \in T} \|(\mathbf{x}_j' - \mathbf{x}_k') - \mathbf{R}_i(\mathbf{x}_j - \mathbf{x}_k)\|^2$$

- so-called shape-matching or Procrustes problem, solved e.g. by a 3x3 SVD

$\mathbf{R}_i$ is a nonlinear function of $\mathbf{x}'$

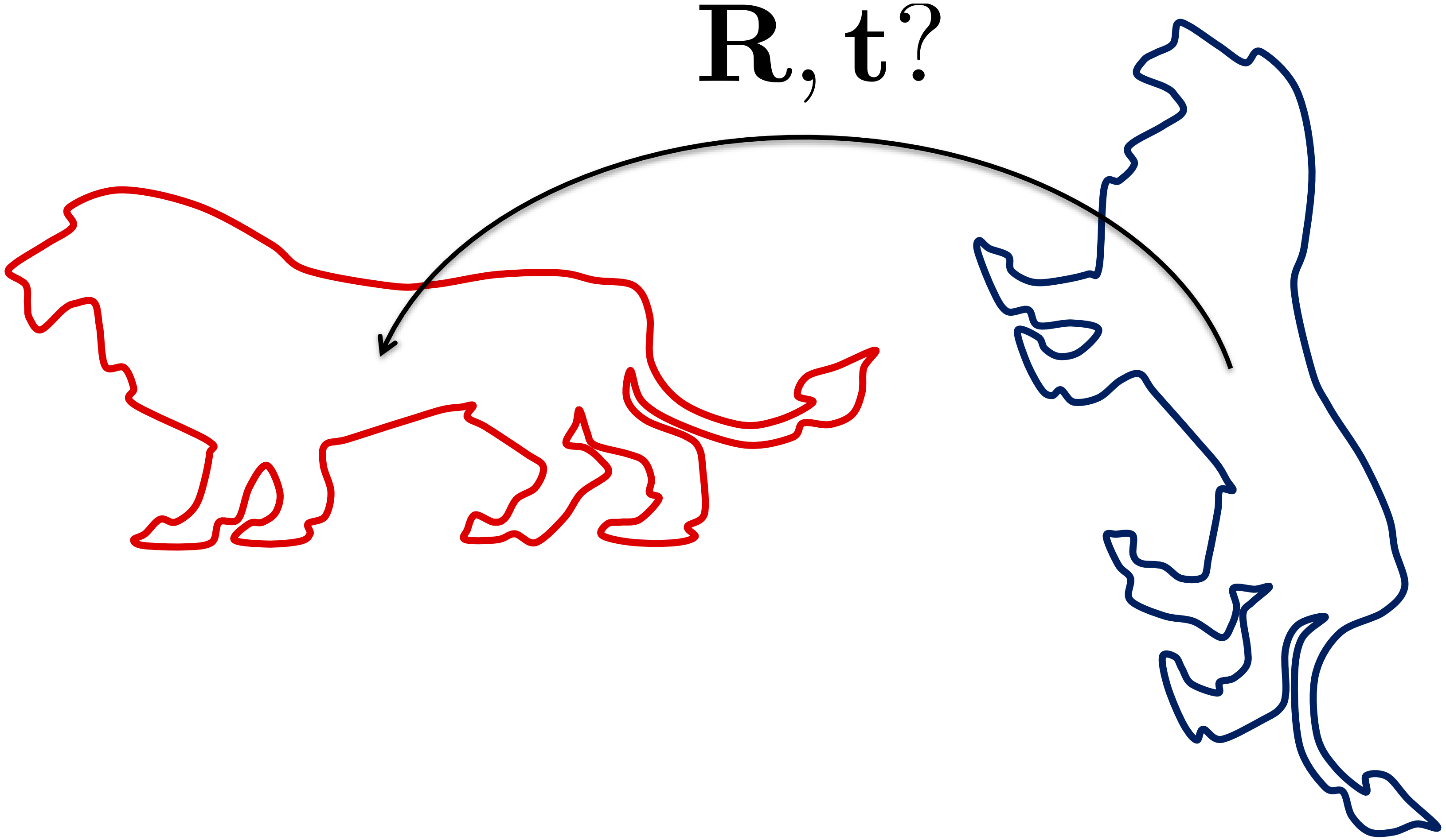
Optimal Rotation



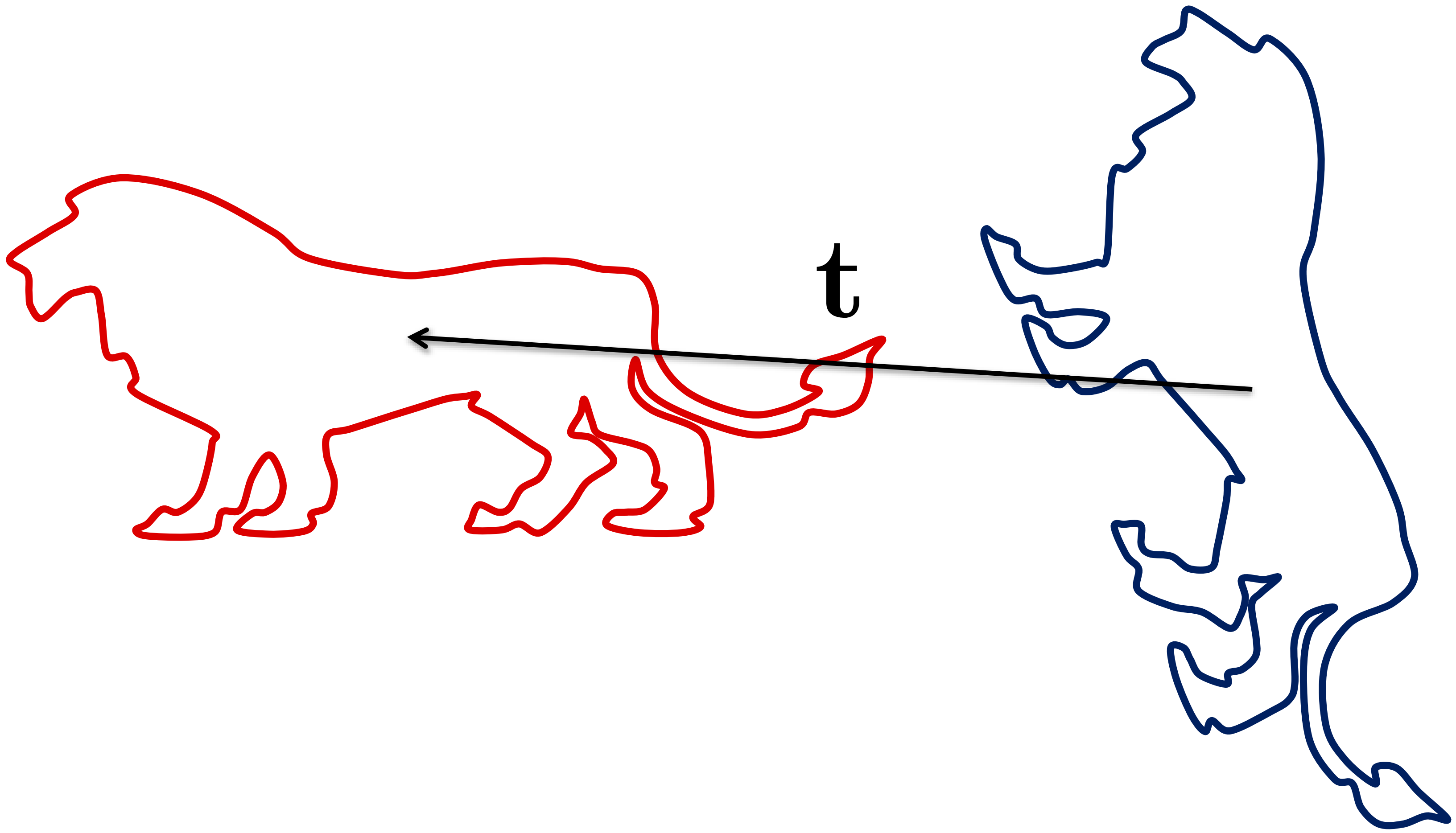
$$\min_{\mathbf{R}\in SO(3)} \sum_{T\in \text{Cell}_i} \sum_{(j,k)\in T} \left\|(\mathbf{x}'_j - \mathbf{x}'_k) - \mathbf{R}(\mathbf{x}_j - \mathbf{x}_k)\right\|^2$$

Rotation group

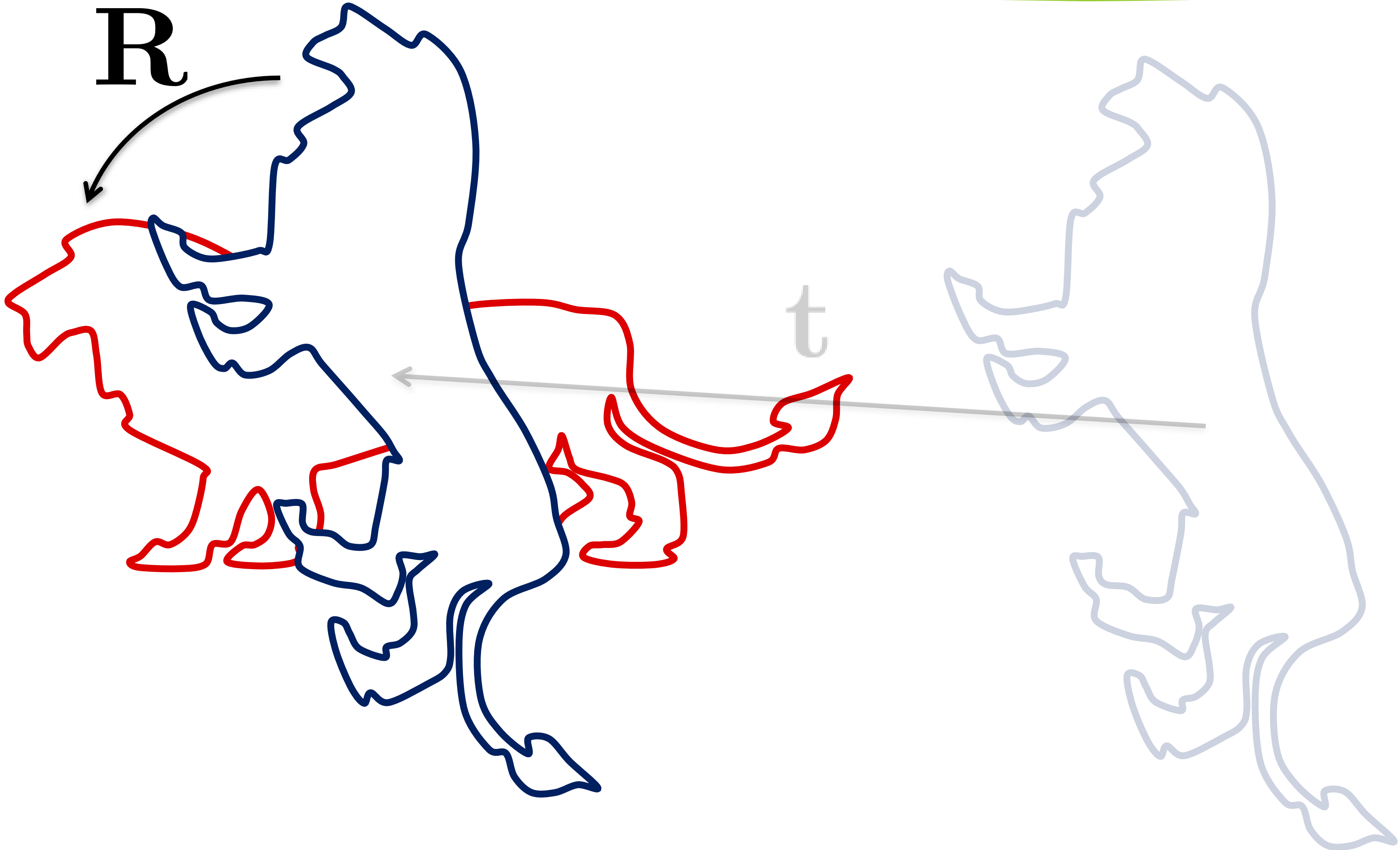
Shape Matching Problem



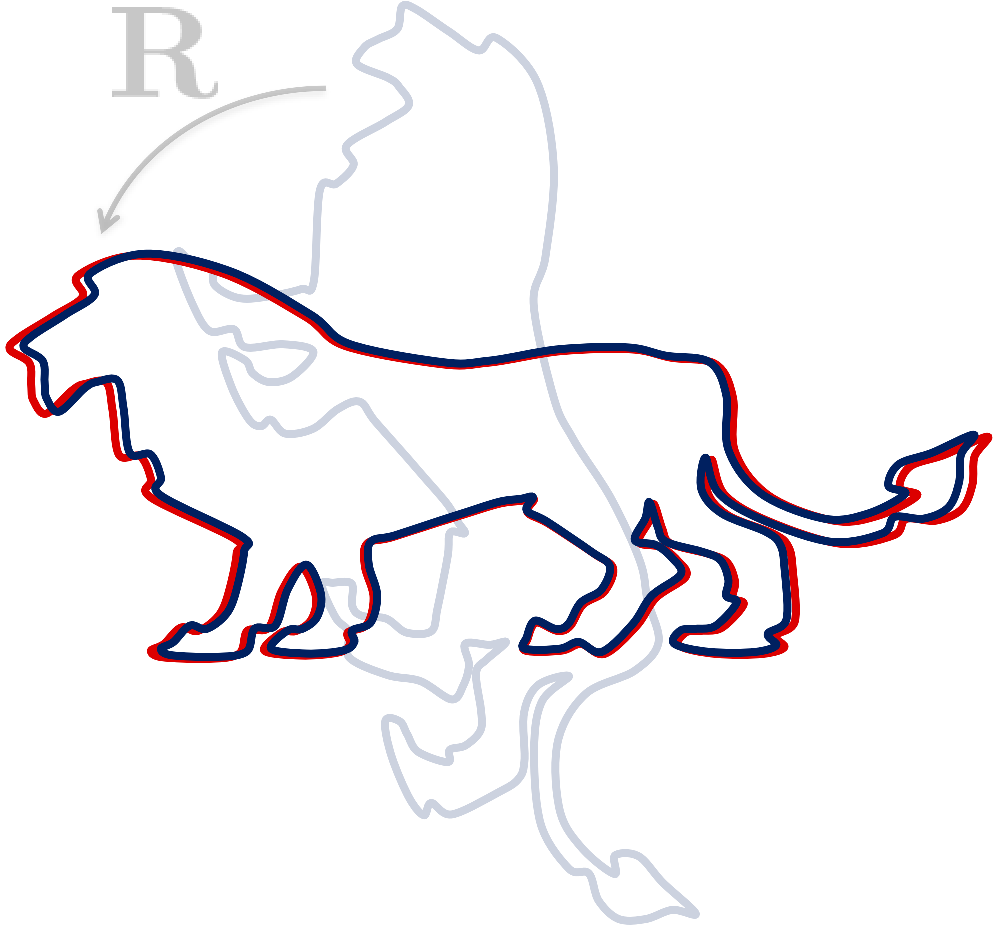
Shape Matching Problem



Shape Matching Problem



Shape Matching Problem



Shape Matching Problem

- Align two point sets ($\mathbf{q}_i$ corresponds to $\mathbf{p}_i$)*

$$\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \text{ and } \mathcal{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$$

- Find a translation vector $\mathbf{t}$ and rotation matrix $\mathbf{R}$ so that

$$\sum_{i=1}^{n} \|(\mathbf{R}\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i\|^2 \text{ is minimized}$$

* How to find the correspondence? Recall the 2nd lecture of the semester ☺

Shape Matching – Solution

- Solve for translation first (w.r.t. $\mathbf{R}$, $\mathbf{p}$, and $\mathbf{q}$)

$$\frac{\partial}{\partial \mathbf{t}} \sum_{i=1}^{n} \|(\mathbf{R}\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i\|^2 = \sum_{i=1}^{n} 2\left((\mathbf{R}\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i\right) \stackrel{!}{=} 0$$

$$\mathbf{R}\sum_{i=1}^{n} \mathbf{p}_i + \sum_{i=1}^{n} \mathbf{t} - \sum_{i=1}^{n} \mathbf{q}_i = 0$$

$$\mathbf{t} = \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{q}_i\right)}_{\bar{\mathbf{q}}} - \mathbf{R}\underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_i\right)}_{\bar{\mathbf{p}}}$$

Take a look at the Matrix Cookbook!

Finding the Rotation $\mathbf{R}$

- To find the optimal $\mathbf{R}$, we bring the centroids of both point sets to the origin

$$\mathbf{v}_i = \mathbf{p}_i - \bar{\mathbf{p}}, \qquad \mathbf{v}'_i = \mathbf{q}_i - \bar{\mathbf{q}}$$

- We want to find $\mathbf{R}$ that minimizes

$$\sum_{i=1}^{n} \|\mathbf{R}\mathbf{v}_i - \mathbf{v}'_i\|^2$$

Finding the Rotation $\mathbf{R}$

$$\sum_{i=1}^{n} \|\mathbf{R}\mathbf{v}_i - \mathbf{v}'_i\|^2 = \sum_{i=1}^{n} (\mathbf{R}\mathbf{v}_i - \mathbf{v}'_i)^\mathsf{T} (\mathbf{R}\mathbf{v}_i - \mathbf{v}'_i) =$$

$$= \sum_{i=1}^{n} \left(\mathbf{v}_i^\mathsf{T} \underbrace{\mathbf{R}^\mathsf{T}\mathbf{R}}_{\mathbf{I}} \mathbf{v}_i - {\mathbf{v}'_i}^\mathsf{T}\mathbf{R}\mathbf{v}_i - \mathbf{v}_i^\mathsf{T}\mathbf{R}^\mathsf{T}\mathbf{v}'_i + {\mathbf{v}'_i}^\mathsf{T}\mathbf{v}'_i \right)^\mathsf{T}$$

These terms do not depend on $\mathbf{R}$,
so we can ignore them in the minimization

Finding the Rotation $\mathbf{R}$

$$\underset{\mathbf{R}\in SO(3)}{\operatorname{argmin}} \sum_{i=1}^{n} \left(-{\mathbf{v}_i'}^\mathsf{T} \mathbf{R} \mathbf{v}_i - \mathbf{v}_i^\mathsf{T} \mathbf{R}^\mathsf{T} \mathbf{v}_i' \right) = \underset{\mathbf{R}\in SO(3)}{\operatorname{argmax}} \sum_{i=1}^{n} \left({\mathbf{v}_i'}^\mathsf{T} \mathbf{R} \mathbf{v}_i + \underbrace{\mathbf{v}_i^\mathsf{T} \mathbf{R}^\mathsf{T} \mathbf{v}_i'} \right) =$$

$$= \boxed{\underset{\mathbf{R}\in SO(3)}{\operatorname{argmax}} \sum_{i=1}^{n} {\mathbf{v}_i'}^\mathsf{T} \mathbf{R} \mathbf{v}_i}$$

$$\mathbf{v}_i^\mathsf{T} \mathbf{R}^\mathsf{T} \mathbf{v}_i' = \left(\mathbf{v}_i^\mathsf{T} \mathbf{R}^\mathsf{T} \mathbf{v}_i' \right)^\mathsf{T} = {\mathbf{v}_i'}^\mathsf{T} \mathbf{R} \mathbf{v}_i$$

Finding the Rotation $\mathbf{R}$

$$\sum_{i=1}^{n} \mathbf{v}_i'^{\mathsf{T}} \mathbf{R} \mathbf{v}_i = tr\left(\mathbf{V}'^{\mathsf{T}} \mathbf{R} \mathbf{V}\right)$$

$$\underbrace{\begin{bmatrix} \mathbf{v}_1'^{\mathsf{T}} \\ \mathbf{v}_2'^{\mathsf{T}} \\ \vdots \\ \mathbf{v}_n'^{\mathsf{T}} \end{bmatrix}}_{\mathbf{V}'^{\mathsf{T}}} \begin{bmatrix} \mathbf{R} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{bmatrix}}_{\mathbf{V}} = \begin{bmatrix} \mathbf{v}_1'^{\mathsf{T}} \\ \mathbf{v}_2'^{\mathsf{T}} \\ \vdots \\ \mathbf{v}_n'^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \mathbf{R}\mathbf{v}_1 & \mathbf{R}\mathbf{v}_2 & \cdots & \mathbf{R}\mathbf{v}_n \end{bmatrix}$$

Finding the Rotation $\mathbf{R}$

$$\sum_{i=1}^{n} \mathbf{v}_i'^{\top} \mathbf{R} \mathbf{v}_i = tr\left(\mathbf{V}'^{\top} \mathbf{R} \mathbf{V}\right)$$

$$\begin{bmatrix} \mathbf{v}_1'^{\top} \\ \mathbf{v}_2'^{\top} \\ \vdots \\ \mathbf{v}_n'^{\top} \end{bmatrix} \begin{bmatrix} \mathbf{R}\mathbf{v}_1 & \mathbf{R}\mathbf{v}_2 & \cdots & \mathbf{R}\mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1'^{\top}\mathbf{R}\mathbf{v}_1 & & & \\ & \mathbf{v}_2'^{\top}\mathbf{R}\mathbf{v}_2 & & \\ & & \ddots & \\ & & & \mathbf{v}_n'^{\top}\mathbf{R}\mathbf{v}_n \end{bmatrix}$$

Finding the Rotation $\mathbf{R}$

Take a look at the Matrix Cookbook!

- Find $\mathbf{R}$ that maximizes

$$tr\left(\mathbf{V'}^{\mathsf{T}}\mathbf{R}\mathbf{V}\right) = tr\left(\mathbf{R}\mathbf{V}\mathbf{V'}^{\mathsf{T}}\right)$$

- SVD: $\mathbf{V}\mathbf{V'}^{\mathsf{T}} = \mathbf{U}\mathbf{\Sigma}\tilde{\mathbf{U}}^{\mathsf{T}}$

$$tr\left(\mathbf{R}\mathbf{V}\mathbf{V'}^{\mathsf{T}}\right) = tr\left(\underbrace{\mathbf{R}\,\mathbf{U}}\underbrace{\mathbf{\Sigma}\tilde{\mathbf{U}}^{\mathsf{T}}}\right) = tr\left(\mathbf{\Sigma}\,\underbrace{\tilde{\mathbf{U}}^{\mathsf{T}}\mathbf{R}\mathbf{U}}_{\text{orthogonal matrix}}\right)$$

Finding the Rotation $\mathbf{R}$

- We want to maximize

$$tr\left(\Sigma\,\mathbf{M}\right)$$

$\mathbf{M}$: orthogonal matrix
all coeffs ≤ 1

$$\begin{pmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{pmatrix} \begin{pmatrix} m_{11} & \cdots & \\ \vdots & m_{22} & \vdots \\ & \cdots & m_{33} \end{pmatrix}$$

$$tr\left(\Sigma\,\mathbf{M}\right) = \sum_{i=1}^{3} \sigma_i\, m_{ii} \leq \sum_{i=1}^{3} \sigma_i$$

Finding the Rotation $\mathbf{R}$

$$tr\,(\Sigma\,\mathbf{M}) = \sum_{i=1}^{3} \sigma_i\, m_{ii} \leq \sum_{i=1}^{3} \sigma_i$$

- Our best shot is $m_{ii} = 1$, i.e. to make $\mathbf{M} = \mathbf{I}$

$$\begin{aligned} \mathbf{M} = \tilde{\mathbf{U}}^\mathsf{T}\mathbf{R}\mathbf{U} &\overset{!}{=} \mathbf{I} \\ \mathbf{R}\mathbf{U} &= \tilde{\mathbf{U}} \\ \boxed{\mathbf{R} = \tilde{\mathbf{U}}\mathbf{U}^\mathsf{T}} \end{aligned}$$

Summary of Rigid Alignment

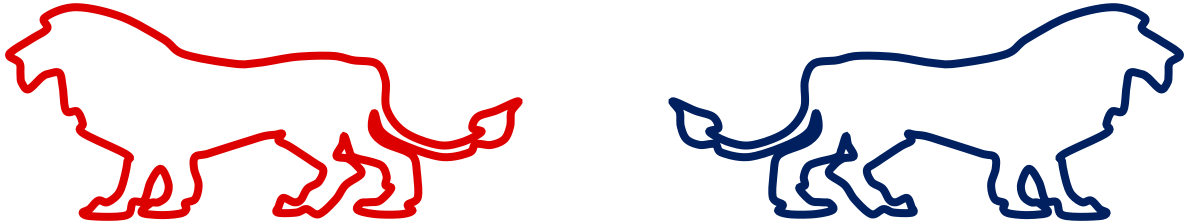
- Translate the input points to the centroids

$$\mathbf{v}_i = \mathbf{p}_i - \bar{\mathbf{p}}, \qquad \mathbf{v}_i' = \mathbf{q}_i - \bar{\mathbf{q}}$$

- Compute the "covariance matrix" $\mathbf{V}\mathbf{V}'^{\mathsf{T}}$
- Compute its SVD: $\mathbf{V}\mathbf{V}'^{\mathsf{T}} = \mathbf{U}\boldsymbol{\Sigma}\tilde{\mathbf{U}}^{\mathsf{T}}$
- The optimal orthogonal $\mathbf{R}$ is $\mathbf{R} = \tilde{\mathbf{U}}\mathbf{U}^{\mathsf{T}}$

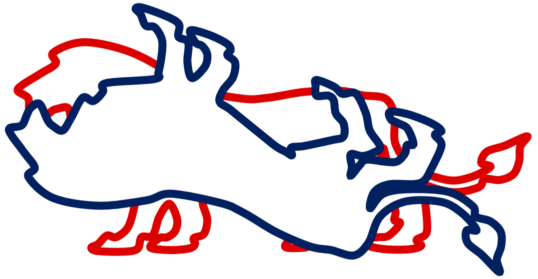
Sign Correction

- It is possible that $\det(\tilde{\mathbf{U}}\mathbf{U}^\top) = -1$: sometimes reflection is the best orthogonal transform



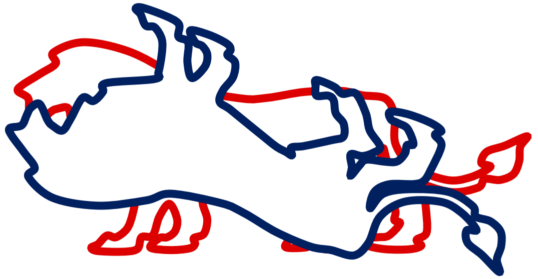
Sign Correction

- It is possible that $\det(\tilde{\mathbf{U}}\mathbf{U}^\top) = -1$: sometimes reflection is the best orthogonal transform



Sign Correction

- To restrict ourselves to rotations only:
  take the last column of $\mathbf{U}$ (corresponding to the smallest singular value) and invert its sign.



- Why? See http://igl.ethz.ch/projects/ARAP/svd_rot.pdf

Summary of Best Fit Rotation

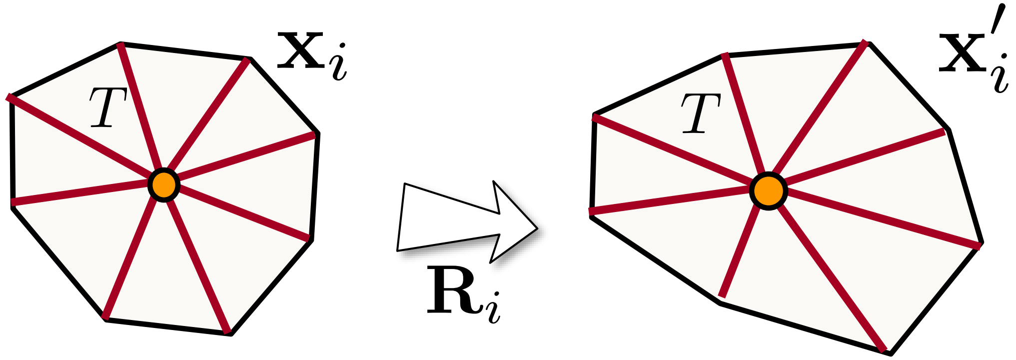
- Translate the input points to the centroids

$$\mathbf{v}_i = \mathbf{p}_i - \bar{\mathbf{p}}, \qquad \mathbf{v}'_i = \mathbf{q}_i - \bar{\mathbf{q}}$$

- Compute the "covariance matrix" $\mathbf{V}\mathbf{V}'^{\mathsf{T}}$
- Compute its SVD: $\mathbf{V}\mathbf{V}'^{\mathsf{T}} = \mathbf{U}\boldsymbol{\Sigma}\tilde{\mathbf{U}}^{\mathsf{T}}$
- The optimal rotation $\mathbf{R}$ is $\mathbf{R} = \tilde{\mathbf{U}} \begin{pmatrix} 1 & & \\ & 1 & \\ & & \det(\tilde{\mathbf{U}}\mathbf{U}^{\mathsf{T}}) \end{pmatrix} \mathbf{U}^{\mathsf{T}}$

As-Rigid-As-Possible Deformation

- Optimal $\mathbf{R}_i$ is uniquely defined by $\mathbf{x}_i$, $\mathbf{x}'_i$



$$\min \sum_{T \in \mathrm{Cell}_i} \sum_{(j,k) \in T} \|(\mathbf{x}'_j - \mathbf{x}'_k) - \mathbf{R}_i(\mathbf{x}_j - \mathbf{x}_k)\|^2$$

- so-called shape-matching problem, solved by a 3x3 SVD

$\mathbf{R}_i$ is a nonlinear function of $\mathbf{x}$'

As-Rigid-As-Possible Deformation

- Total ARAP energy: sum up for all the cells i

$$\sum_i \sum_{T \in \text{Cell}_i} \sum_{(j,k) \in T} \left\| (\mathbf{x}'_j - \mathbf{x}'_k) - \mathbf{R}_i (\mathbf{x}_j - \mathbf{x}_k) \right\|^2$$

- Treat $\mathbf{x}$' and $\mathbf{R}$ as separate sets of variables
- Simple **local-global** iterative optimization process
  - Decreases the energy at each step

As-Rigid-As-Possible Deformation

- Total ARAP energy: sum up for all the cells i

$$\sum_{i} \sum_{T \in \mathrm{Cell}_i} \sum_{(j,k) \in T} \left\| (\mathbf{x}'_j - \mathbf{x}'_k) - \mathbf{R}_i (\mathbf{x}_j - \mathbf{x}_k) \right\|^2$$

- Local step: keep $\mathbf{x}'$ fixed, find optimal $\mathbf{R}_i$ per cell i
- Global step: keep $\mathbf{R}_i$ fixed, solve for $\mathbf{x}'$ - quadratic minimization problem ➡ $\mathbf{L}\mathbf{x}' = \mathbf{b}$

As-Rigid-As-Possible Deformation

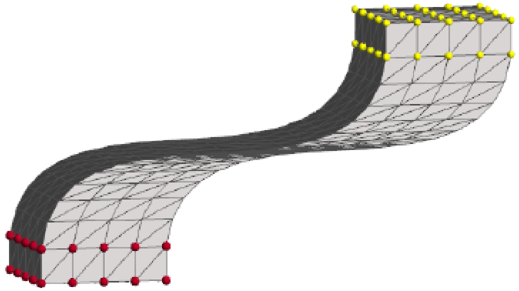
- Total ARAP energy: sum up for all the cells i

$$\sum_i \sum_{T \in \mathrm{Cell}_i} \sum_{(j,k) \in T} \left\| (\mathbf{x}'_j - \mathbf{x}'_k) - \mathbf{R}_i(\mathbf{x}_j - \mathbf{x}_k) \right\|^2$$

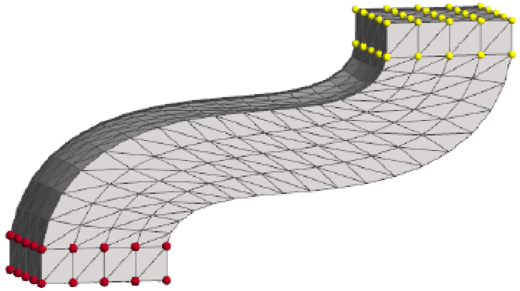
- Local step: keep $\mathbf{x}'$ fixed, find optimal $\mathbf{R}_i$ per cell i
- Global step: keep $\mathbf{R}_i$ fixed, solve for $\mathbf{x}'$ - quadratic minimization problem ➡ $\mathbf{L}\mathbf{x}' = \mathbf{b}$
  - The matrix $\mathbf{L}$ stays fixed, can pre-factorize

Initial Guess

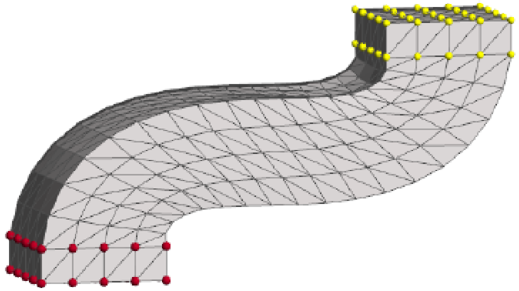
- Can use naïve Laplacian editing



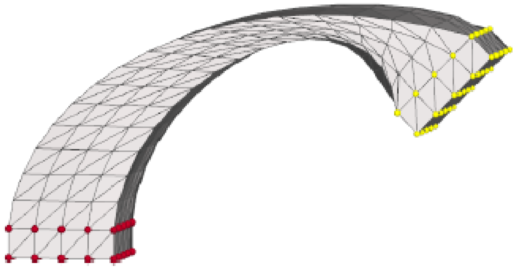
initial guess



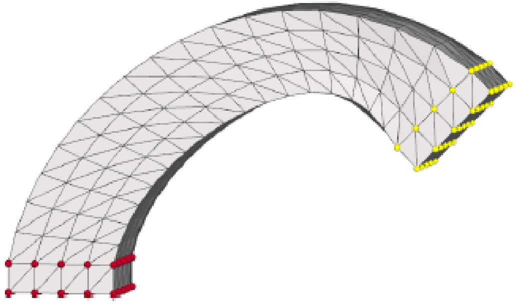
1 iteration



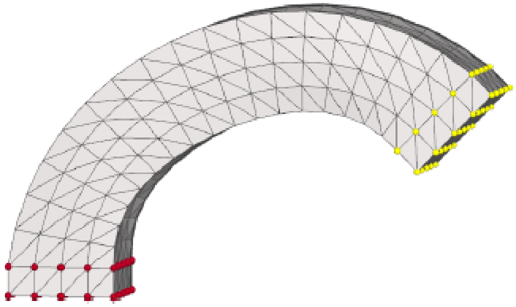
2 iterations



initial guess



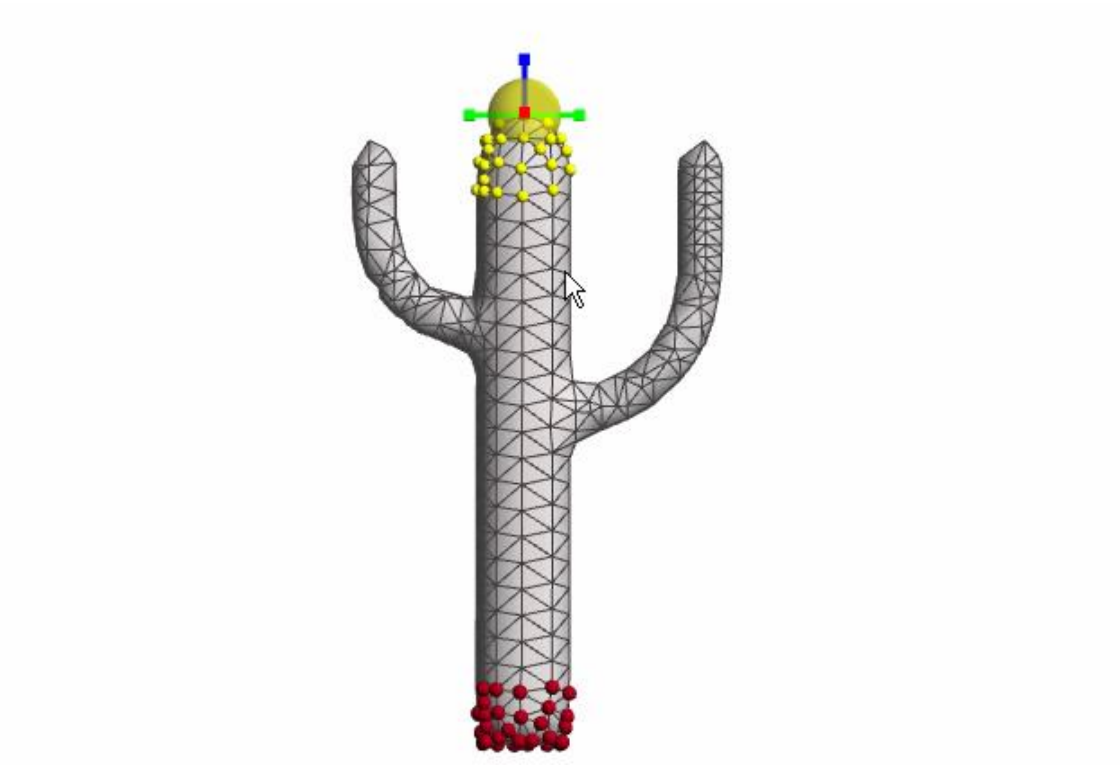
1 iterations



4 iterations

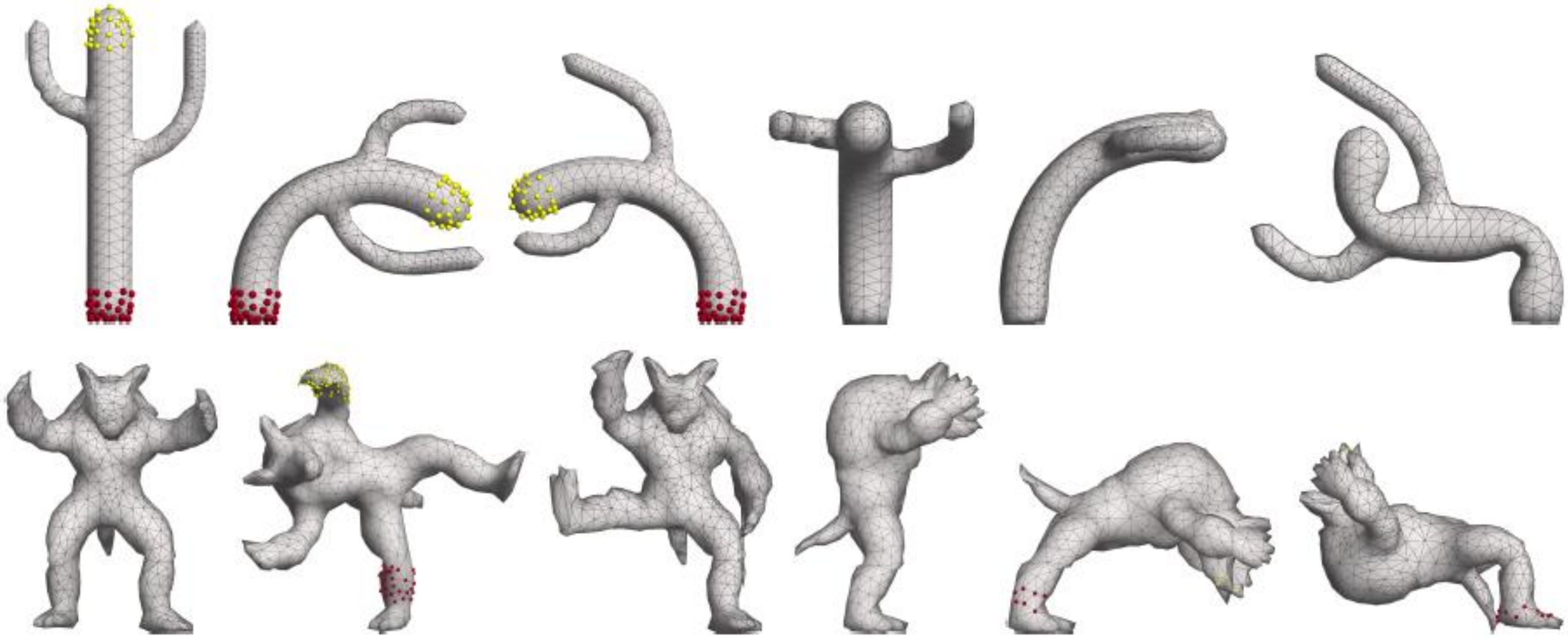
Initial Guess

- Can also use the previous frame
- Replace all handle vertex positions by the currently prescribed ones
- Fast convergence

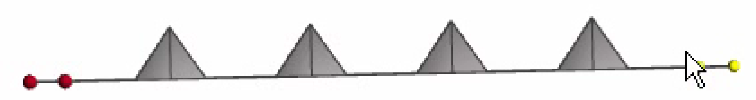


Large Rotations

- Use previous frame as the initial guess



Examples



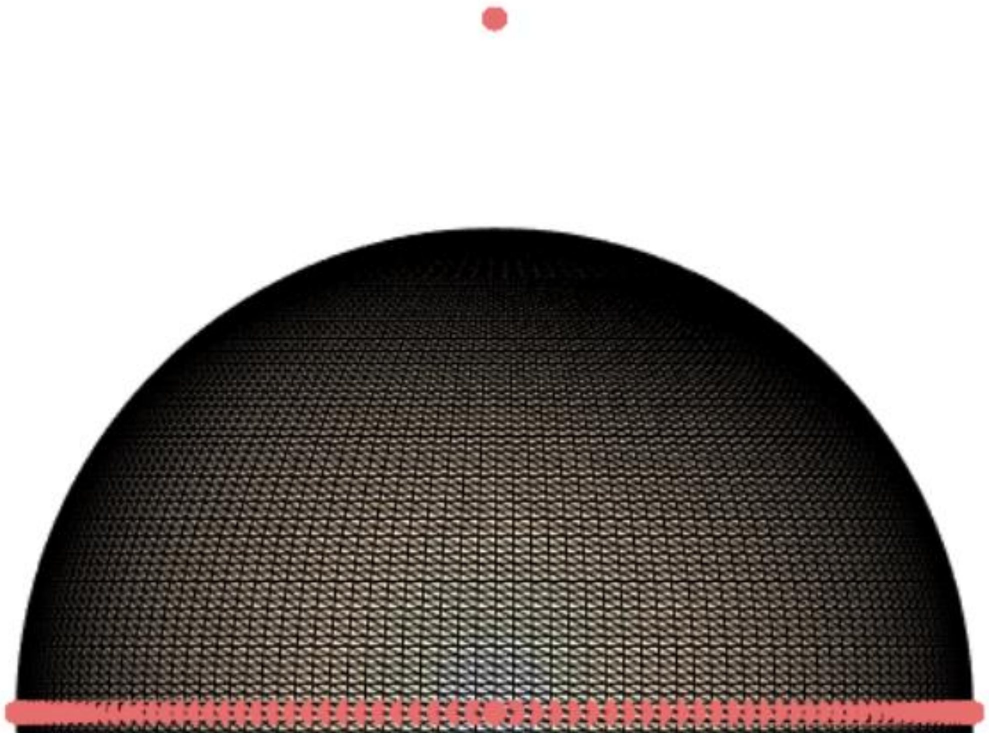
Discussion

- Nonlinear deformation that models a kind of elastic behavior
- Very simple to implement, no parameters to tune except number of iterations
- Each step is guaranteed to not increase the energy
  - Compare with Gauss-Newton…
- Each iteration is relatively cheap, no matrix re-factorization necessary

Discussion

- Works fine on small meshes
- On larger meshes: slower convergence
  - Each iteration is more expensive
  - Need more iterations because the conditioning of the system becomes worse as the matrix grows
- Material stiffness depends on the cell size
- Smoothness issues at point handles

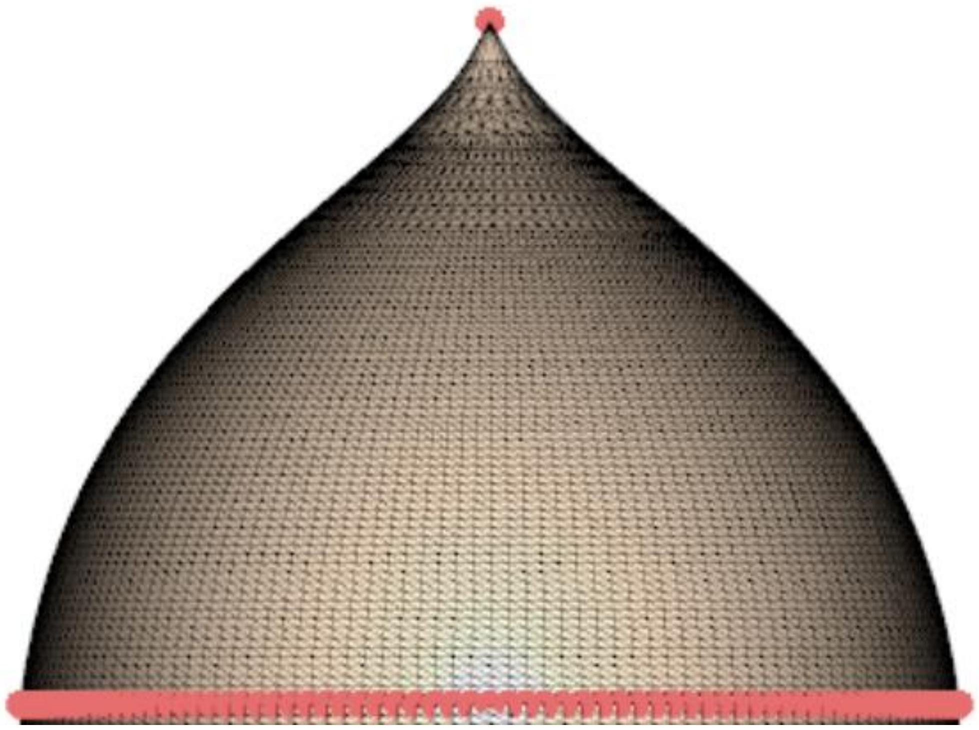
ARAP: Spikes at point handles



input

images from [Öhri et al. 2022]

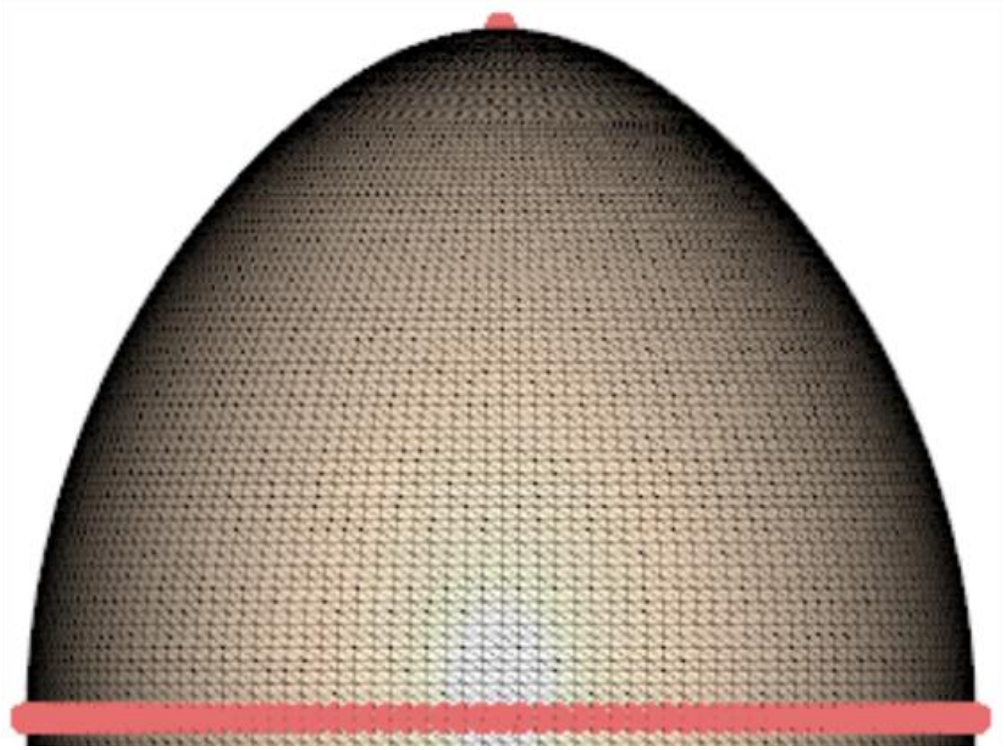
ARAP: Spikes at point handles



ARAP

images from [Öhri et al. 2022]

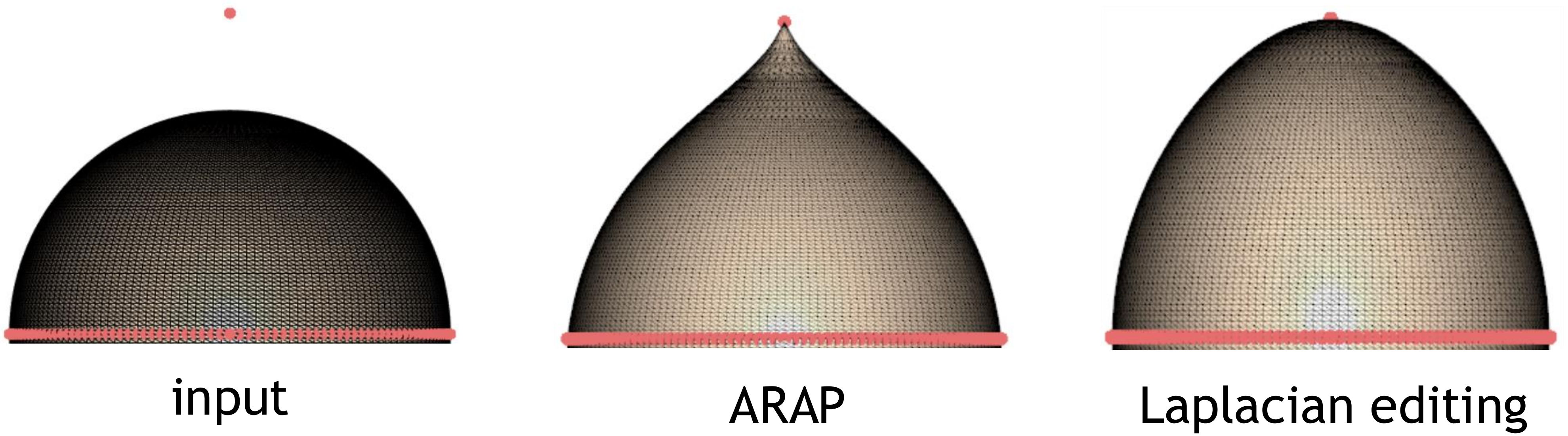
ARAP: Spikes at point handles



Laplacian editing

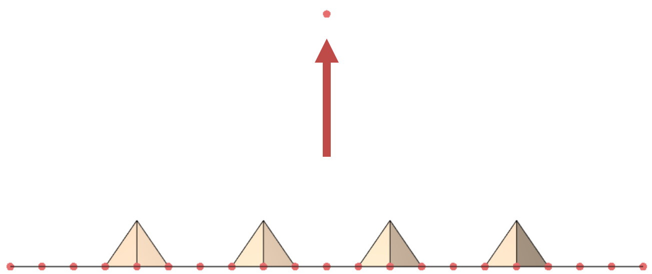
images from [Öhri et al. 2022]

ARAP: Spikes at point handles

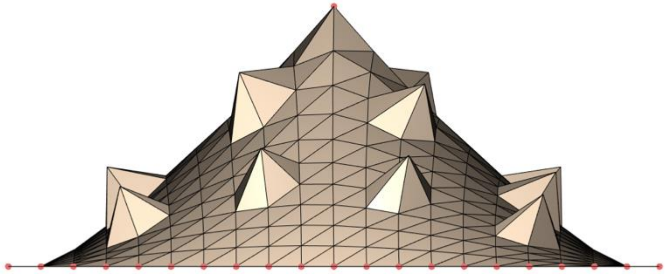


images from [Öhri et al. 2022]

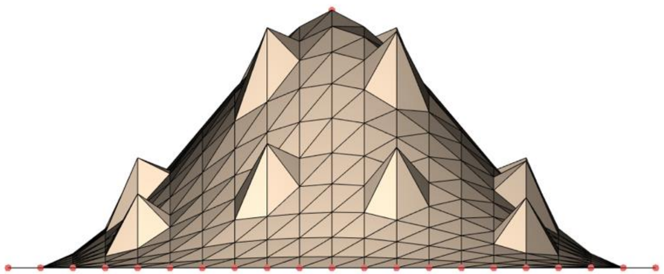
ARAP: Spikes at point handles



input



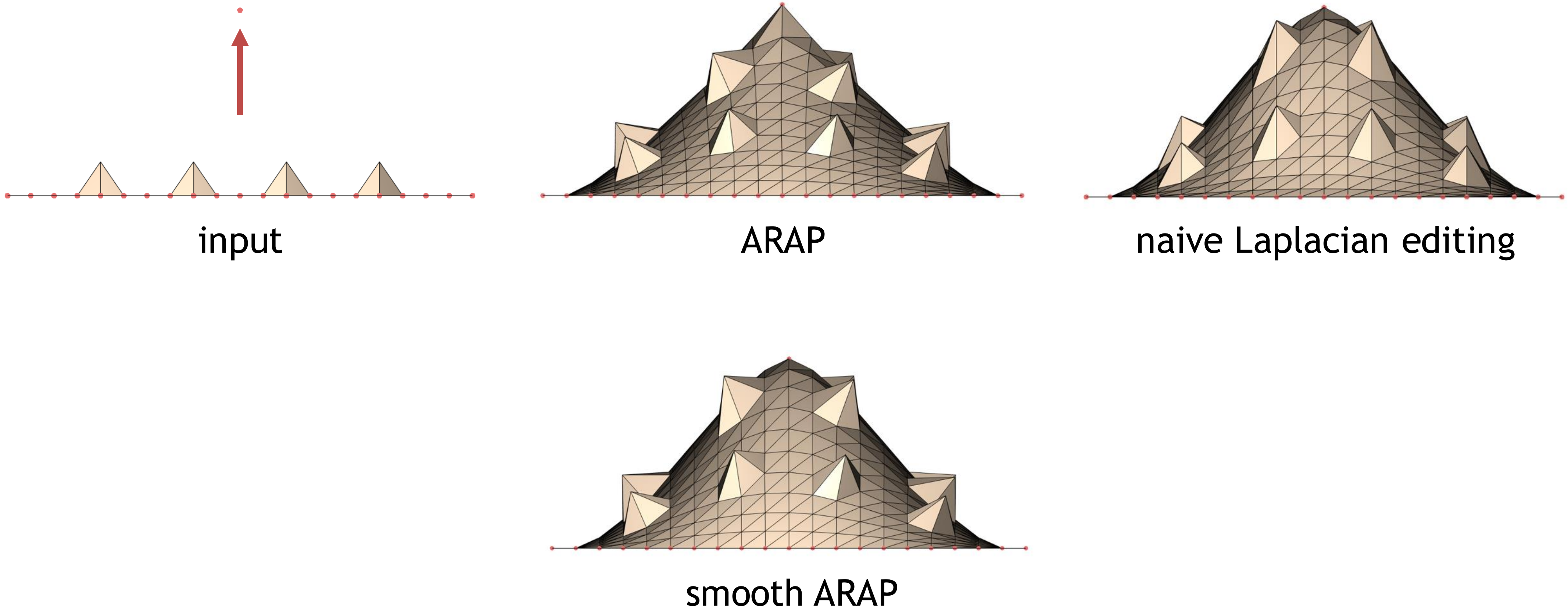
ARAP



naive Laplacian editing

images from [Öhri et al. 2022]

ARAP: Spikes at point handles



images from [Öhri et al. 2022]

As-Rigid-As-Possible Deformation

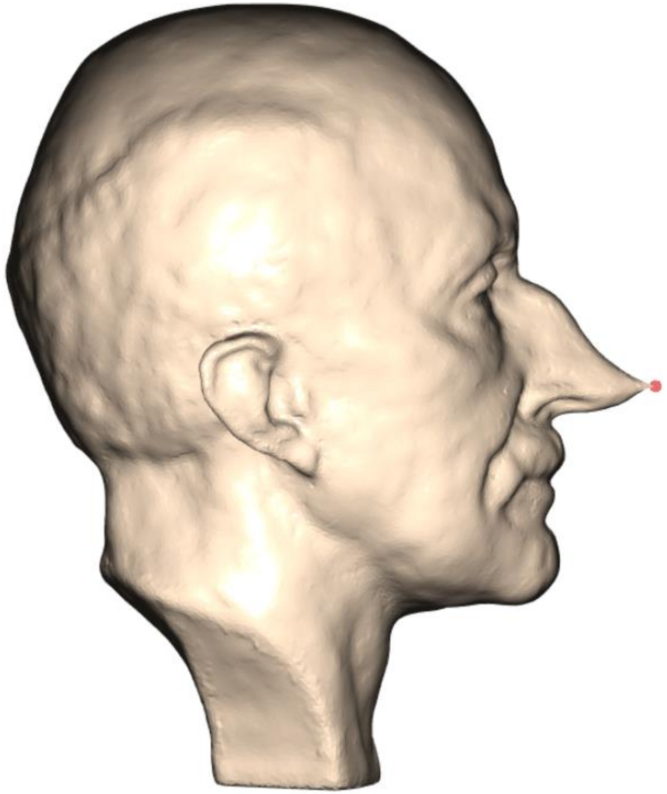
- Total ARAP energy: sum up for all the cells i

$$\sum_i \sum_{T \in \mathrm{Cell}_i} \sum_{(j,k) \in T} \|(\mathbf{x}'_j - \mathbf{x}'_k) - \mathbf{R}_i(\mathbf{x}_j - \mathbf{x}_k)\|^2$$

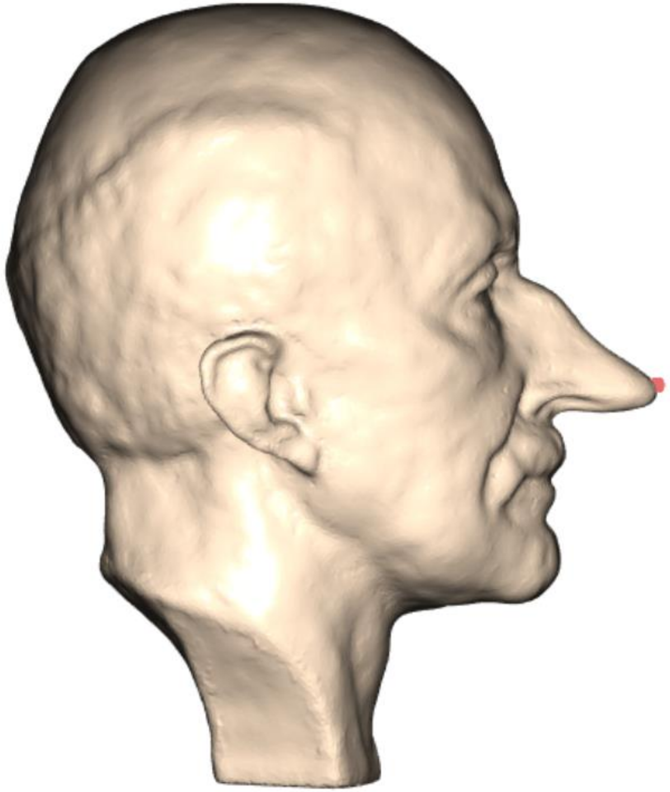
↓

$$E(\mathbf{x}') = \sum_{i=1}^{n} A_i \|\Delta(\mathbf{x}'_i) - R_i \delta_i\|^2$$

ARAP: Spikes at point handles



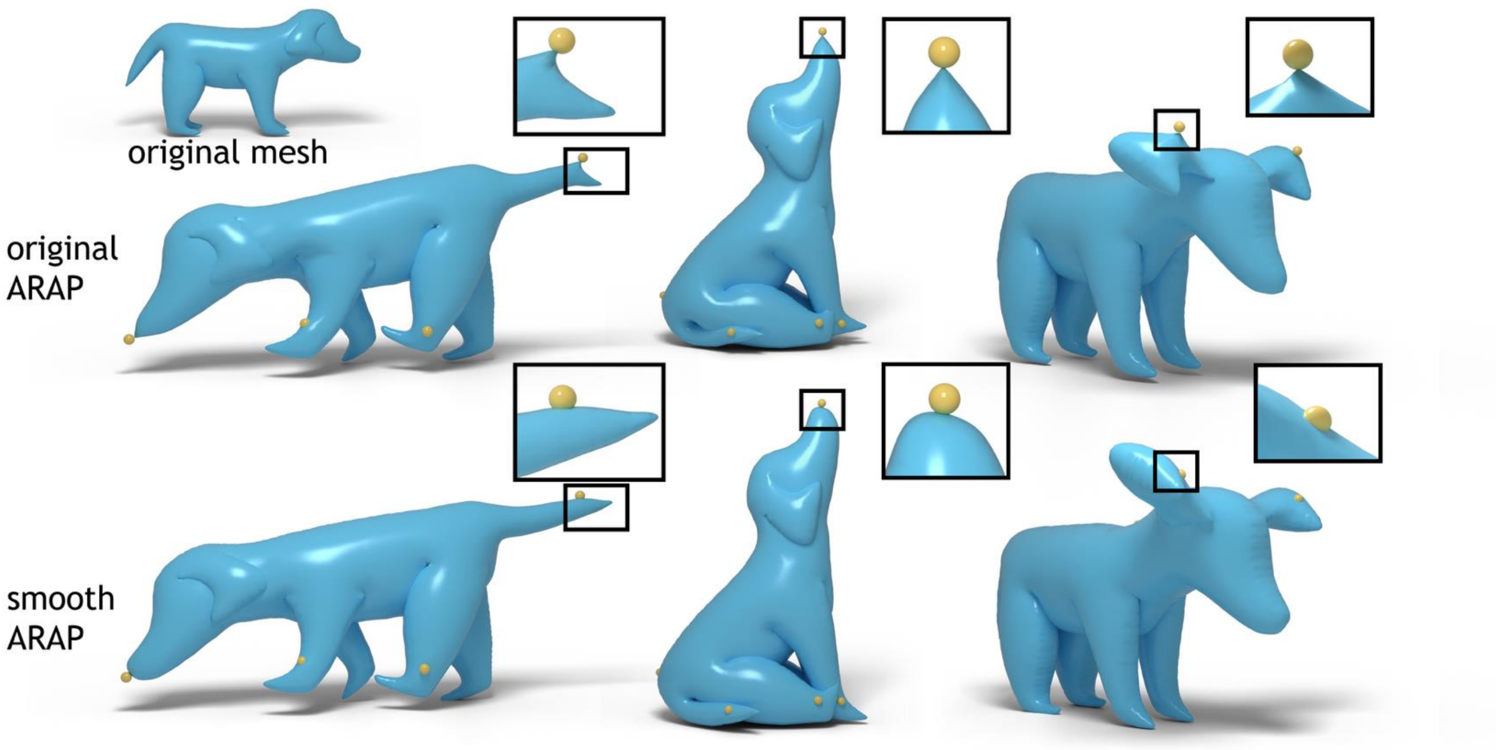
ARAP result



desired result

images from [Öhri et al. 2022]

Smooth ARAP



https://igl.ethz.ch/projects/smootharap/

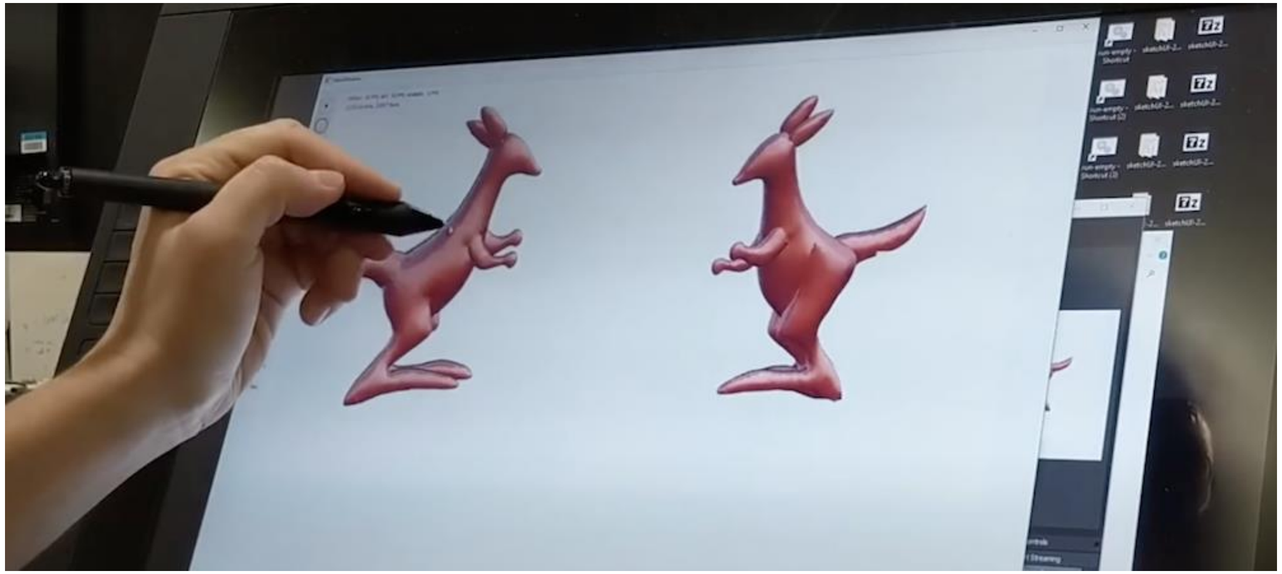
Acceleration and smoothness using subspace techniques

- Subspace created by **smooth** influence weight functions for each handle
- Drastically **reduces** the number of degrees of freedom in the optimization

Alec Jacobson, Ilya Baran, Ladislav Kavan, Jovan Popović, and Olga Sorkine. “Fast Automatic Skinning Transformations,” 2012.

Sketch based + ARAP

- **Monster Mash** https://monstermash.zone/



Thank you!

Thank You!

INTERACTIVE GEOMETRY LAB

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich