geometric optimization

CIX Summer School 2022 | Saarbrücken

Carnegie Mellon University

HCII

Prof. Alexandra Ion

interactive structures lab

hi, I am alex.

Assistant Professor
at CMU HCII

Carnegie Mellon University

HCII

open positions

Postdoc, 2020:
ETH Zurich
Olga Sorkine-Hornung, Graphics

ETH zürich

PhD, 2019:
Hasso Plattner Institute
Patrick Baudisch, HCI

HPI Hasso Plattner Institut

interactive structures lab

digital fabrication

+

interactive structures

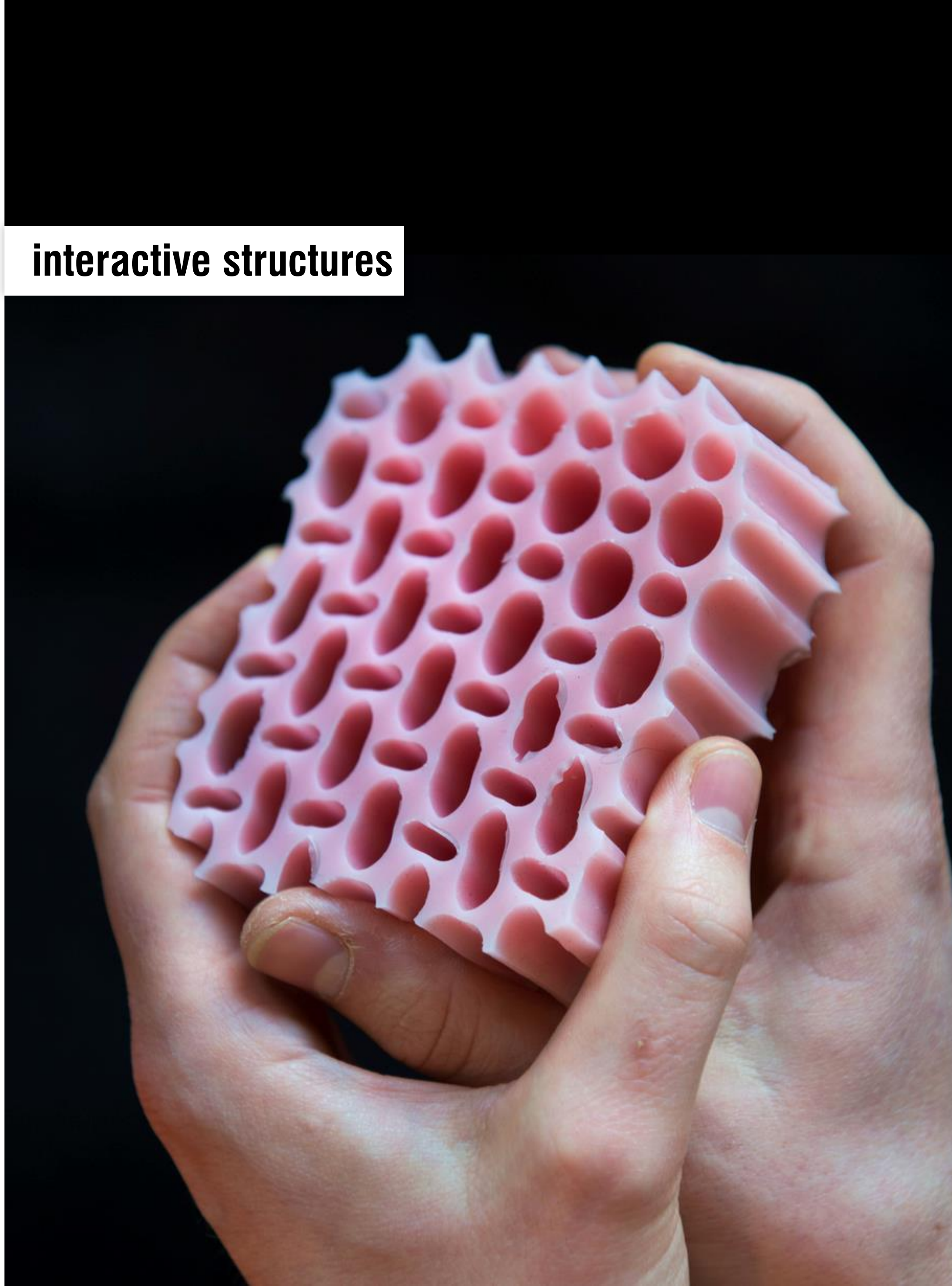
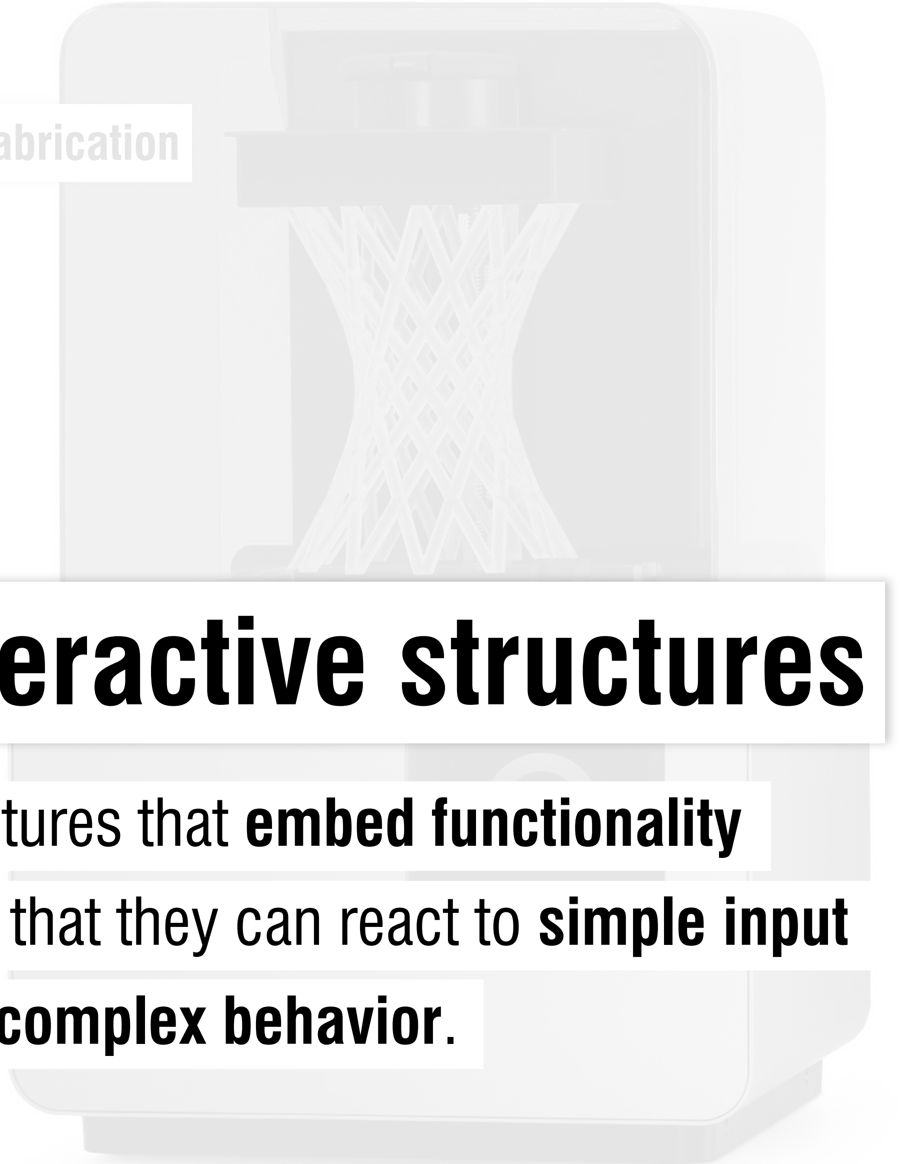


image: formlabs.com

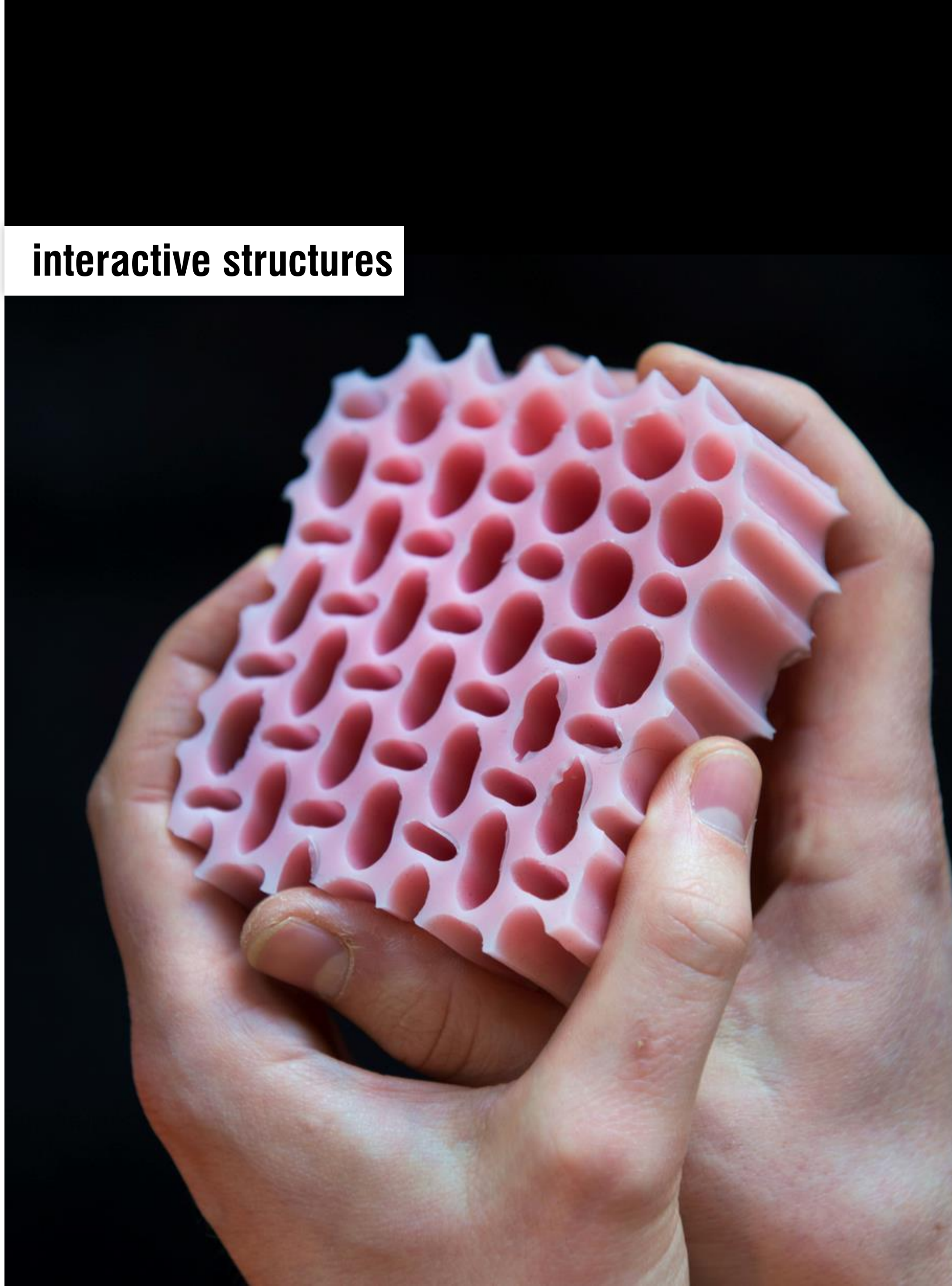


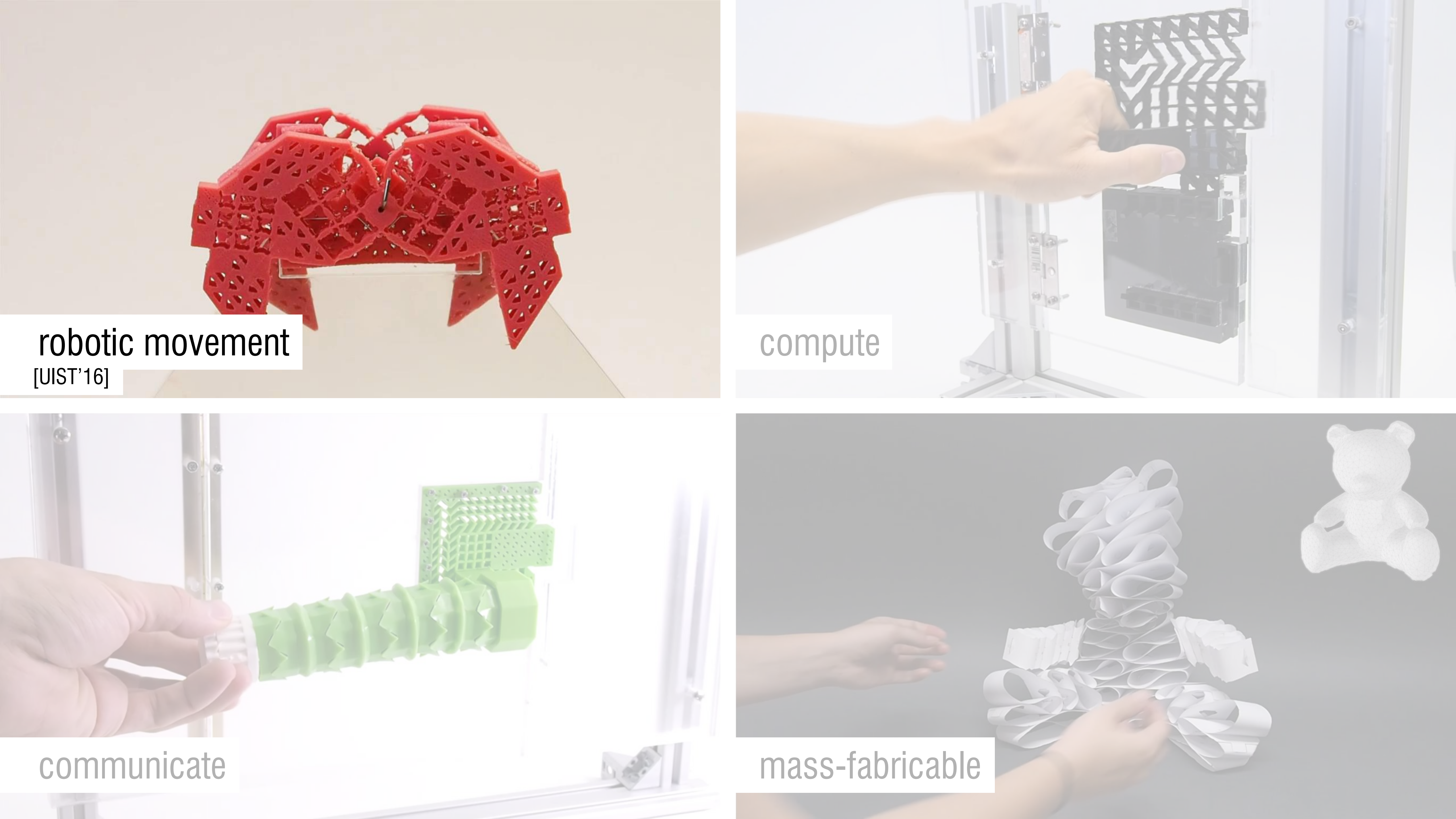
interactive structures

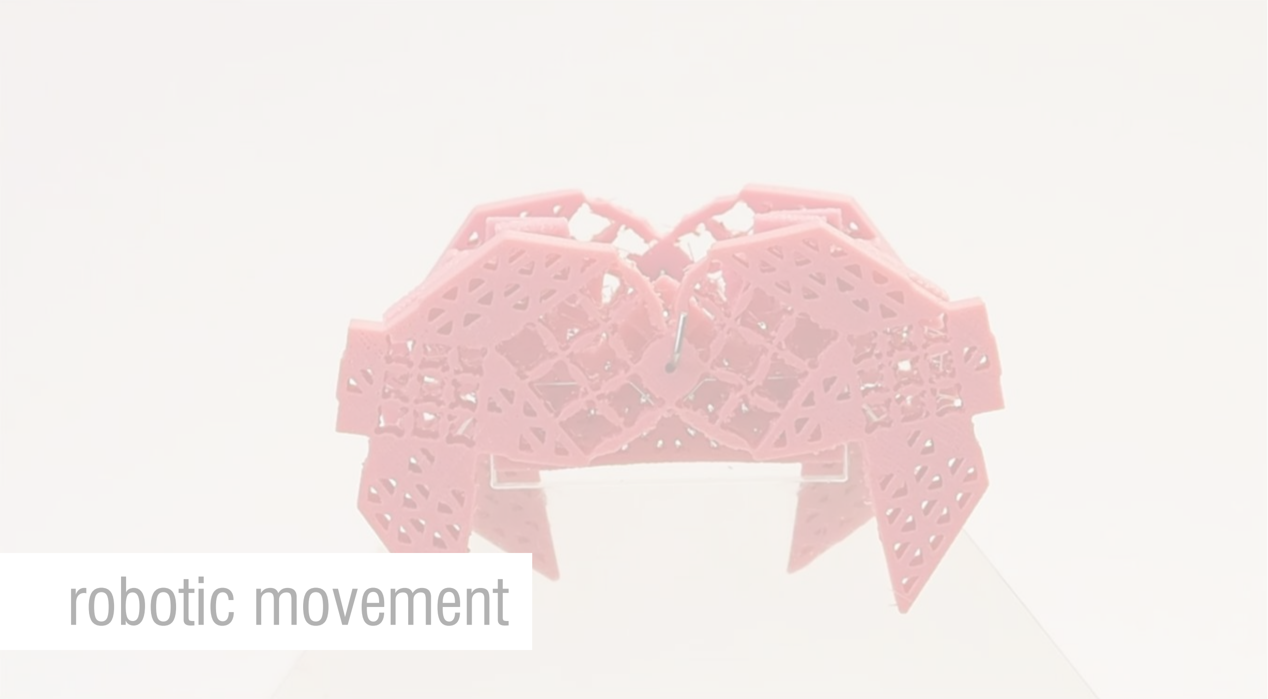
structures that **embed functionality**
such that they can react to **simple input**
with **complex behavior**.

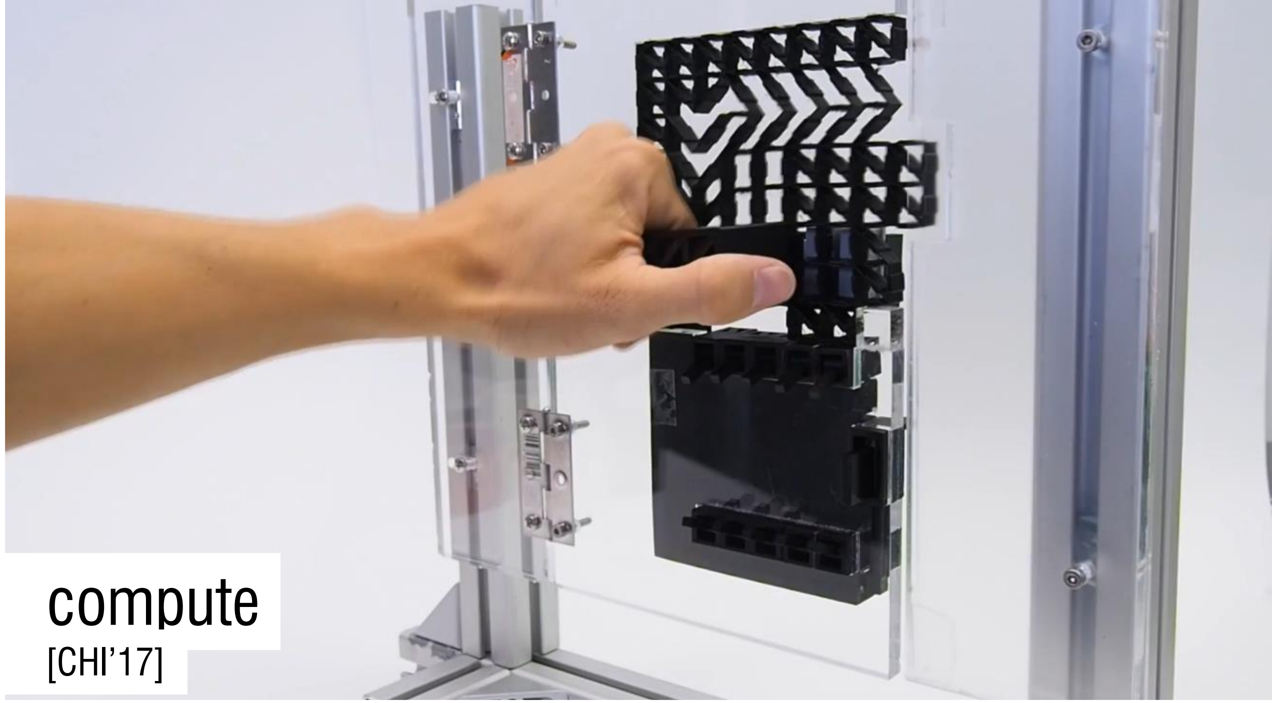
image: formlabs.com

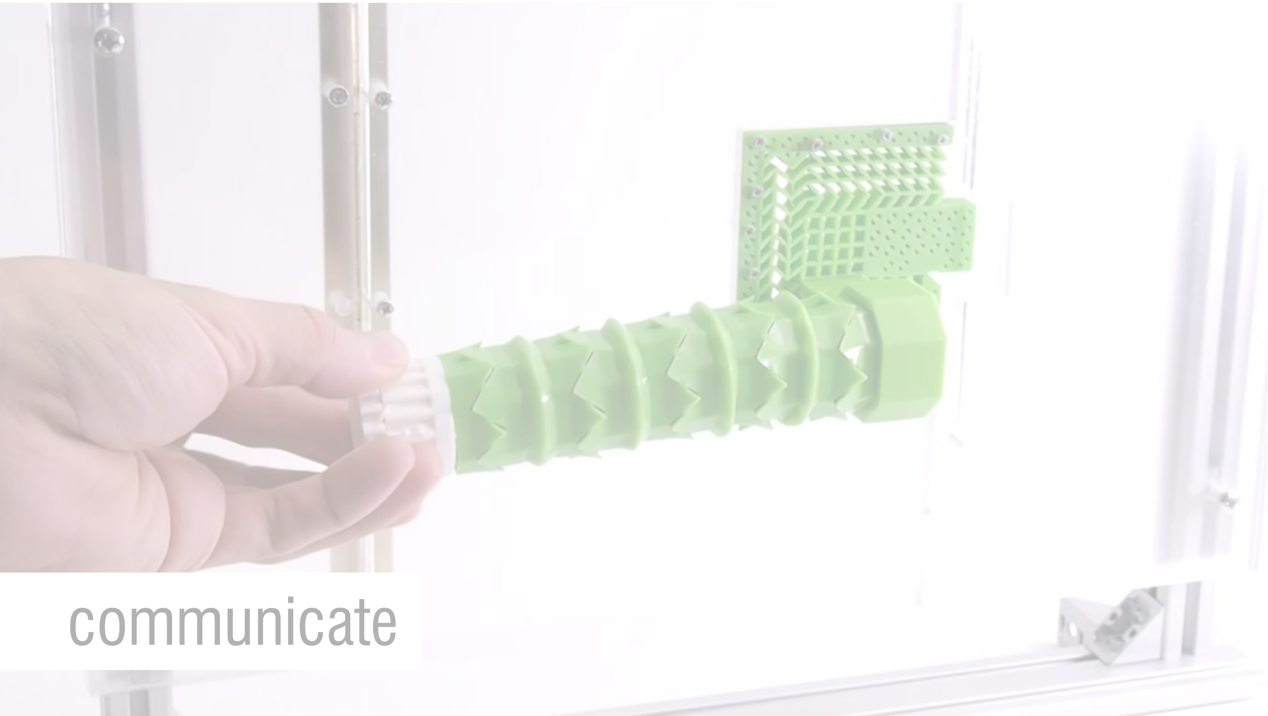
+



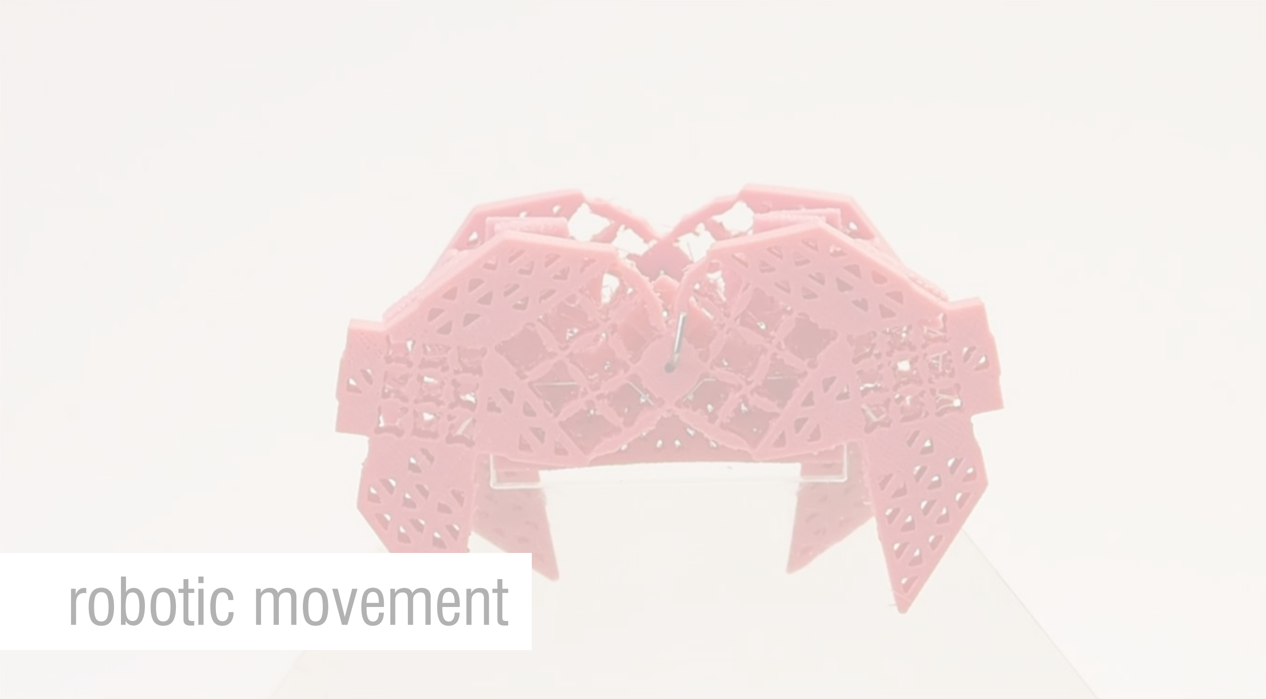

robotic movement
[UIST'16]
compute
communicate
mass-fabricable

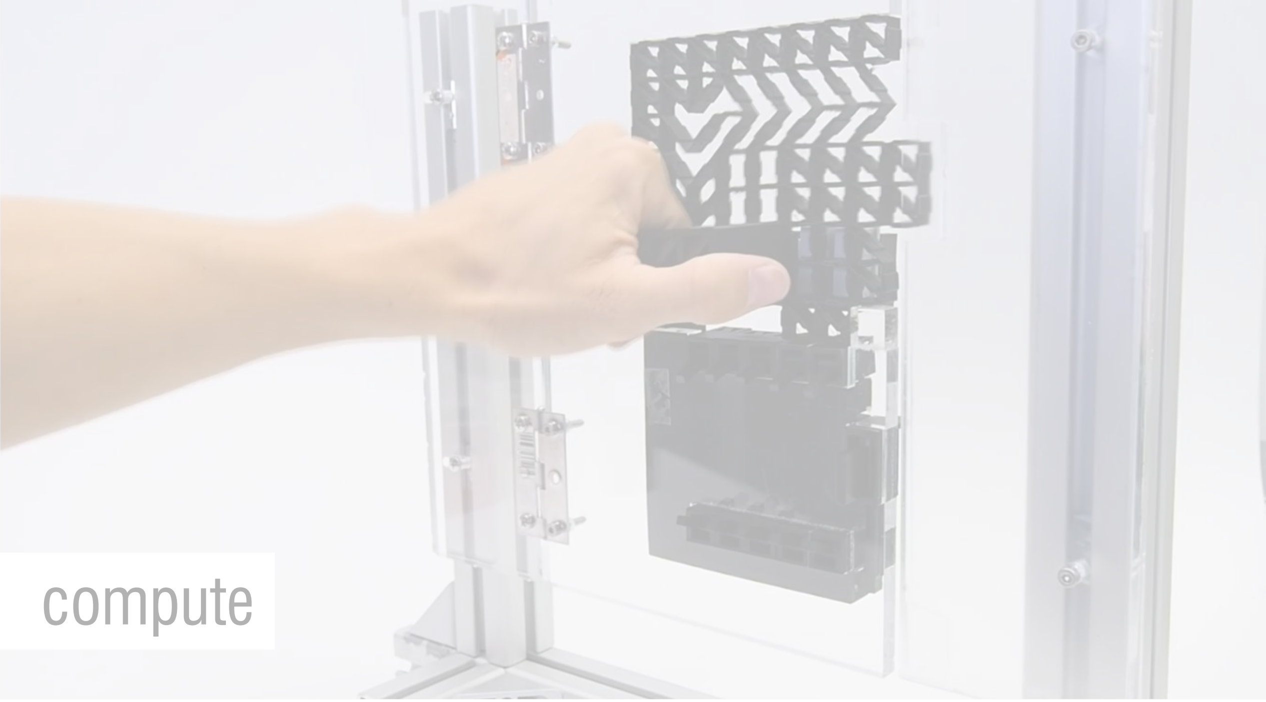

robotic movement

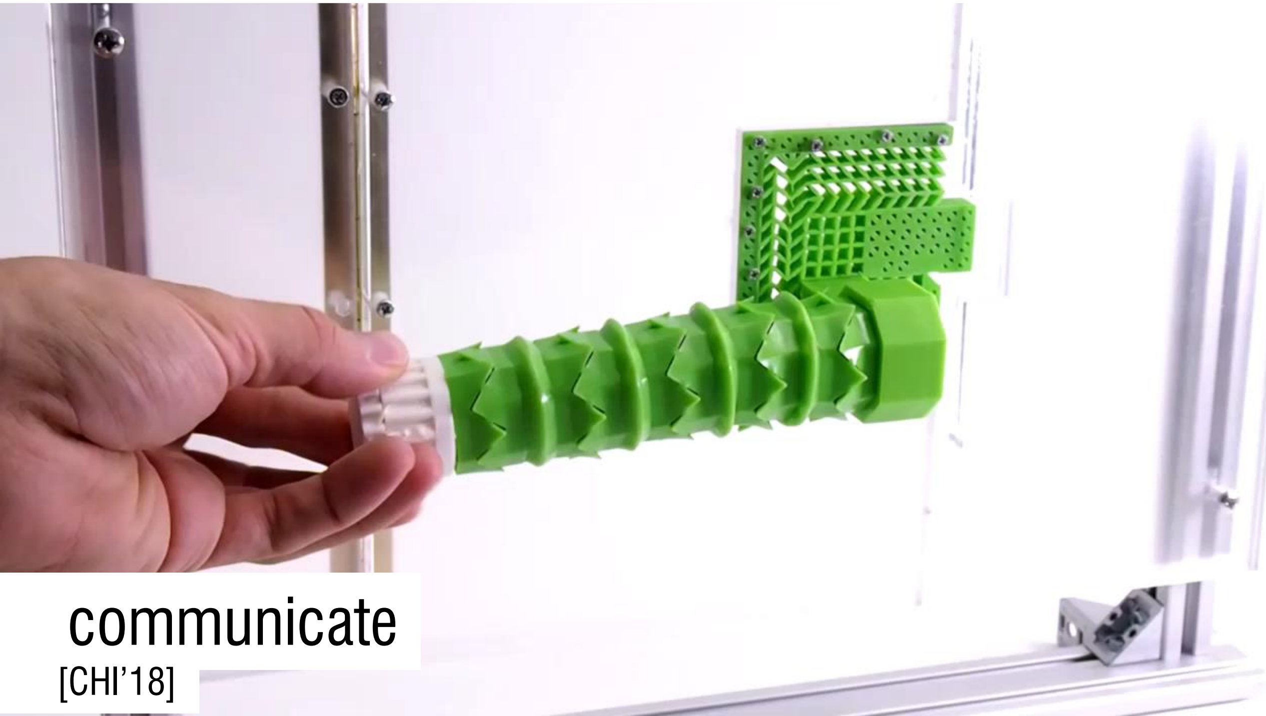

compute
[CHI’17]


communicate

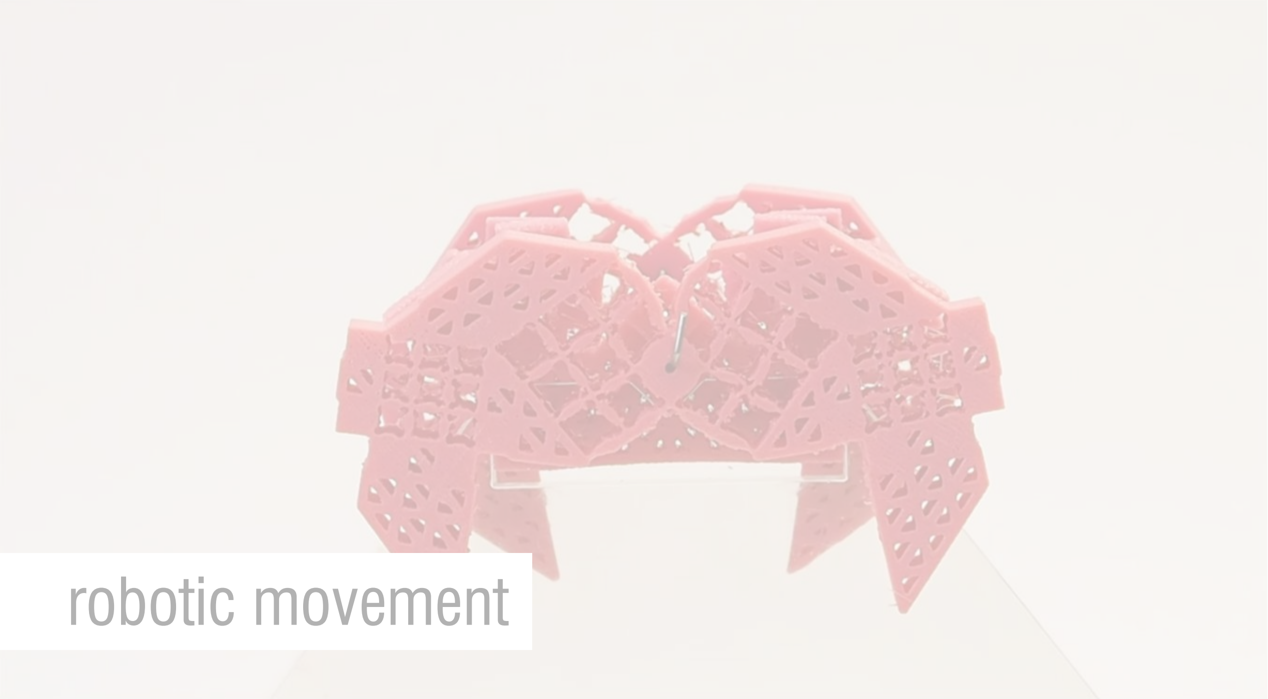

mass-fabricable

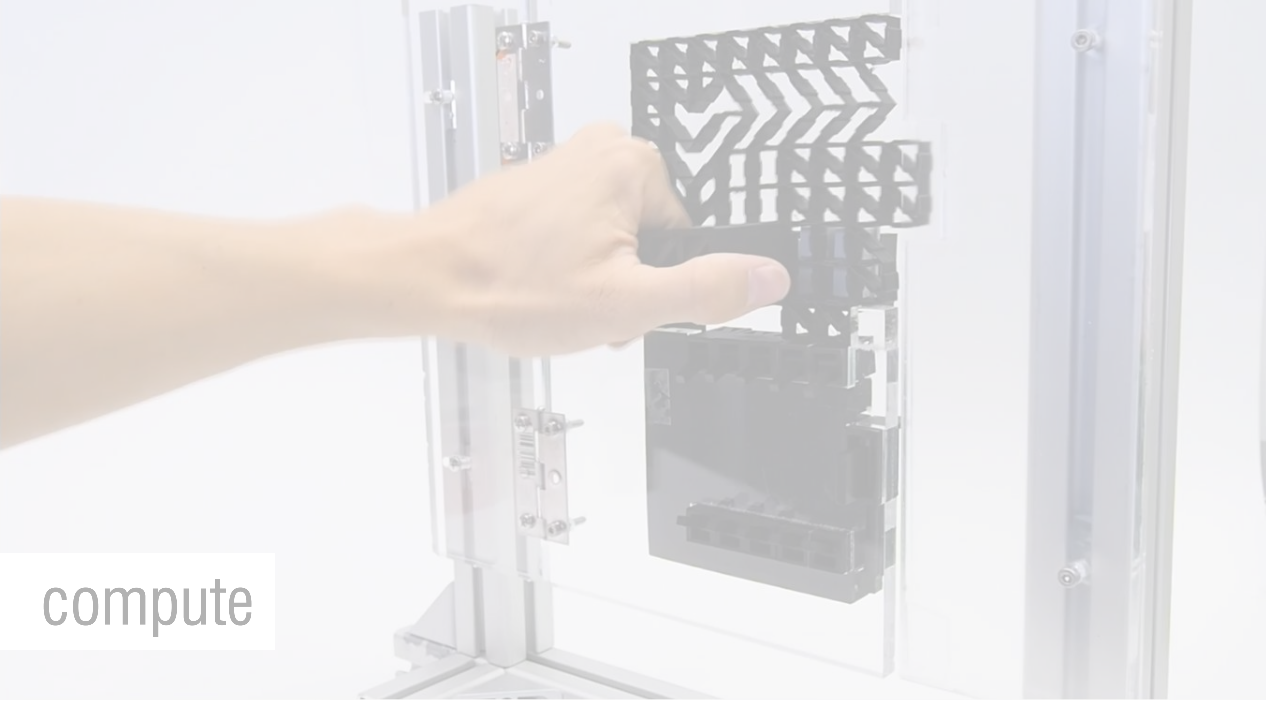

robotic movement

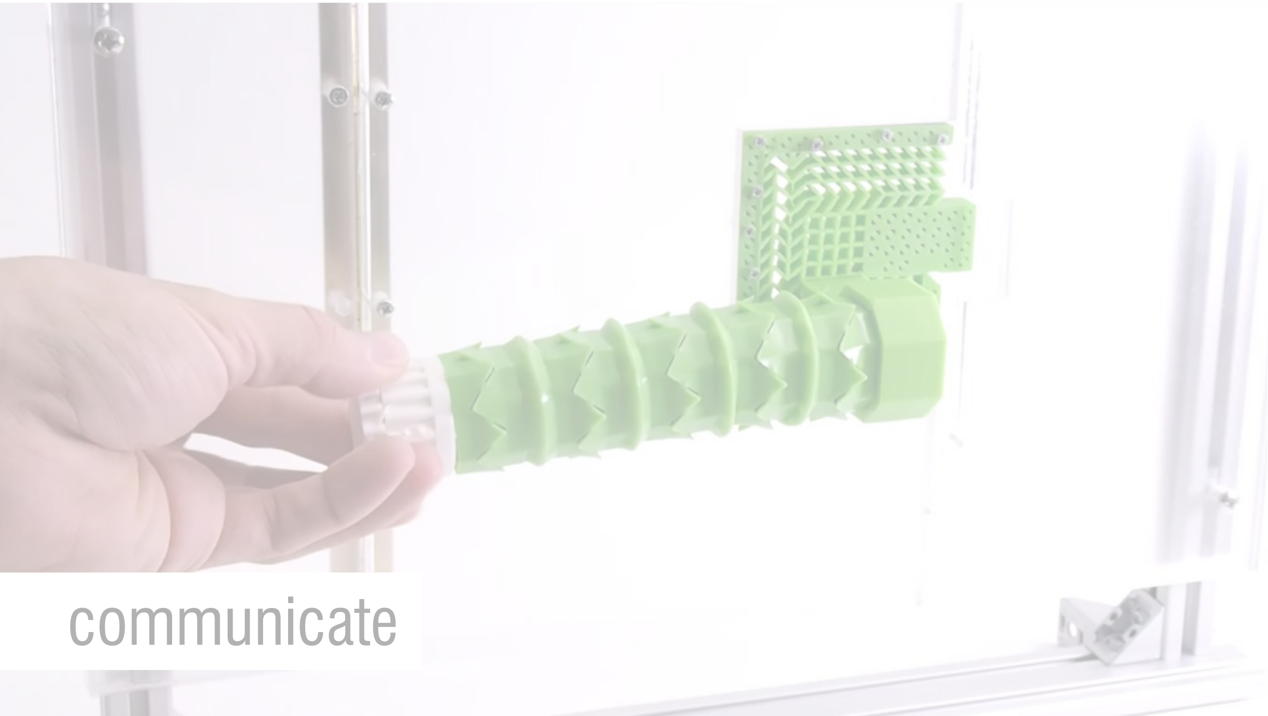

compute

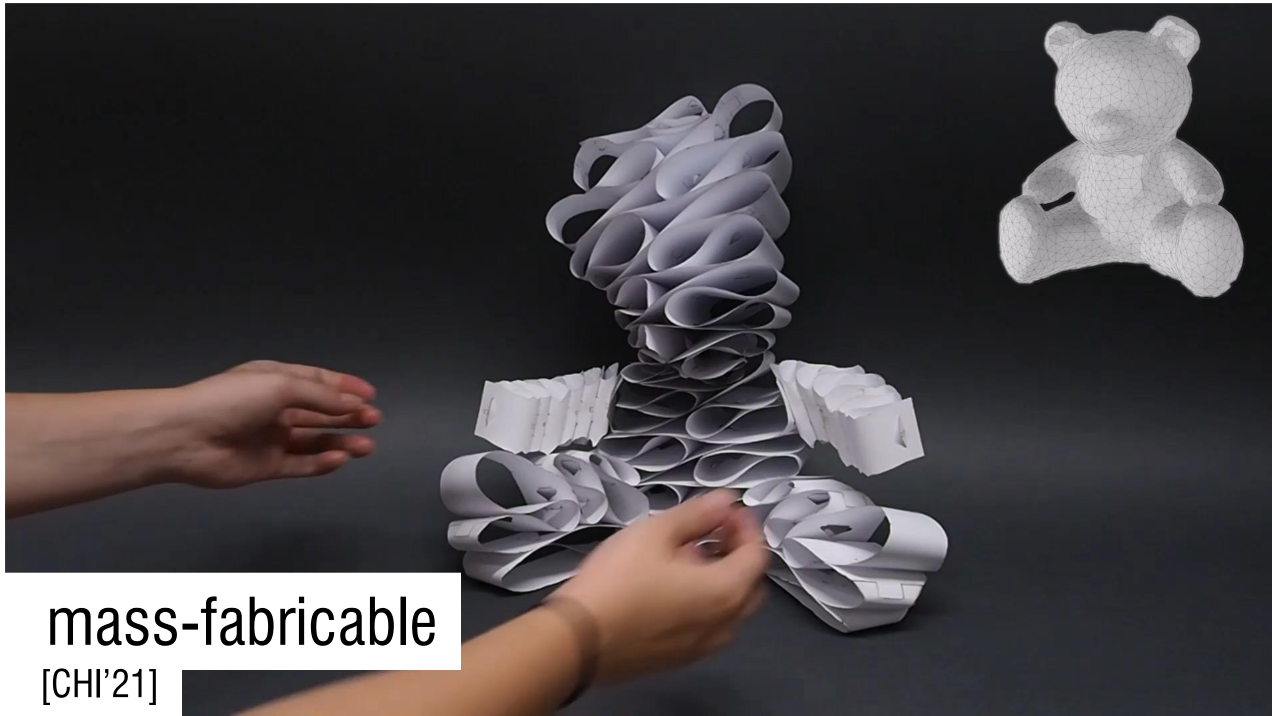

communicate
[CHI'18]


mass-fabricable


robotic movement


compute

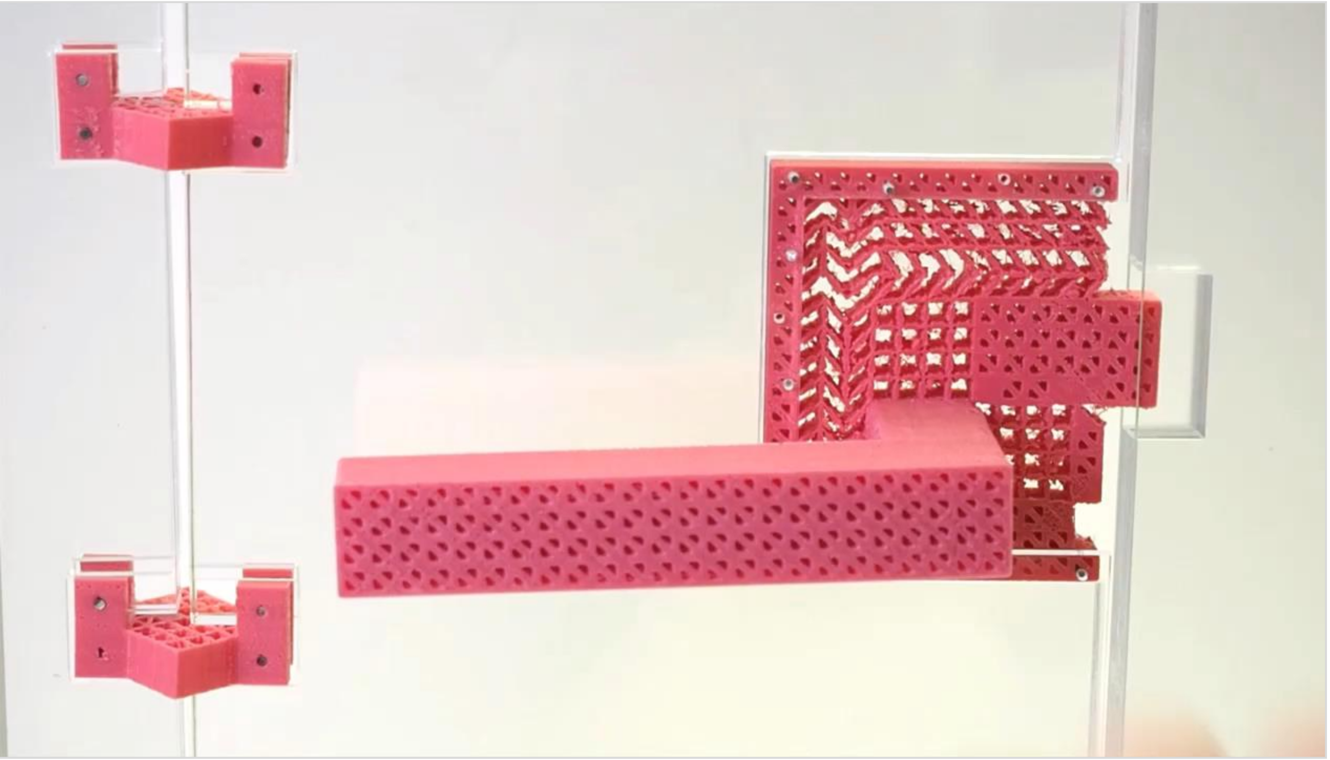

communicate


mass-fabricable
[CHI'21]

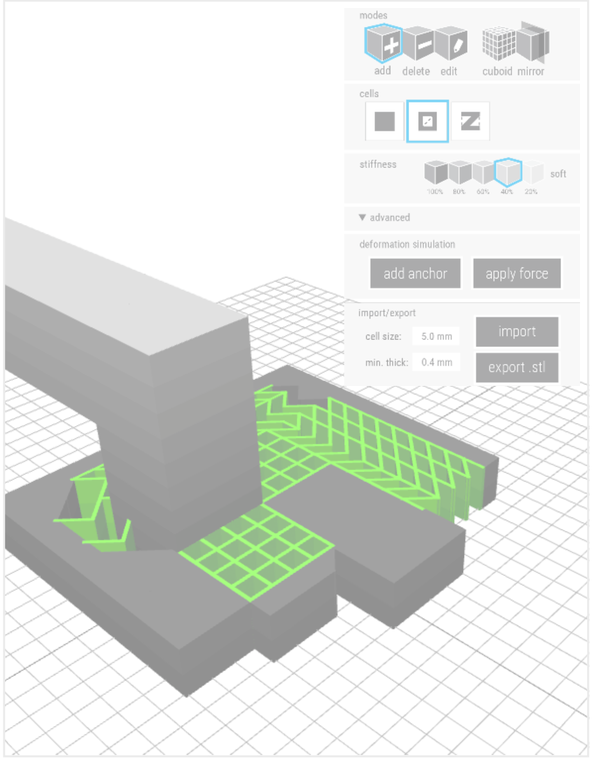
HCI + geometry + engineering + design

the work we do in the **lab**

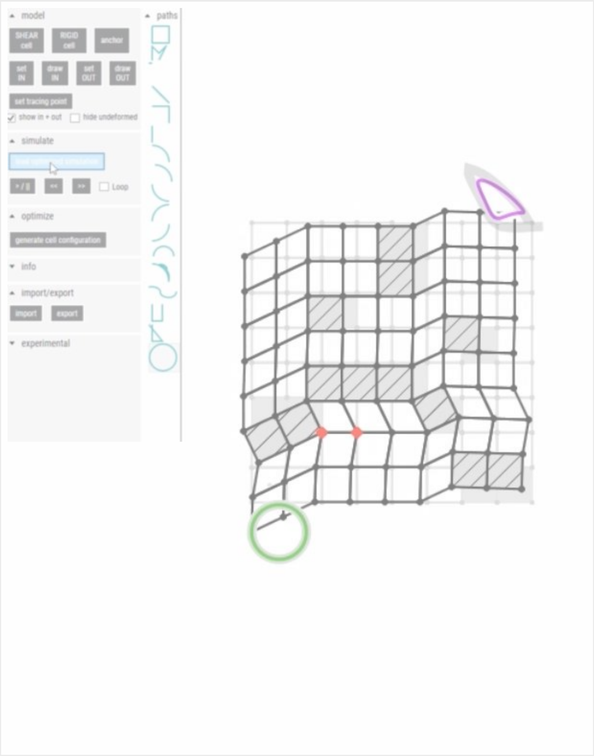
example: Metamaterial Mechanisms
[UIST’16, CHI’19]



structures



editor

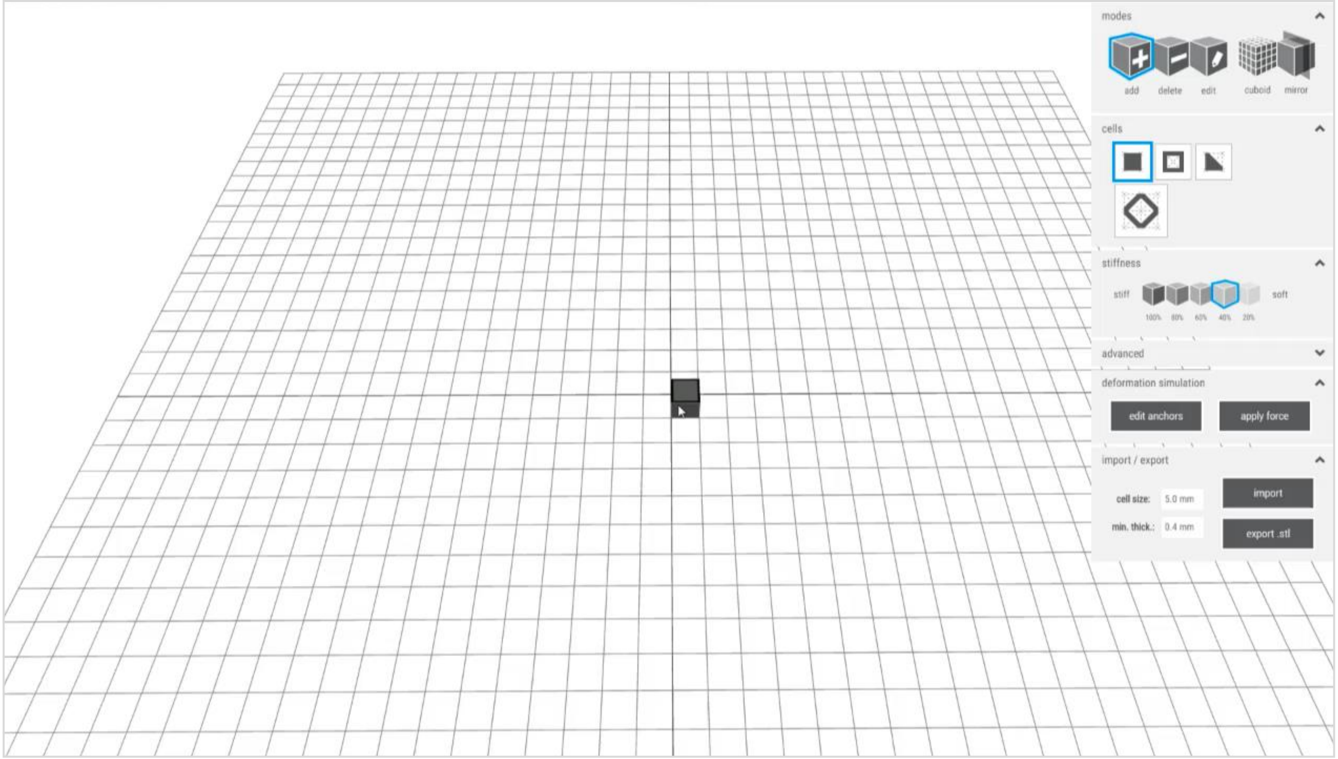


inverse design

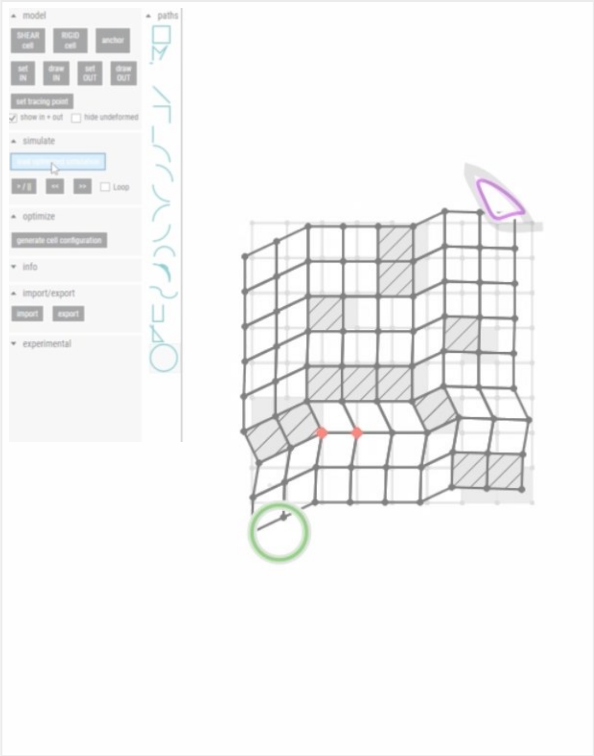
the work we do in the **lab**

example: Metamaterial Mechanisms
[UIST'16, CHI'19]

structures



editor



inverse design

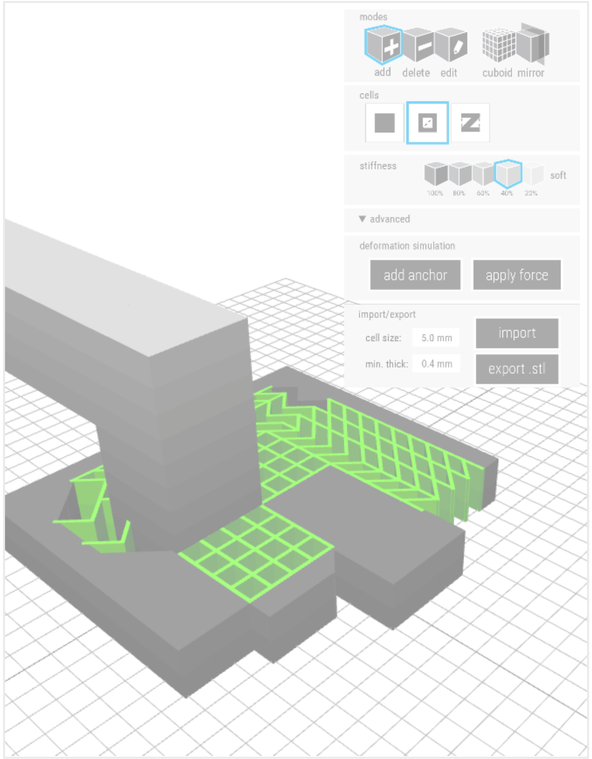
Carnegie Mellon University

the work we do in the **lab**

example: Metamaterial Mechanisms

[UIST’16, CHI’19]

structures



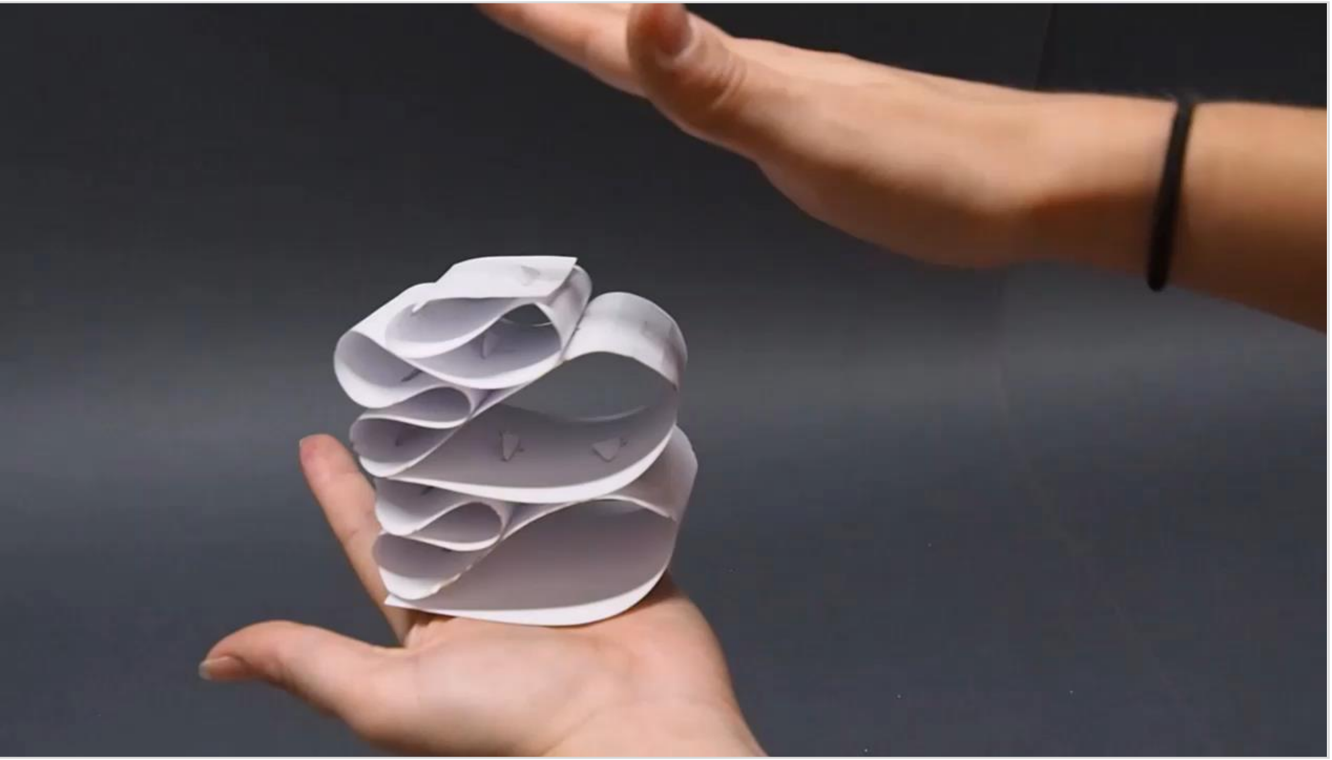
editor

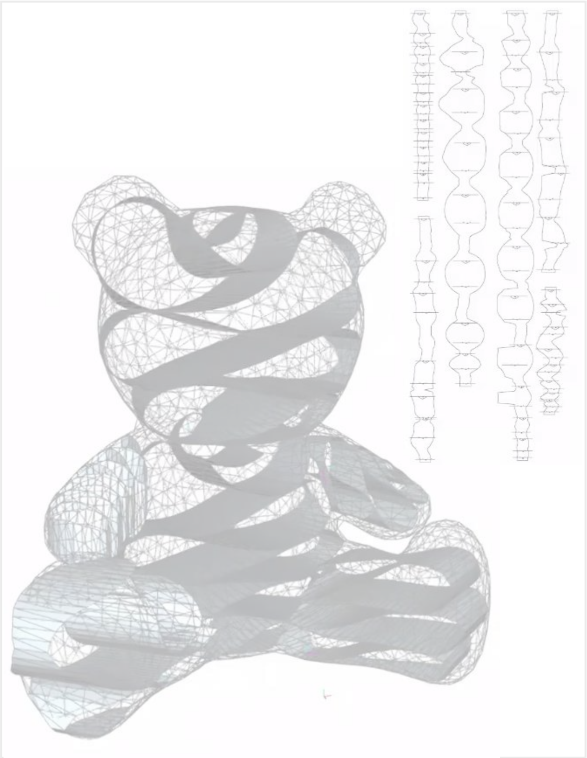
inverse design

combinatorial optimization, meta-heuristics

the work we do in the **lab**

example: Developable Metamaterials [CHI'21]





structures
developable, mass-manufacturable, easy to assemble

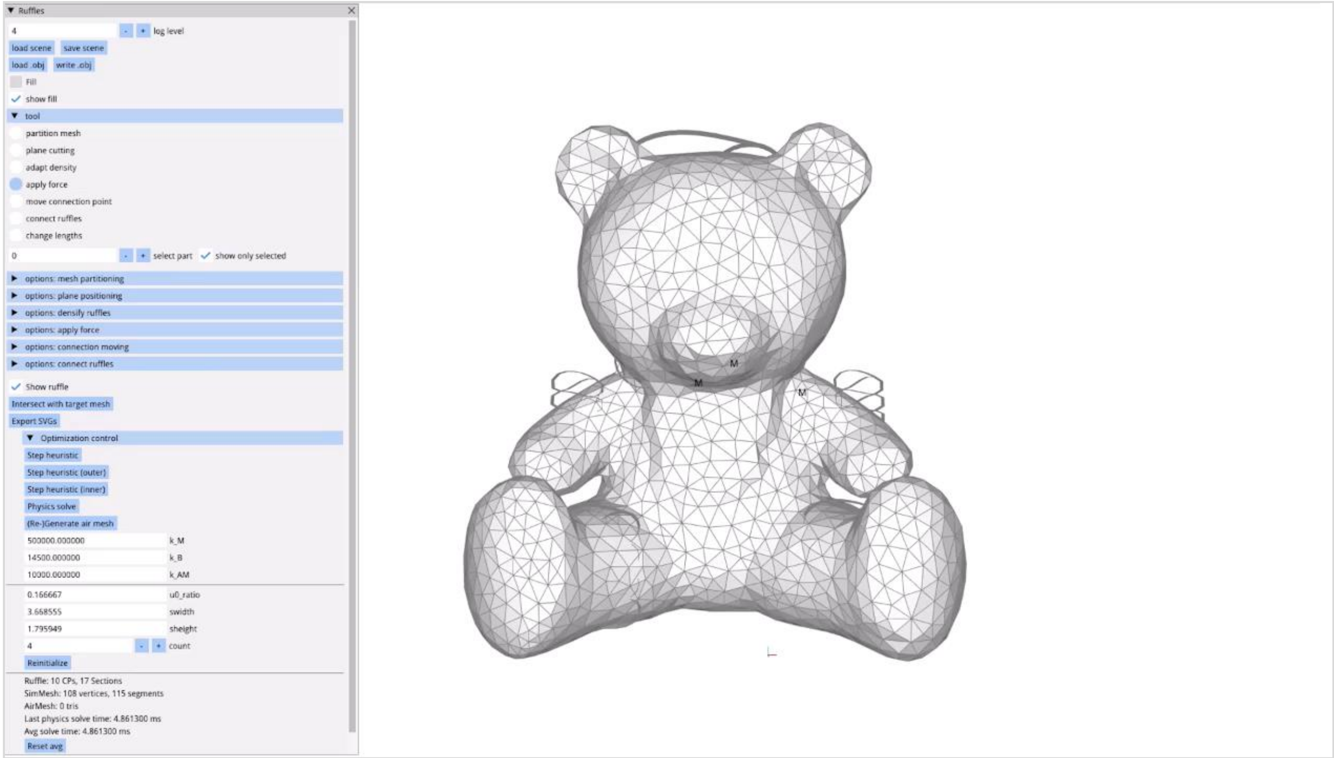
inverse design

impactful applications

the work we do in the **lab**

example: Developable Metamaterials [CHI'21]

structures



inverse design

thin shells, physics simulation, shape optimization

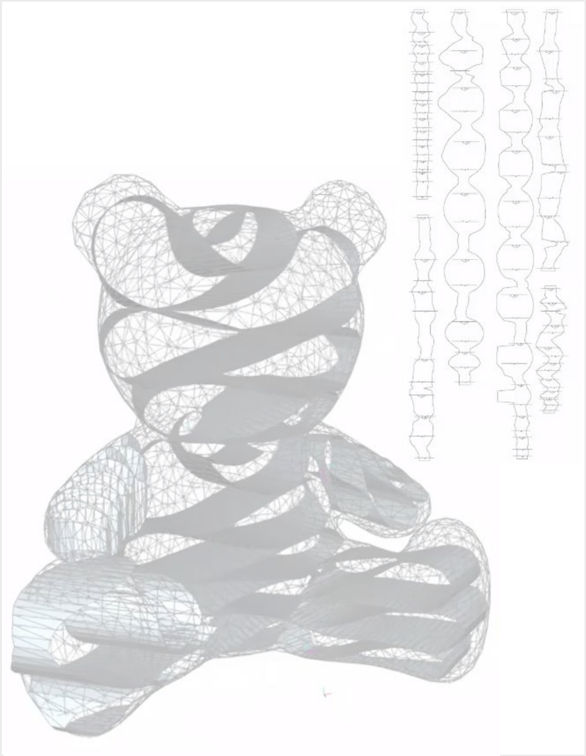
impactful applications

Carnegie Mellon University

the work we do in the **lab**

example: Developable Metamaterials [CHI'21]

structures

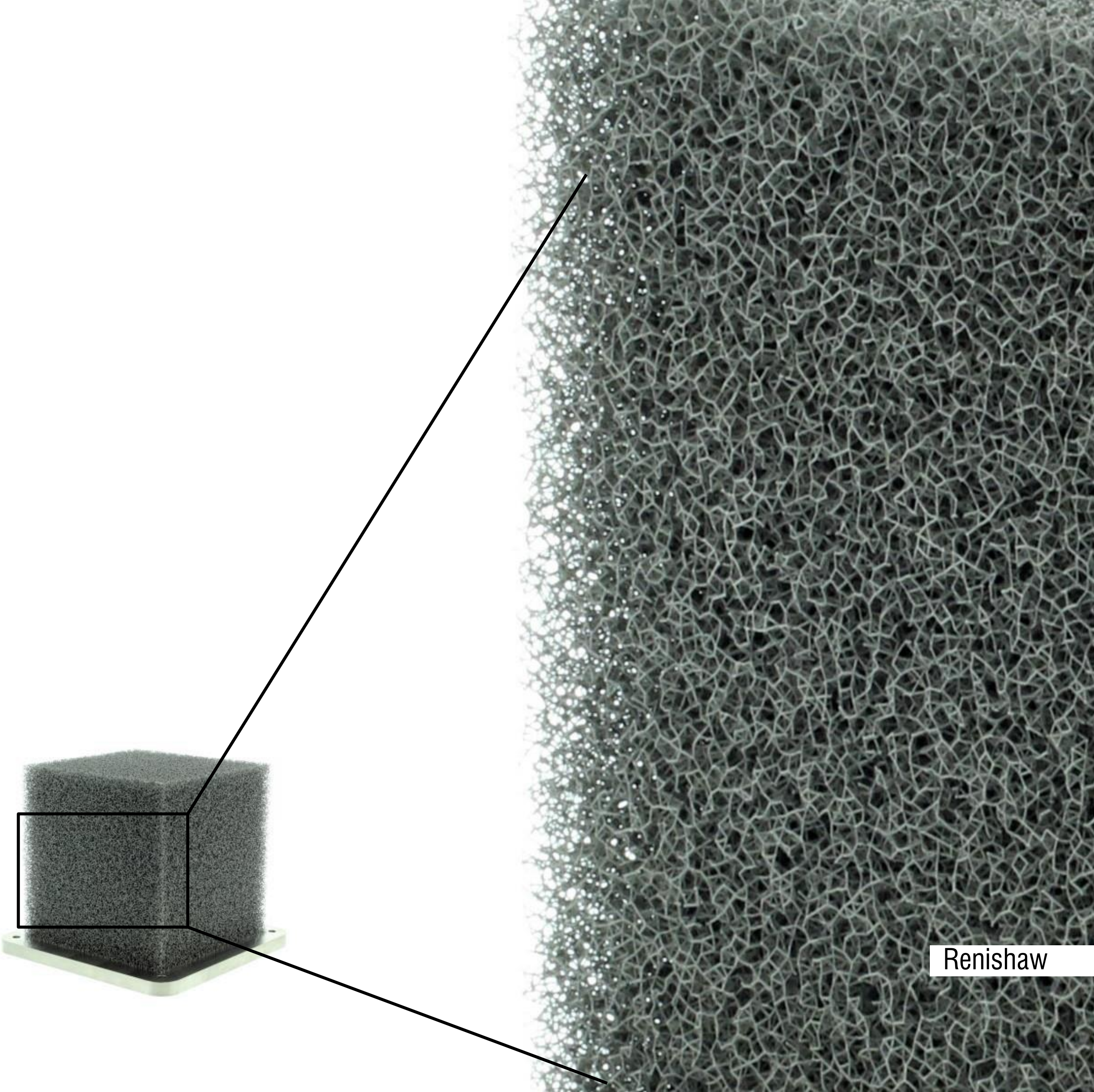


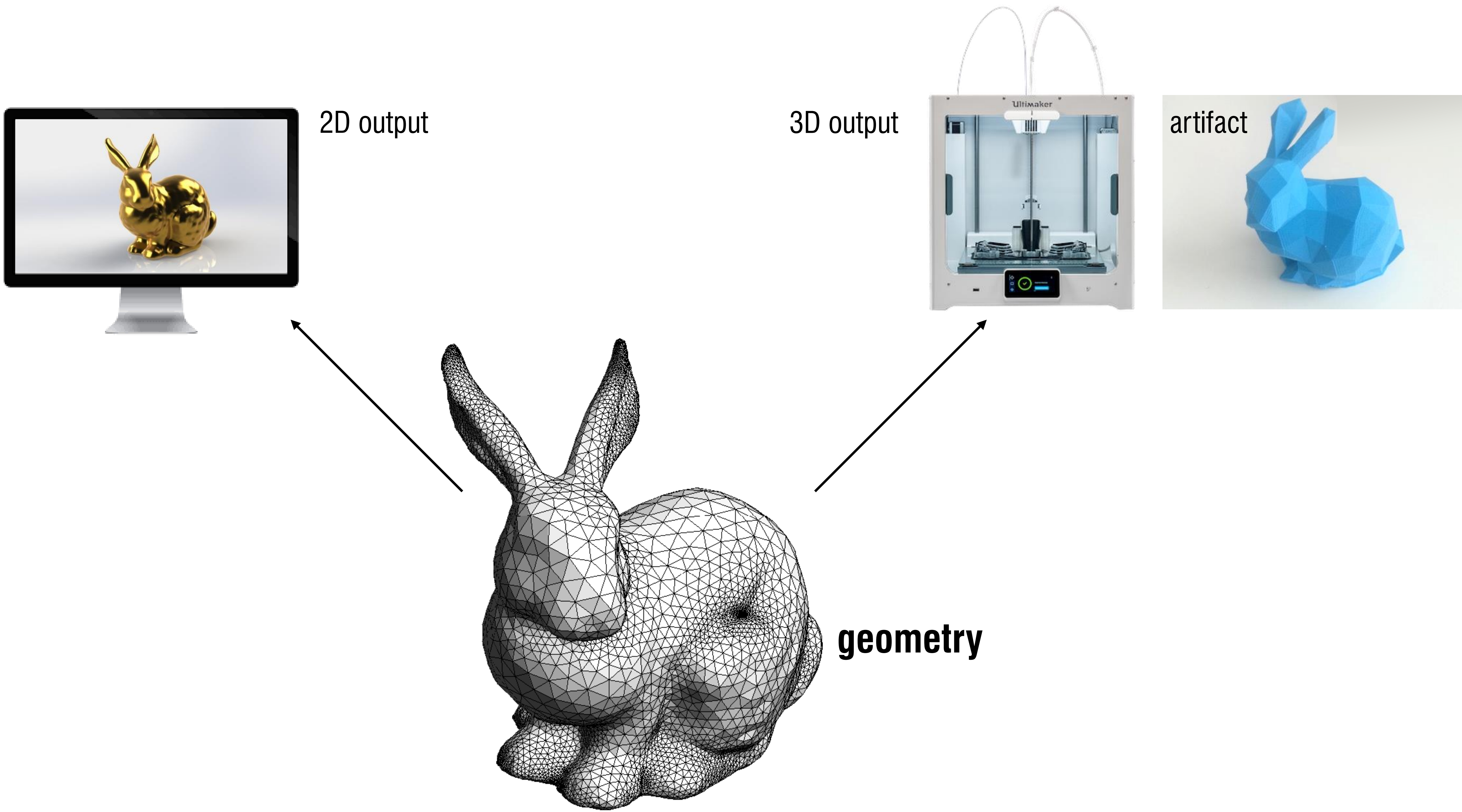
inverse design

impactful applications

customizable, affordable, sensorized and actuated prostheses

Geometric complexity




2D output
3D output
Ultimaker
artifact
geometry

geometry

interactive structures lab

basics of geometry processing

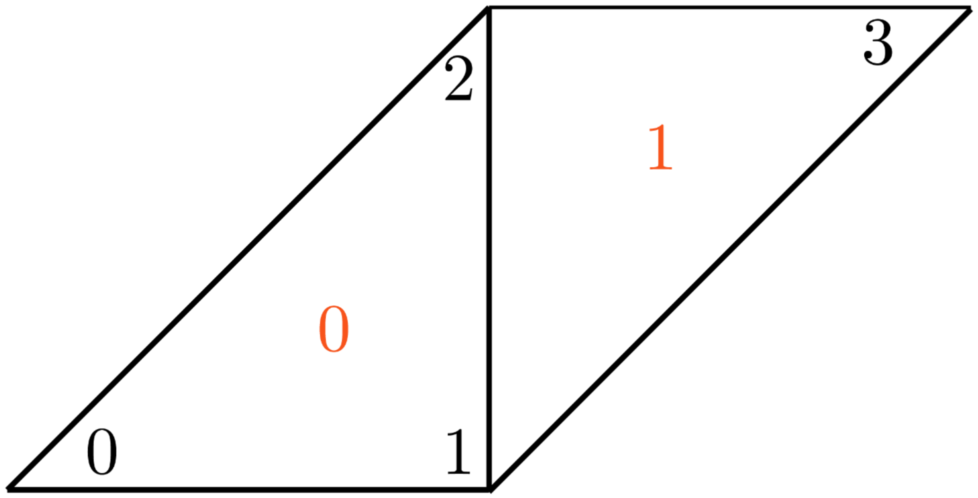
libigl

Mesh representation

Triangles with vertices V and faces F

```
Eigen::MatrixXd V;
Eigen::MatrixXi F;
```

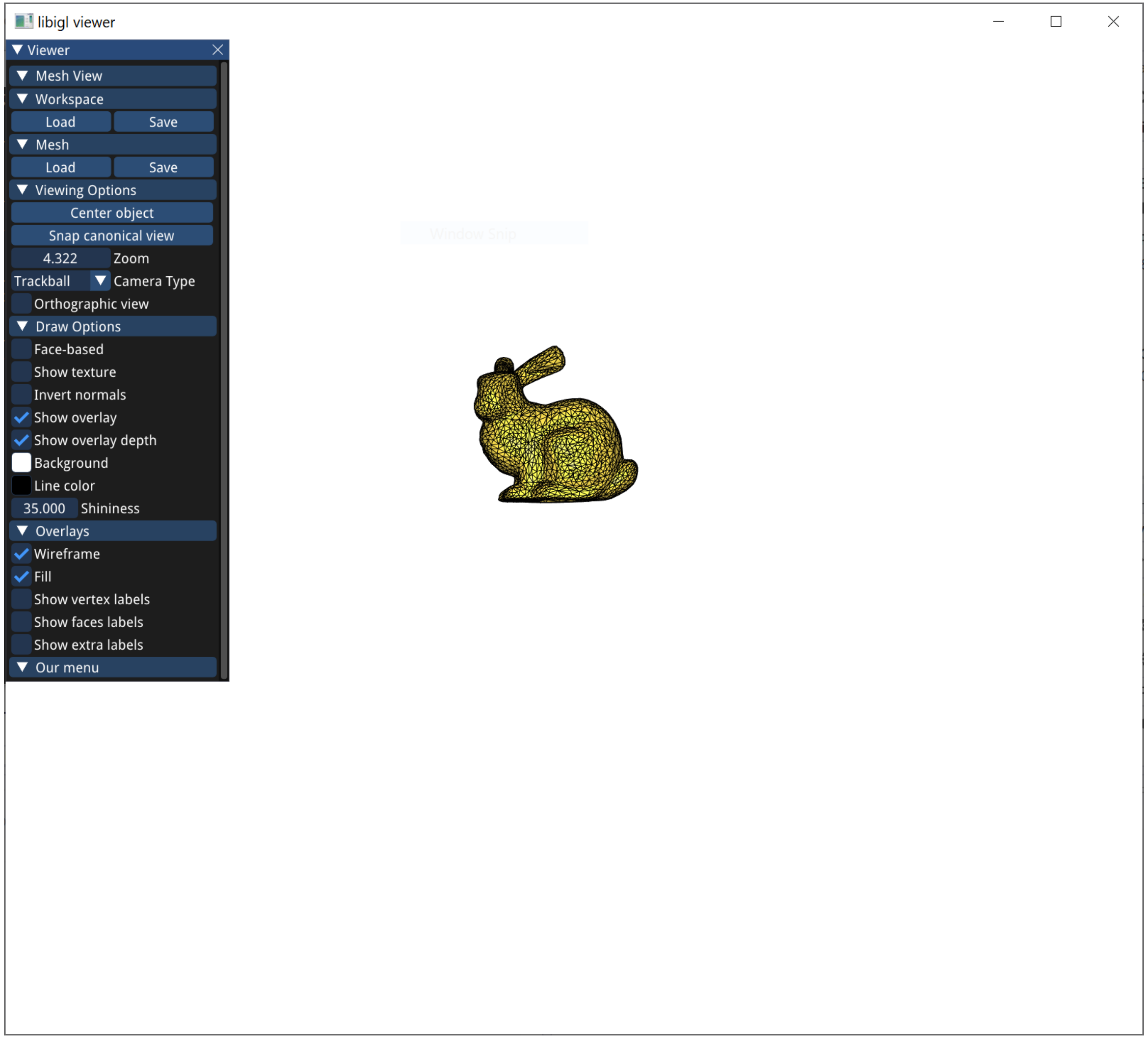
$$V = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 1 & 0 \end{pmatrix} \qquad F = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 3 & 2 \end{pmatrix}$$



https://libigl.github.io/tutorial/

Let's go!

Run `0_compilation_test`



Quick refs

libigl: great tutorials, courses, etc

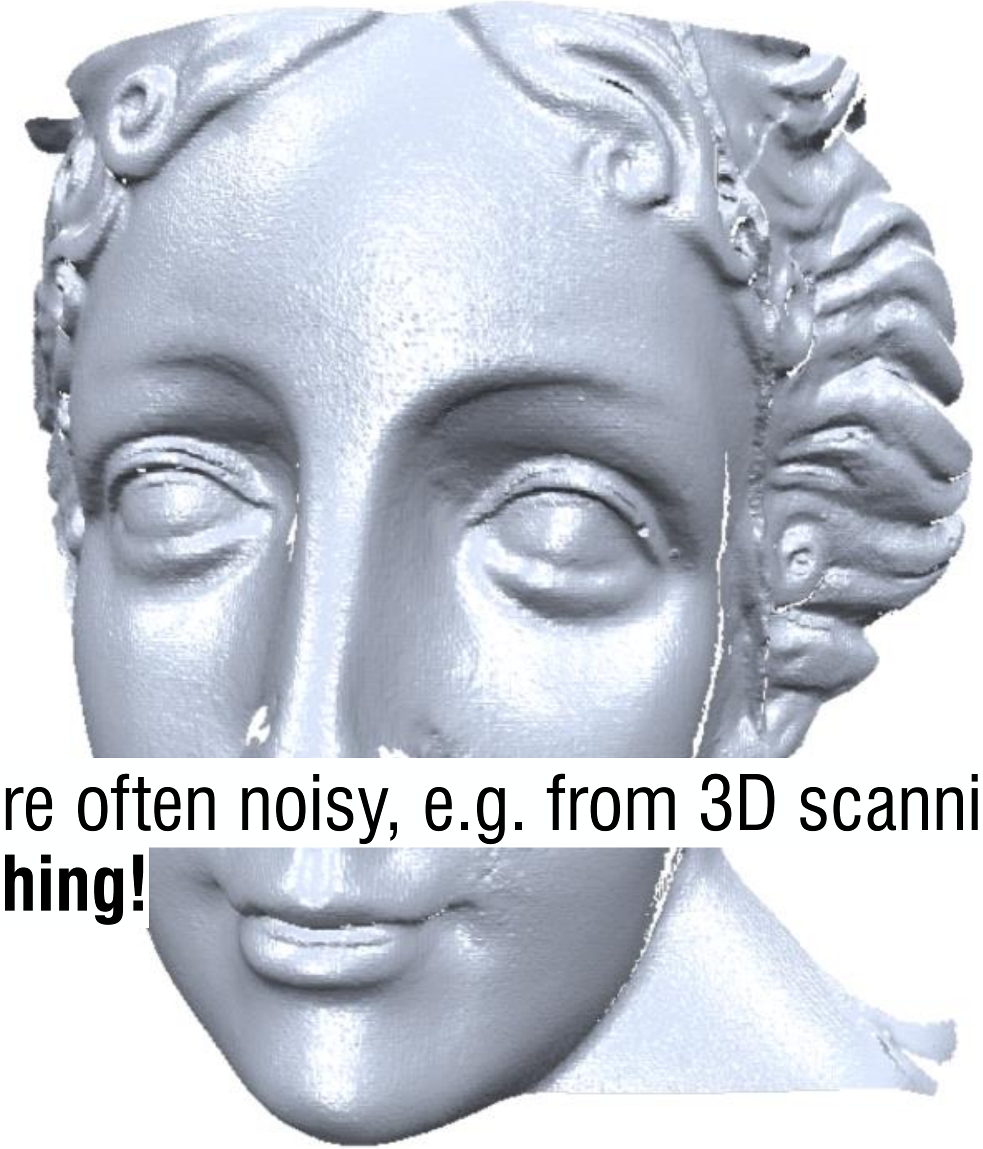
https://libigl.github.io/tutorial

https://libigl.github.io/tutorial/#other-matlab-style-functions

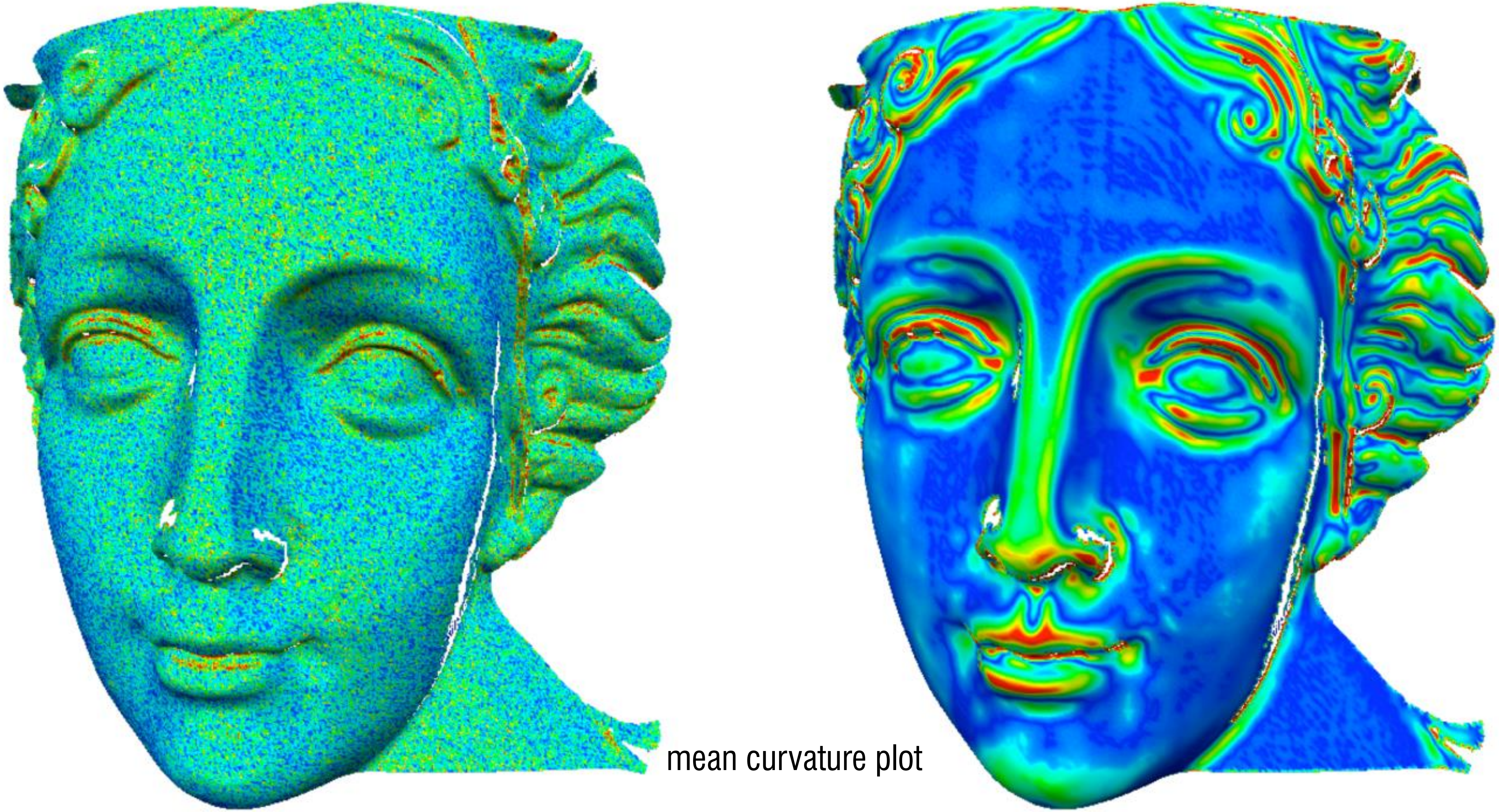
API: none really, but search in github (one-function one-file principle)

→ IGL, main contributors Alec Jacobson & Danielle Panozzo 👏

Eigen: https://eigen.tuxfamily.org/dox/group__QuickRefPage.html

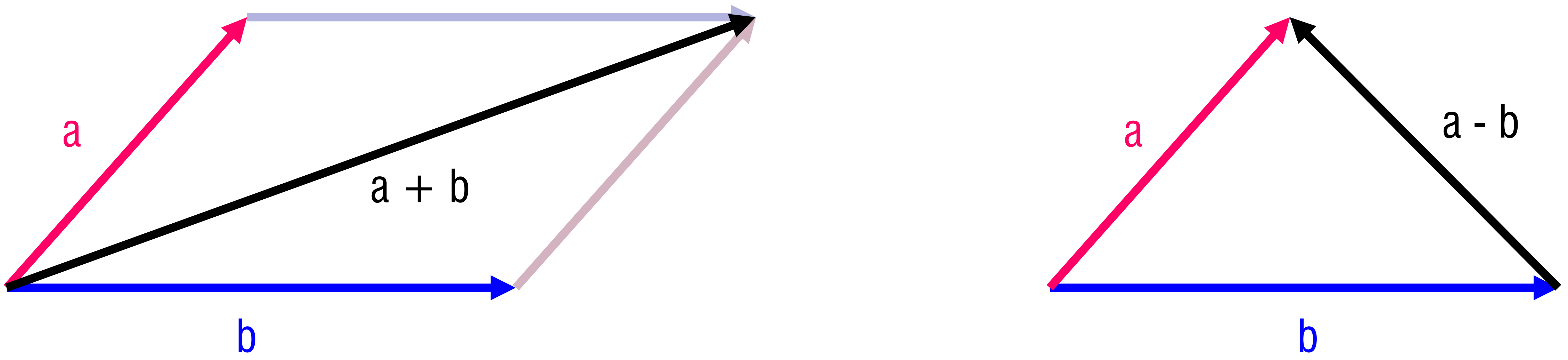


meshes are often noisy, e.g. from 3D scanning
→ **smoothing!**



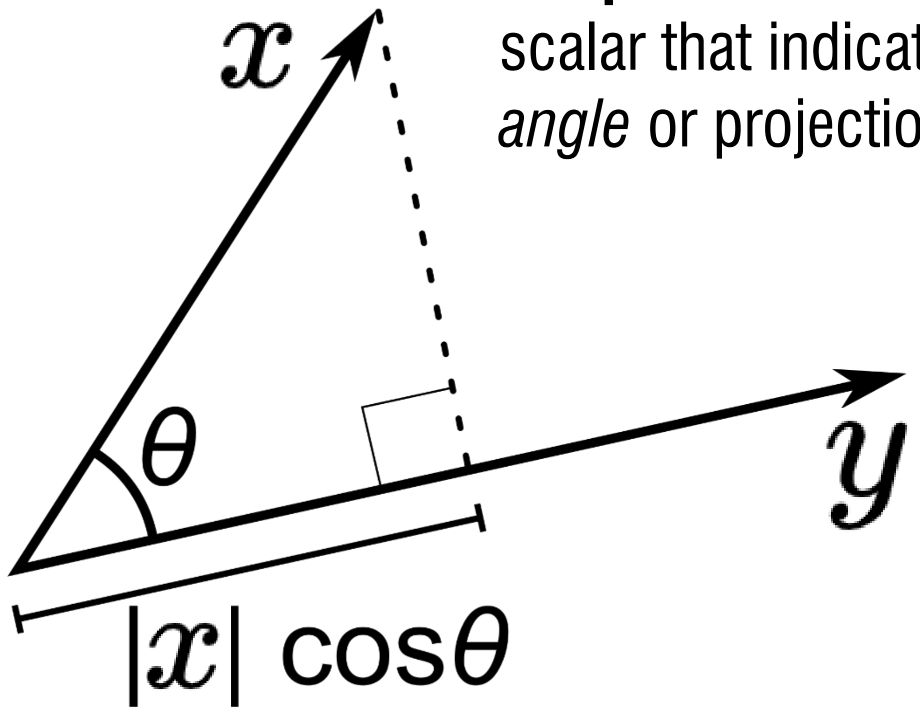
mean curvature plot

Vector cheat sheet



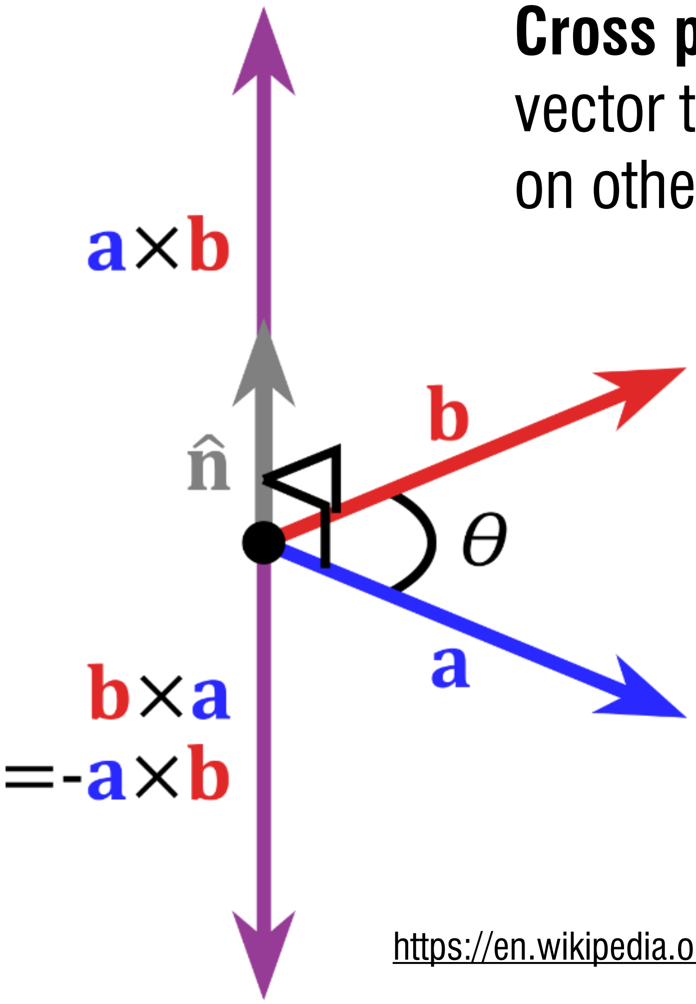
https://en.wikipedia.org/wiki/Euclidean_vector

Vector cheat sheet



Dot product:
scalar that indicates *angle* or projection

$$\angle(x, y) = \arccos \frac{\langle x, y \rangle}{\|x\|\,\|y\|}$$



Cross product:
vector that is normal on other two vectors

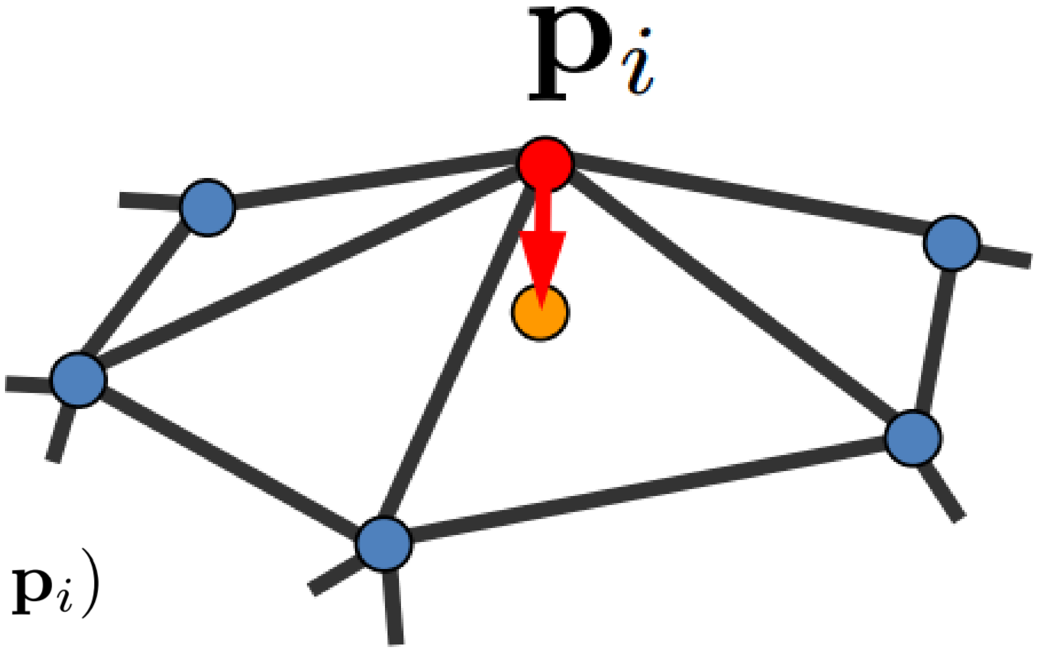
https://en.wikipedia.org/wiki/Cross_product

Smoothing using Laplace-Beltrami operator

High-pass filter: extracts local surface detail

Detail = ***smooth***(surface) – surface

Assumption: smoothing = averaging



$$\delta_i = \frac{1}{W_i} \sum_{j \in \mathcal{N}(i)} w_{ij} (\mathbf{p}_j - \mathbf{p}_i)$$

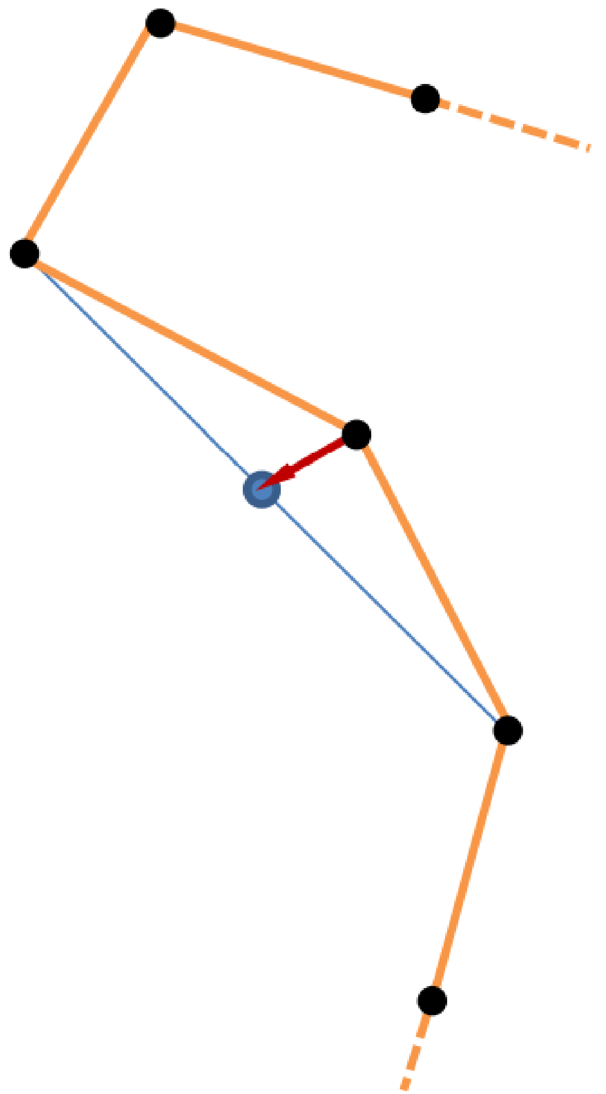
Olga Sorkine-Hornung, ETH, Shape Modeling course

Smoothing curves

Let's look at a simple discrete curve first (DDG!)

Laplace in 1D = second derivative:

$$L(\mathbf{p}_i) = \frac{1}{2}(\mathbf{p}_{i-1} - \mathbf{p}_i) + \frac{1}{2}(\mathbf{p}_{i+1} - \mathbf{p}_i)$$



Olga Sorkine-Hornung, ETH, Shape Modeling course

Any drawbacks?

<30sec brainstorming>

Let's see...

compile **`1_curve_smoothing`**

Laplace-Beltrami weighing schemes

$$\delta_i = \frac{1}{W_i} \sum_{j \in \mathcal{N}(i)} w_{ij}(\mathbf{p}_j - \mathbf{p}_i)$$

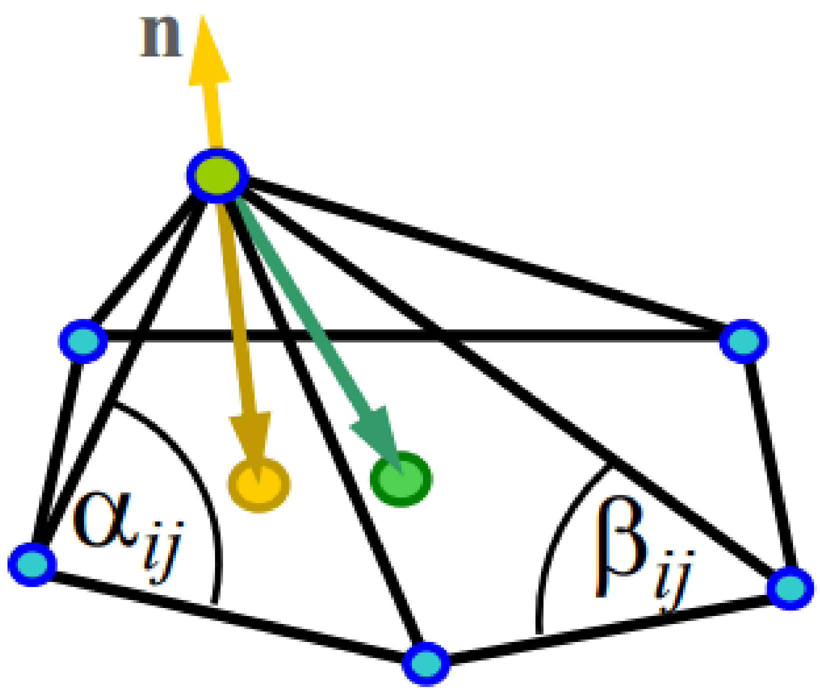
Ignoring the geometry:

$\delta_{\text{uniform}}: W_i = 1,\ w_{ij} = 1/|N(i)|$

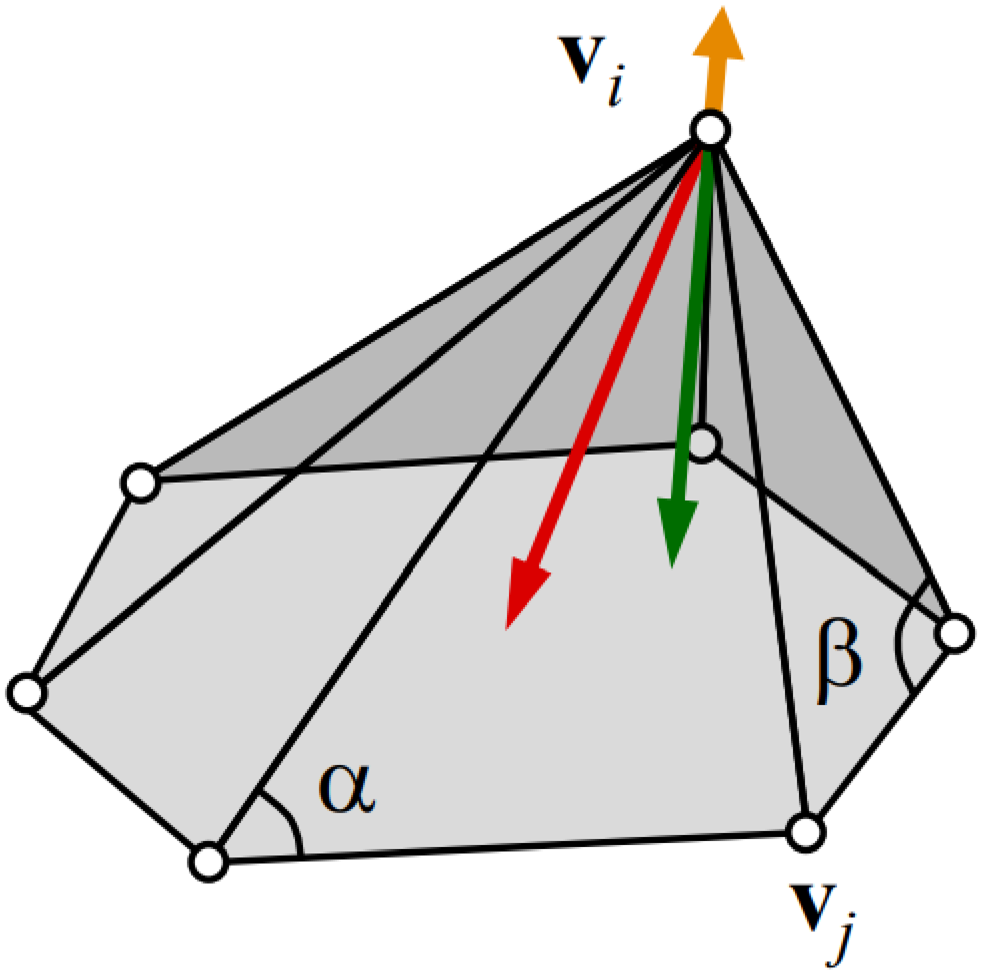
Integrate over Voronoi region of the vertex:

$\delta_{\text{cotan}}: w_{ij} = 0.5(\cot \alpha_{ij} + \cot \beta_{ij})$

$W_i = A_i$



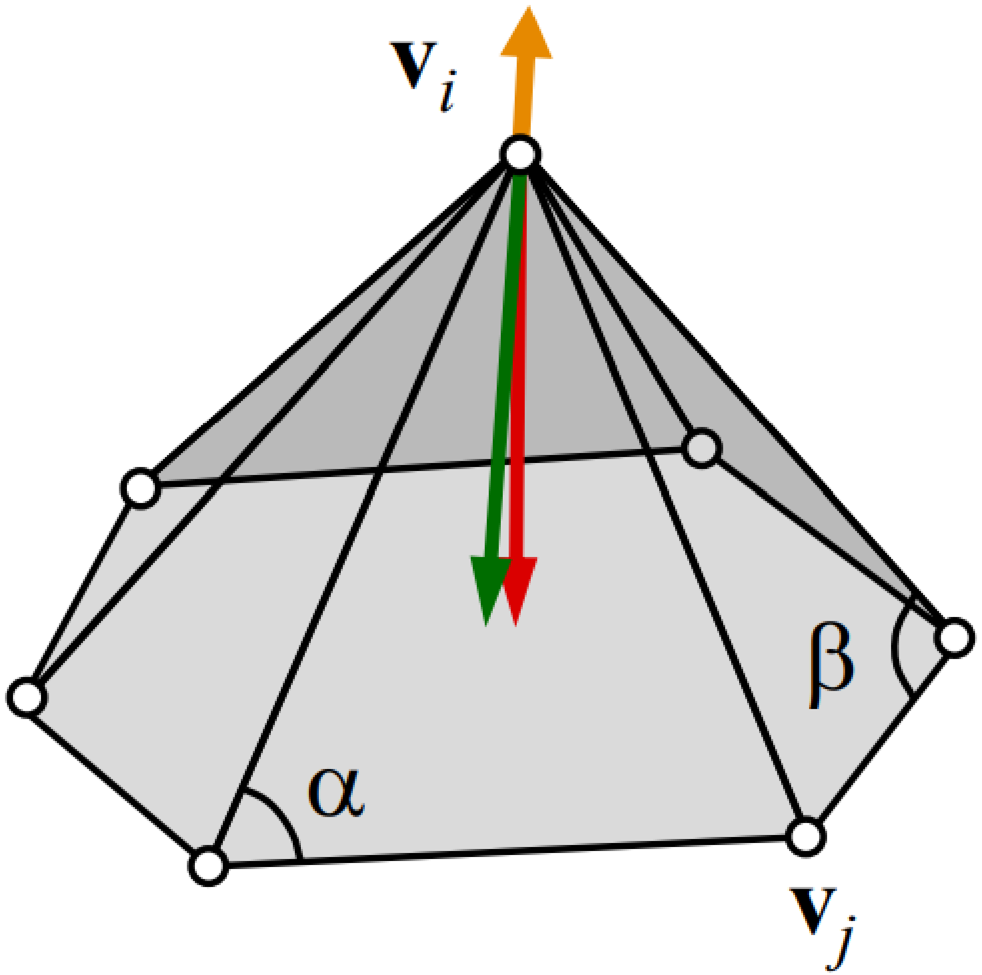
Olga Sorkine-Hornung, ETH, Shape Modeling course



- Uniform Laplacian $\mathbf{L}_u(\mathbf{v}_i)$
- Cotangent Laplacian $\mathbf{L}_c(\mathbf{v}_i)$
- Normal

- For nearly equal edge lengths
  Uniform ≈ Cotangent

Olga Sorkine-Hornung, ETH, Shape Modeling course



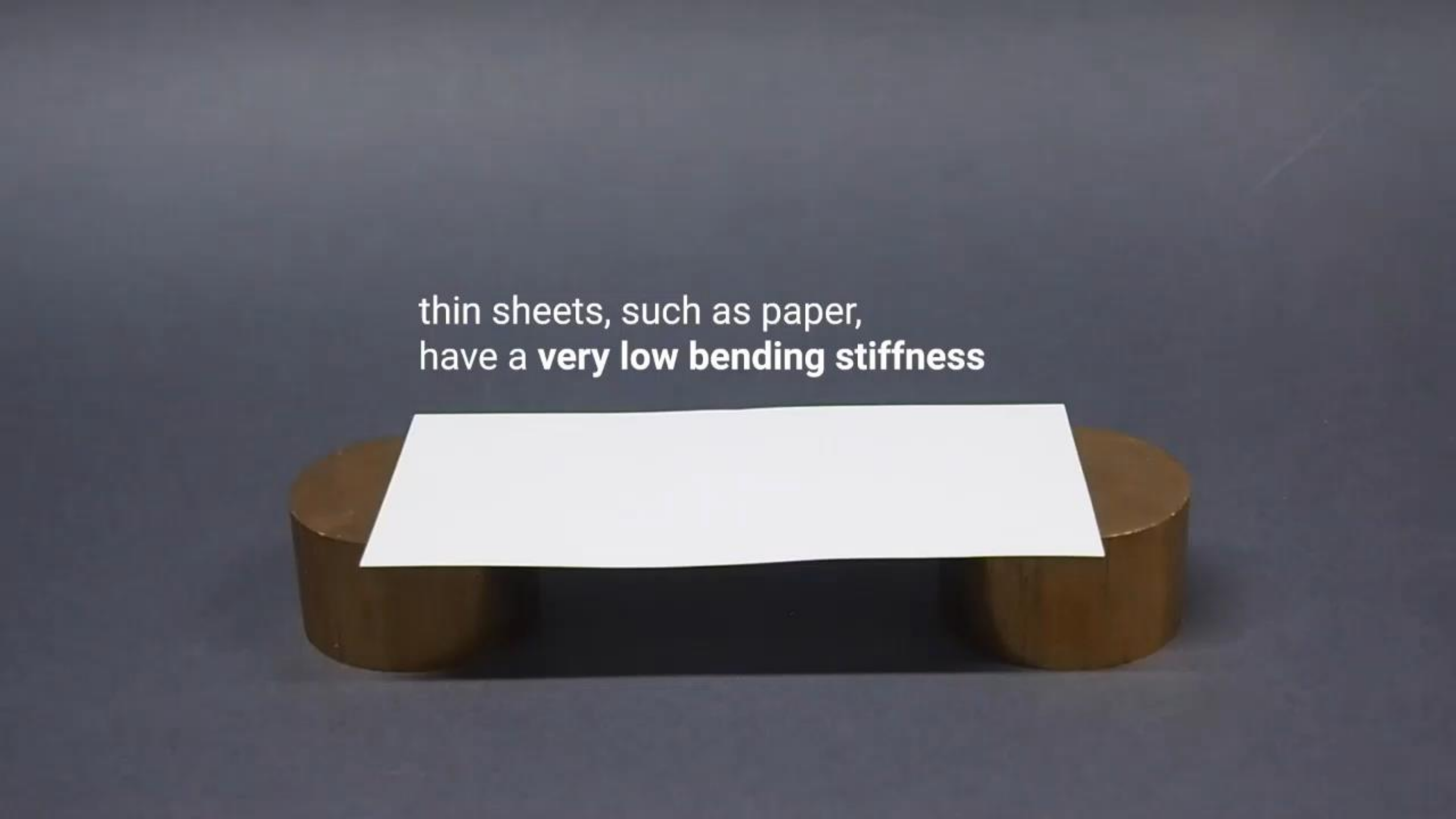
- Uniform Laplacian $\mathbf{L}_u(\mathbf{v}_i)$
- Cotangent Laplacian $\mathbf{L}_c(\mathbf{v}_i)$
- Normal

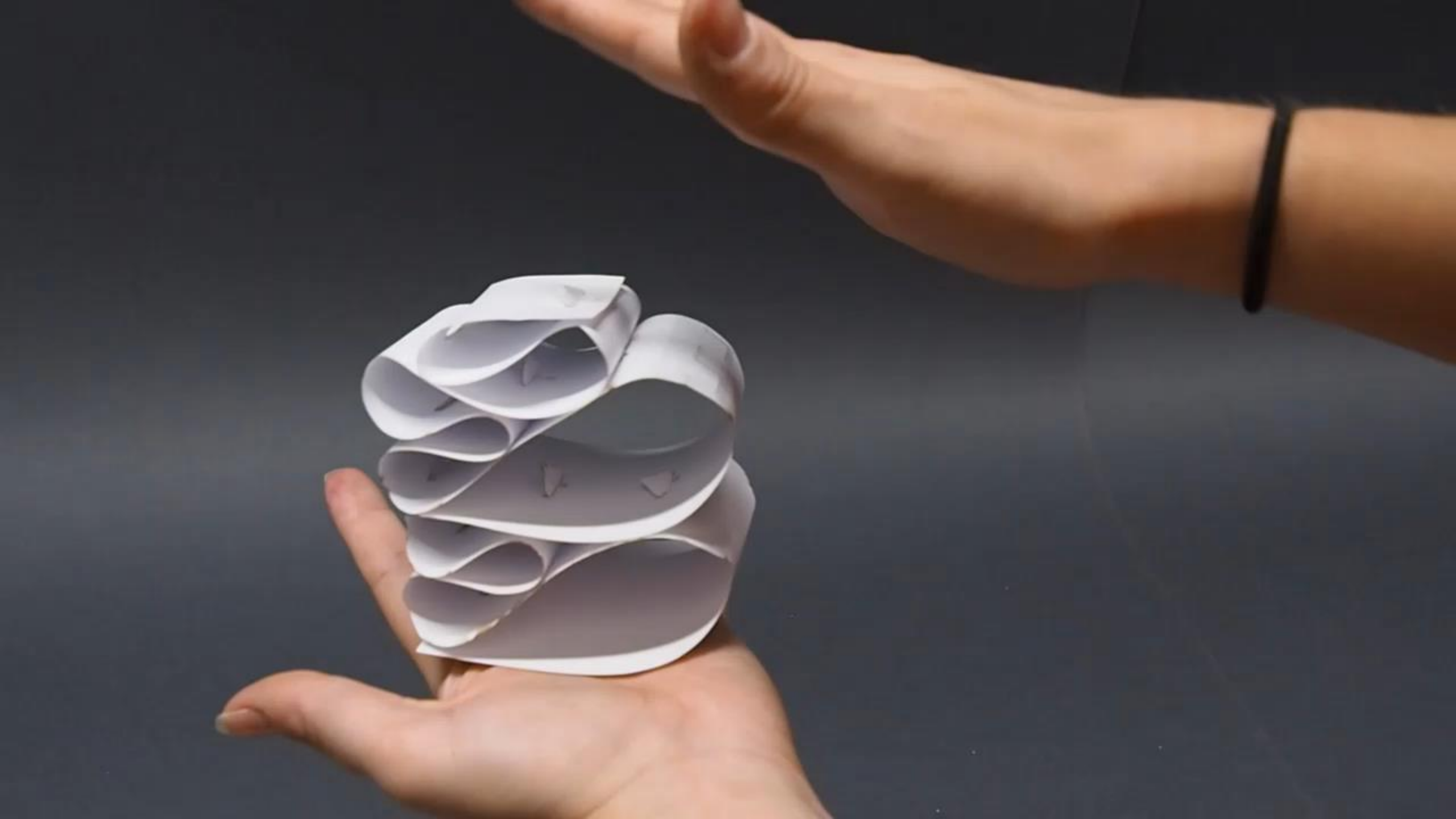
- For nearly equal edge lengths
  Uniform ≈ Cotangent

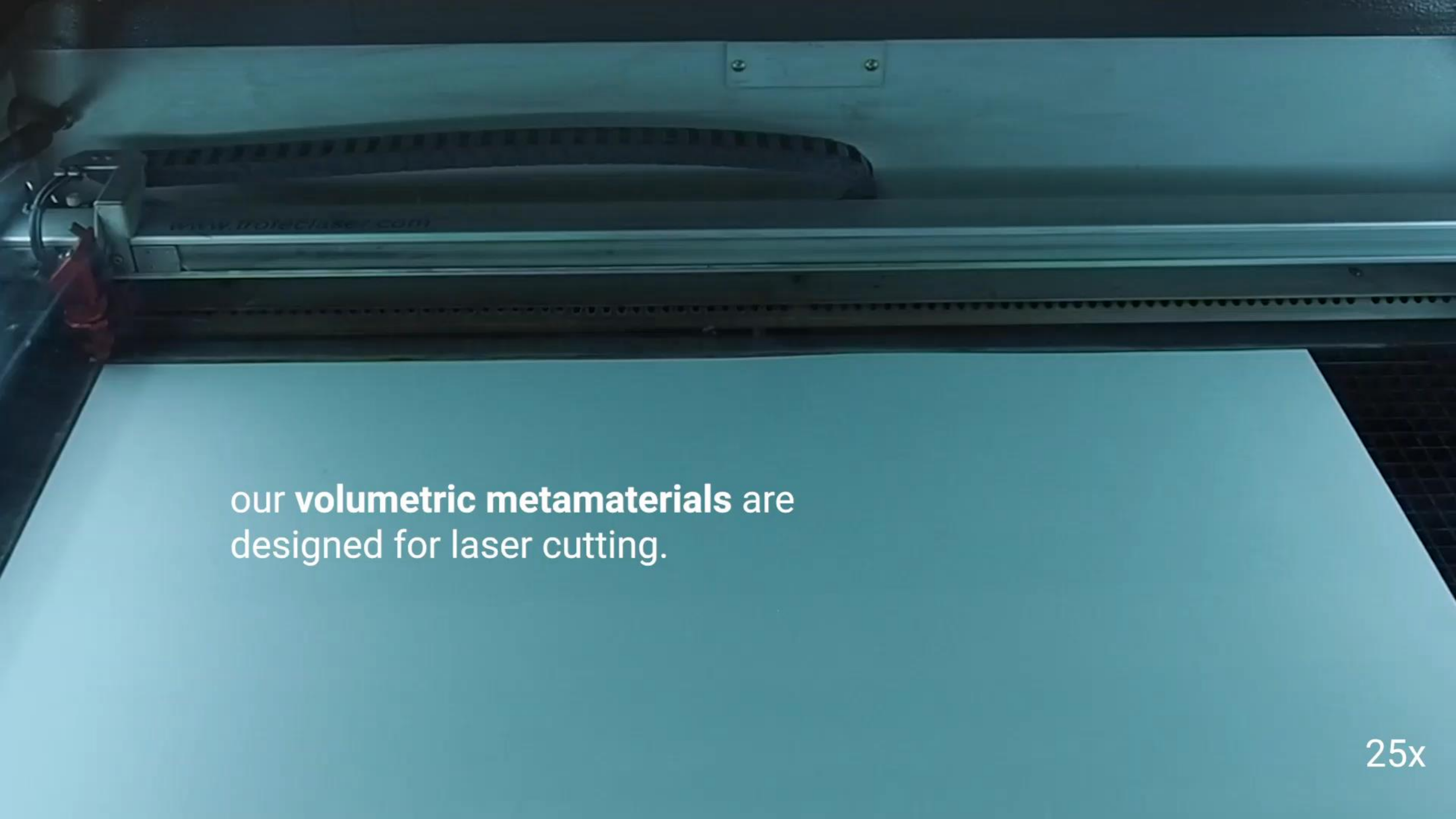
Olga Sorkine-Hornung, ETH, Shape Modeling course

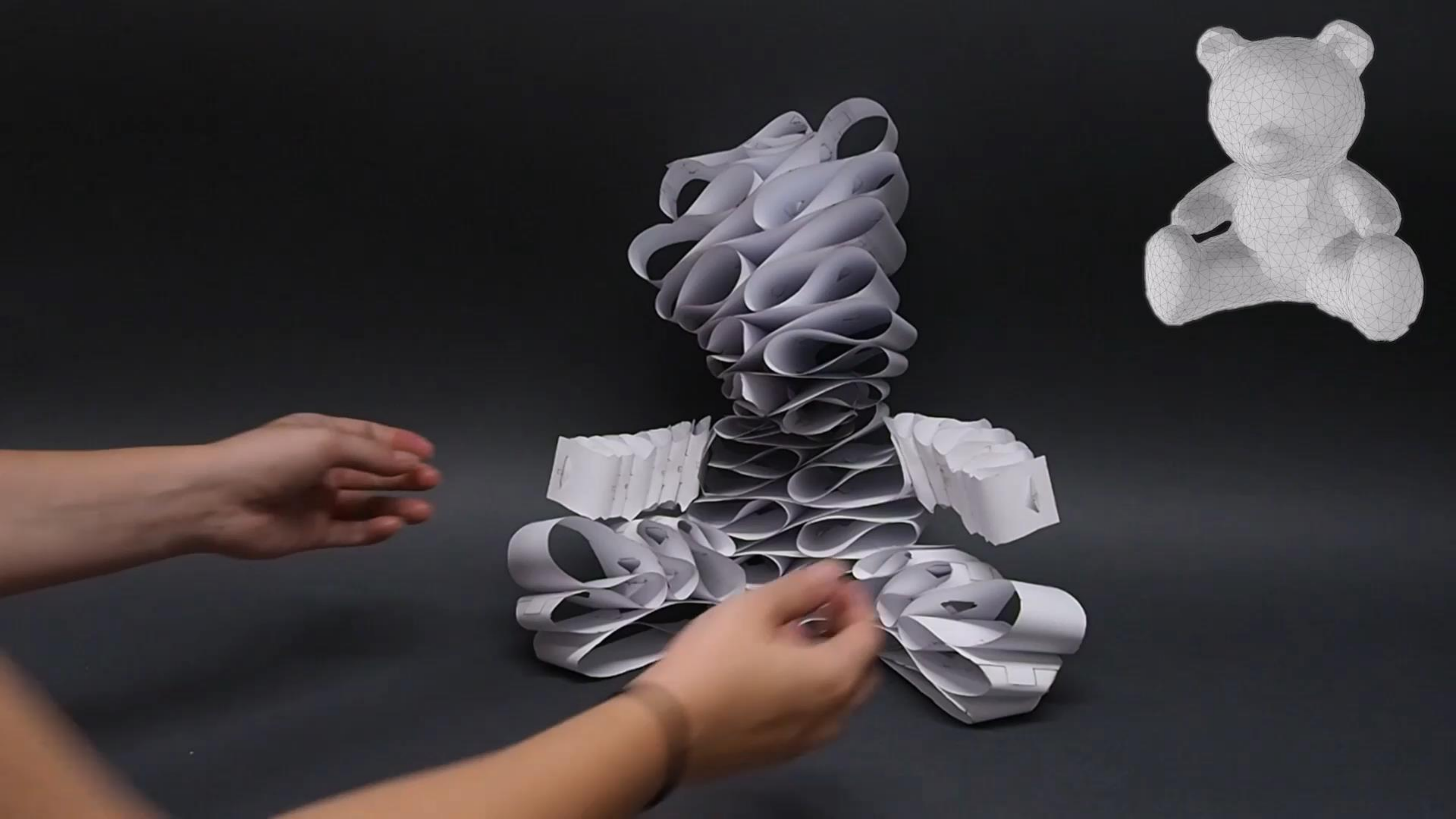
Let's see this in `2_mesh_smoothing`

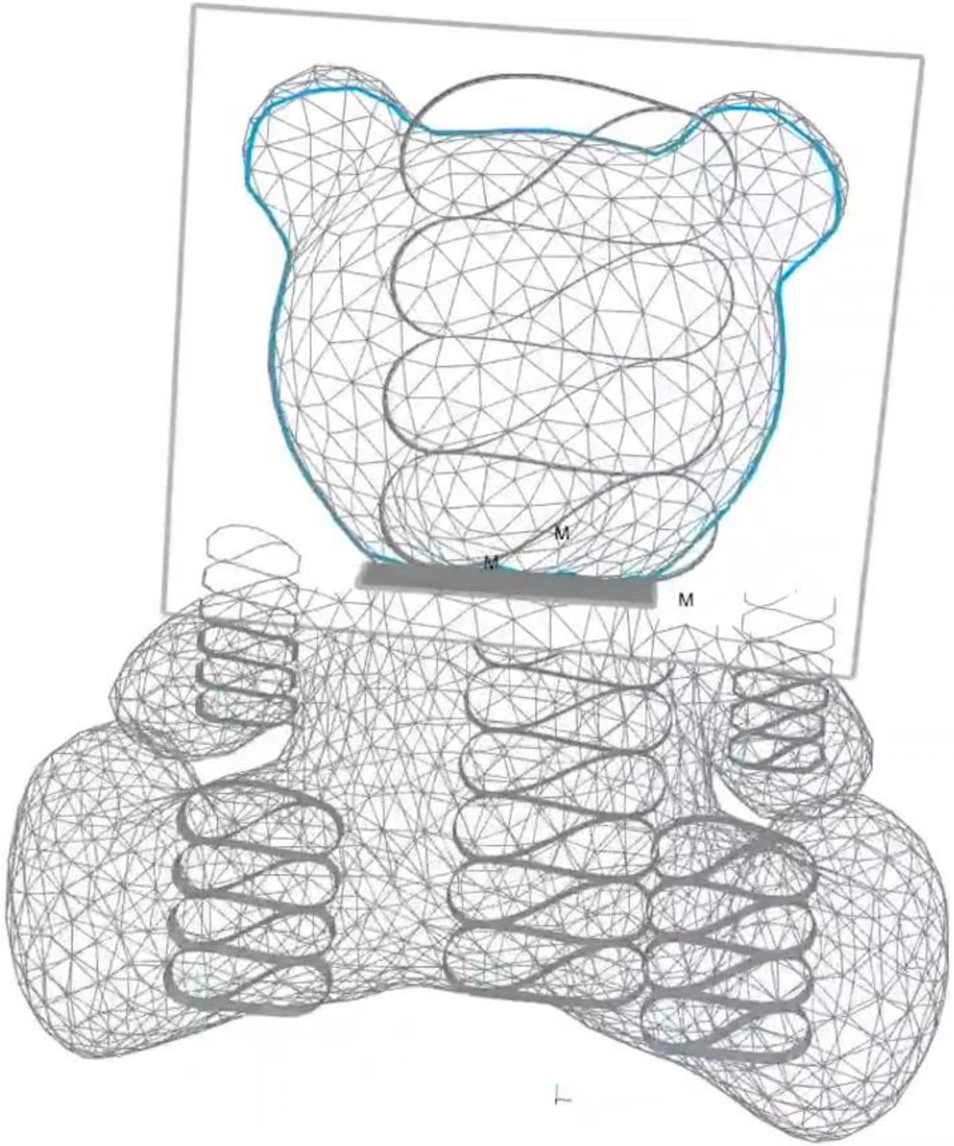
Ruffles

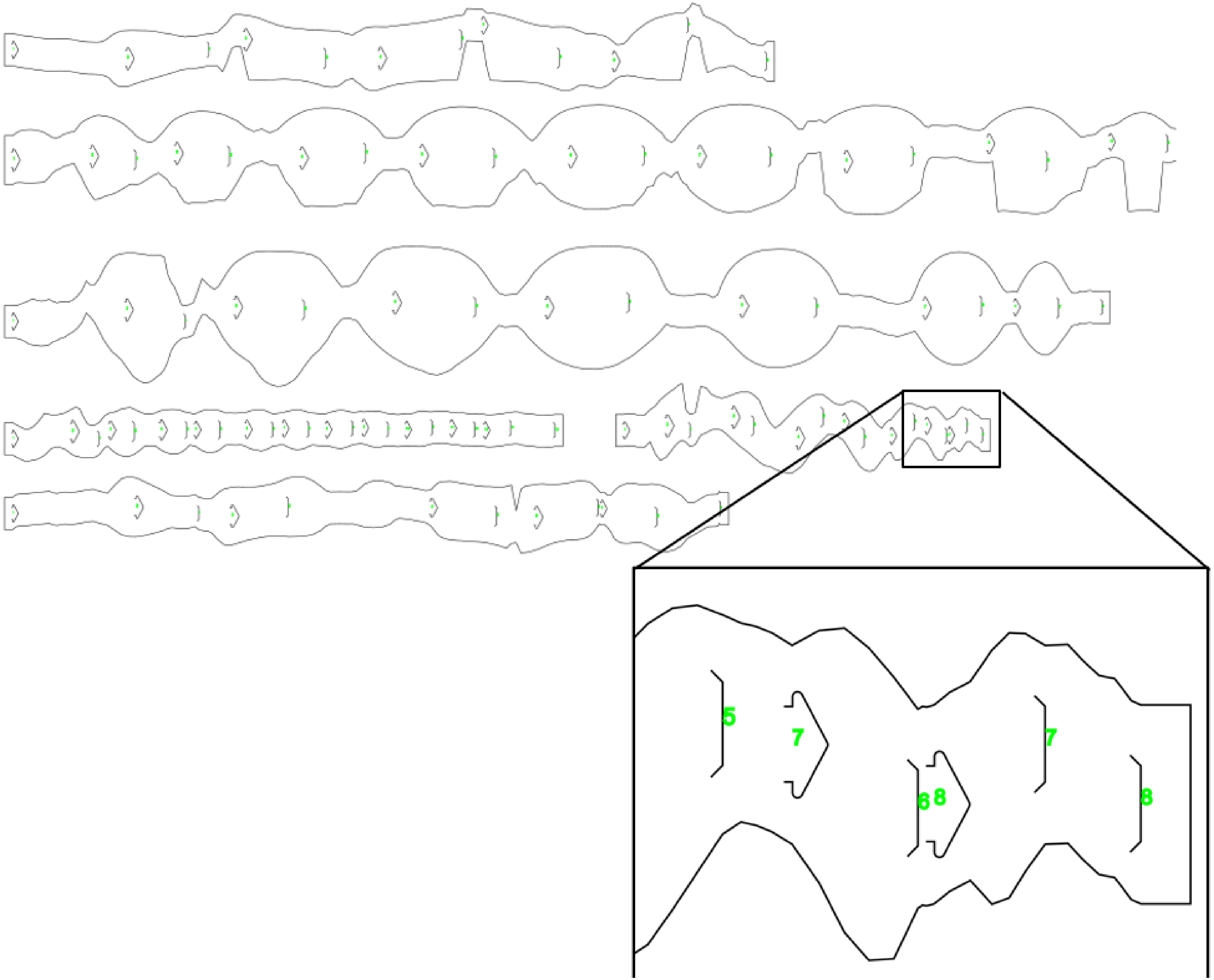

thin sheets, such as paper,
have a **very low bending stiffness**




our **volumetric metamaterials** are designed for laser cutting.
25x

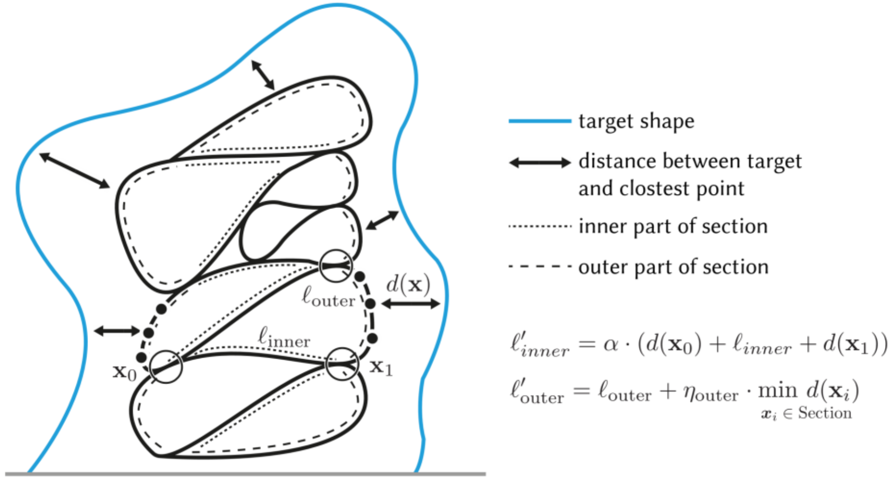


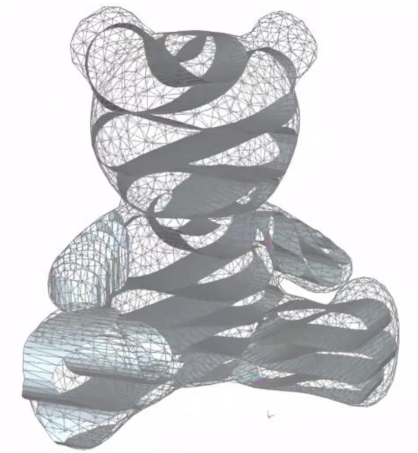

M
M
M


5
7
6
8
7
8

update shape

simulation





properties

1. smooth deformation
   based on Discrete Shells [Grispun et al. 2003]
2. smooth connections
3. no self-intersection
   extening AirMeshes [Müller et al. 2015]

error < ε ?

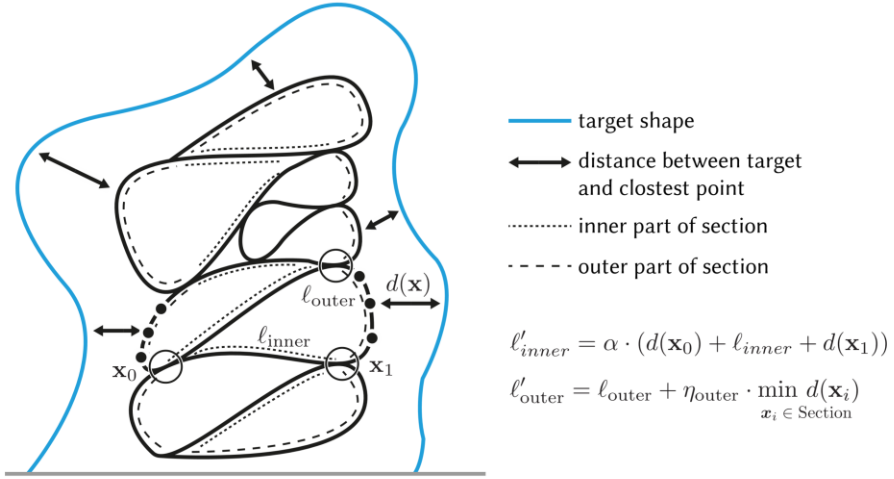
no

yes

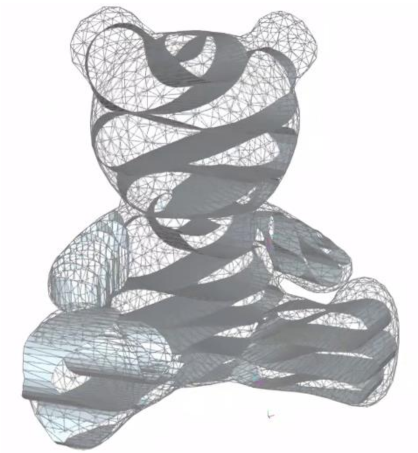
final result

inner: simulation

update shape



simulation



properties

1. smooth deformation
   based on Discrete Shells [Grispun et al. 2003]
2. smooth connections
3. no self-intersection
   extening AirMeshes [Müller et al. 2015]

error < ε ?

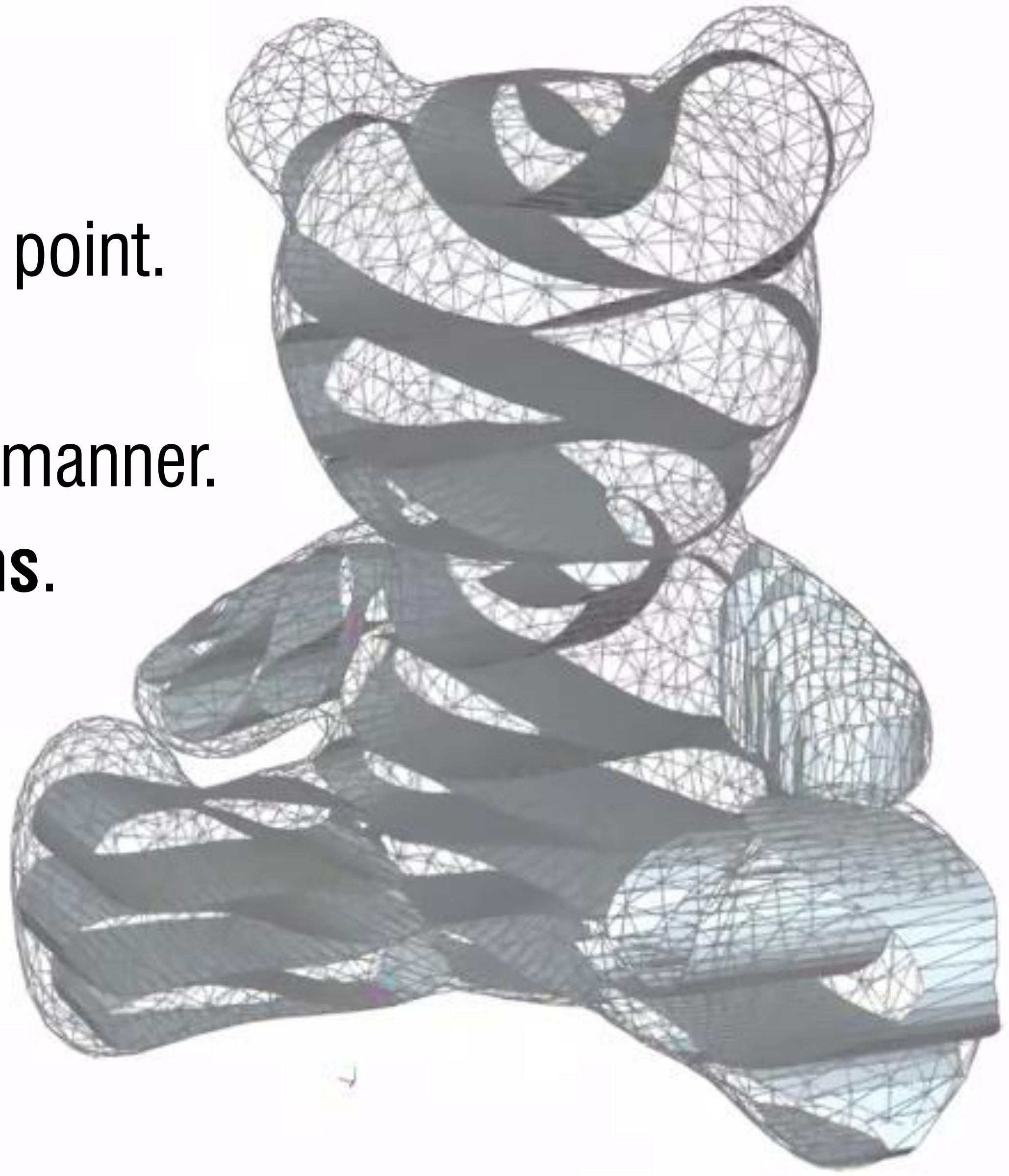
no

yes

final result

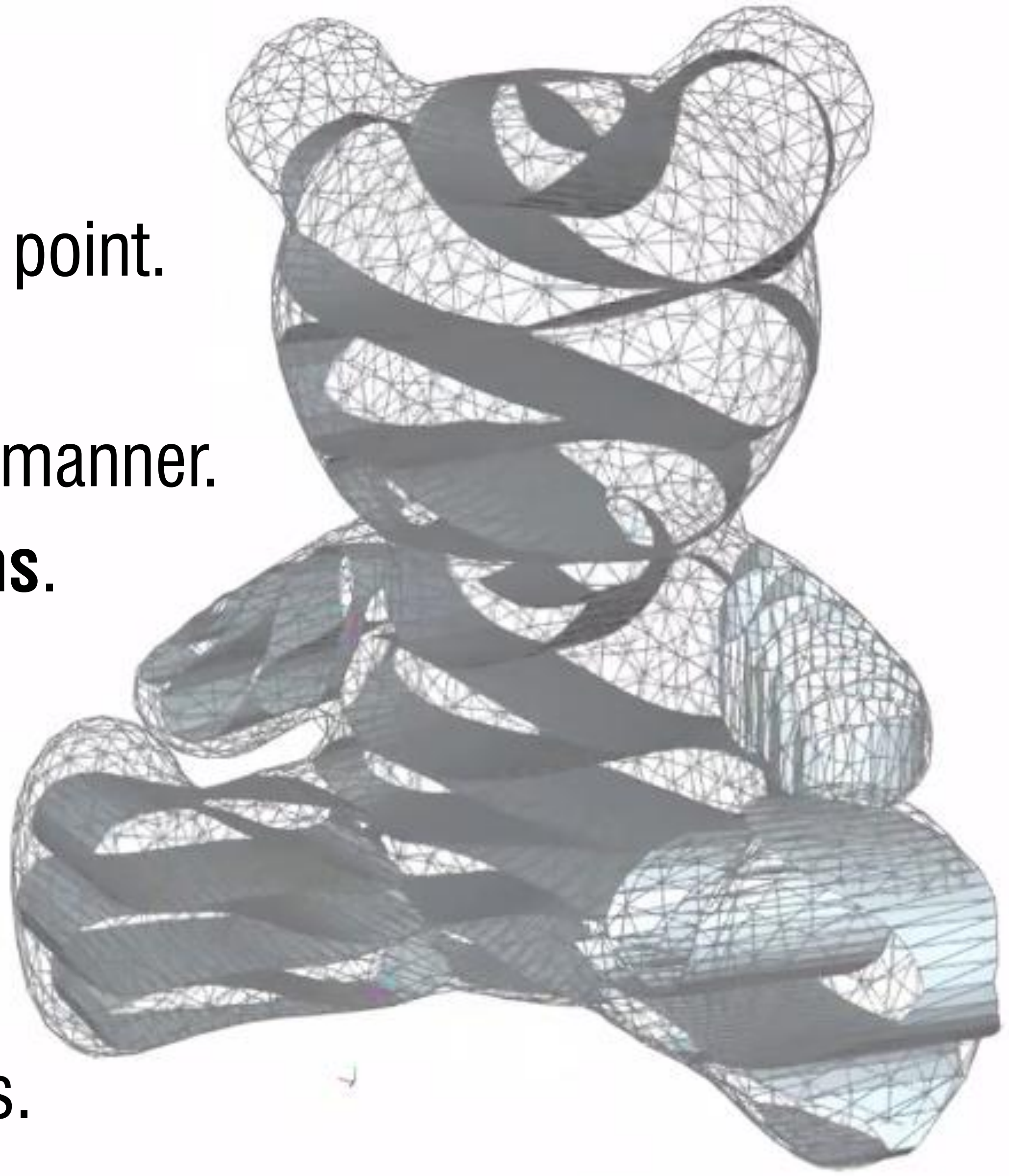
properties

1. The ruffle deforms **smoothly** at any given point.
2. The ruffle does **not stretch**, like paper.
3. The ruffle **connections** meet in a smooth manner.
4. The ruffle does not have **self-intersections**.



properties

1. The ruffle deforms **smoothly** at any given point.
2. The ruffle does **not stretch**, like paper.
3. The ruffle **connections** meet in a smooth manner.
4. The ruffle does not have **self-intersections**.

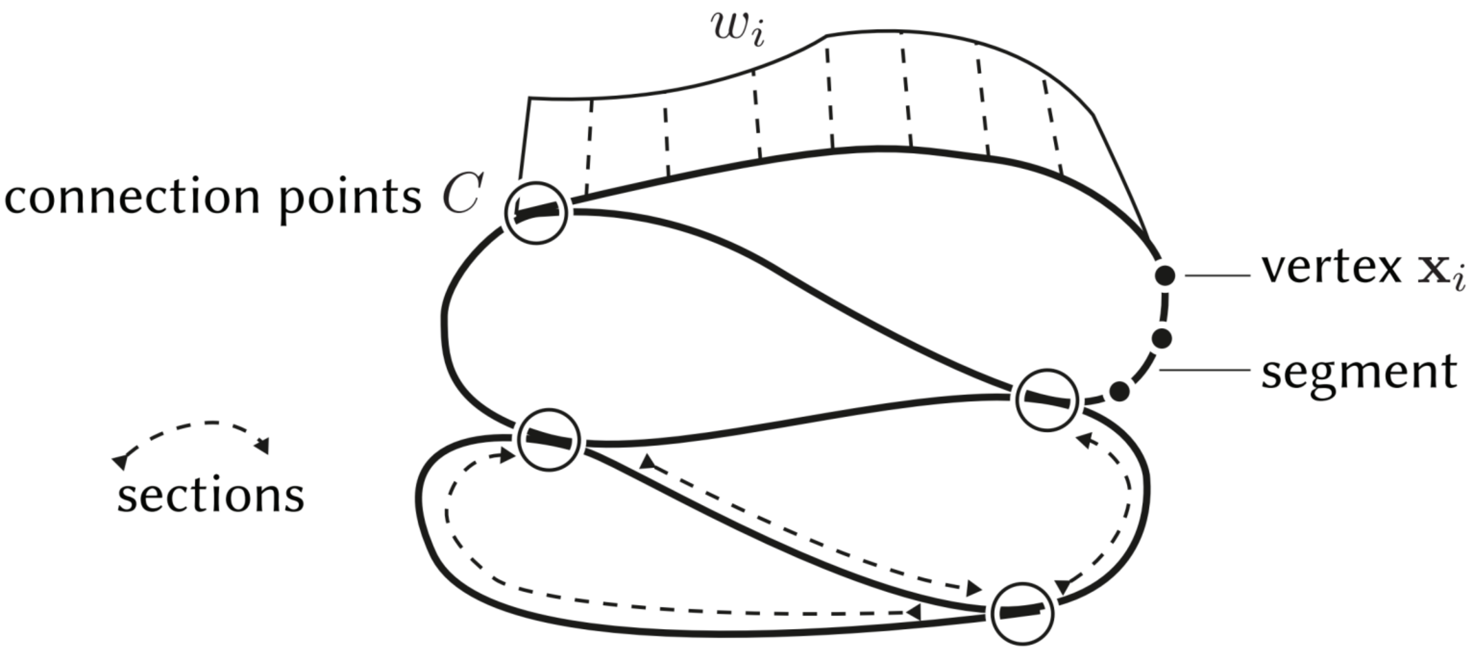


The simulation will be a minimizer

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} E(\mathbf{x})$$

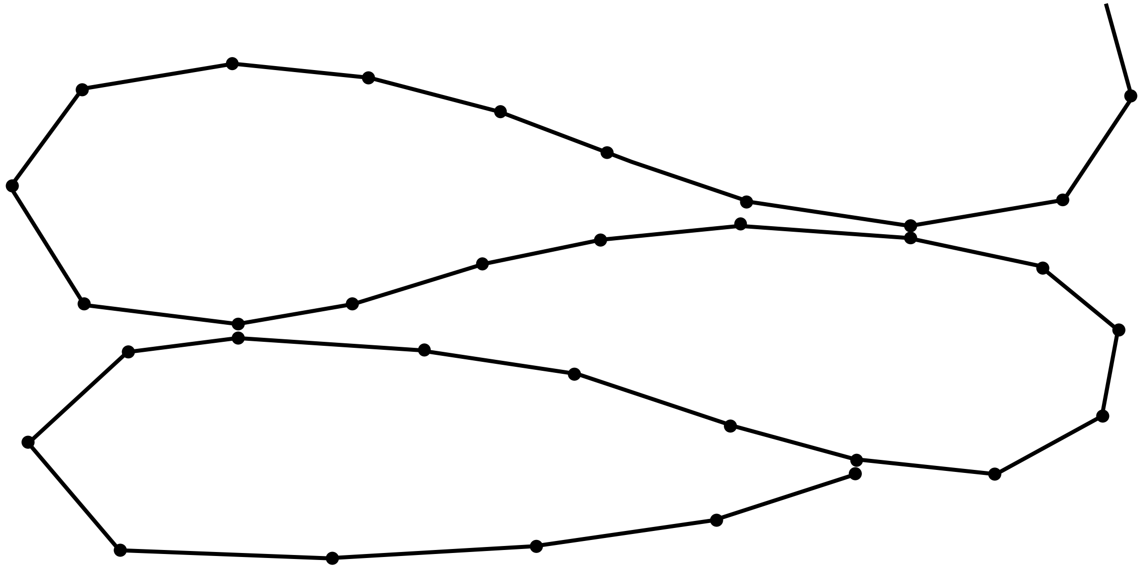
where the energy **E(x)** models the properties.

notation



properties → energies

1. The ruffle deforms **smoothly** at any given point.

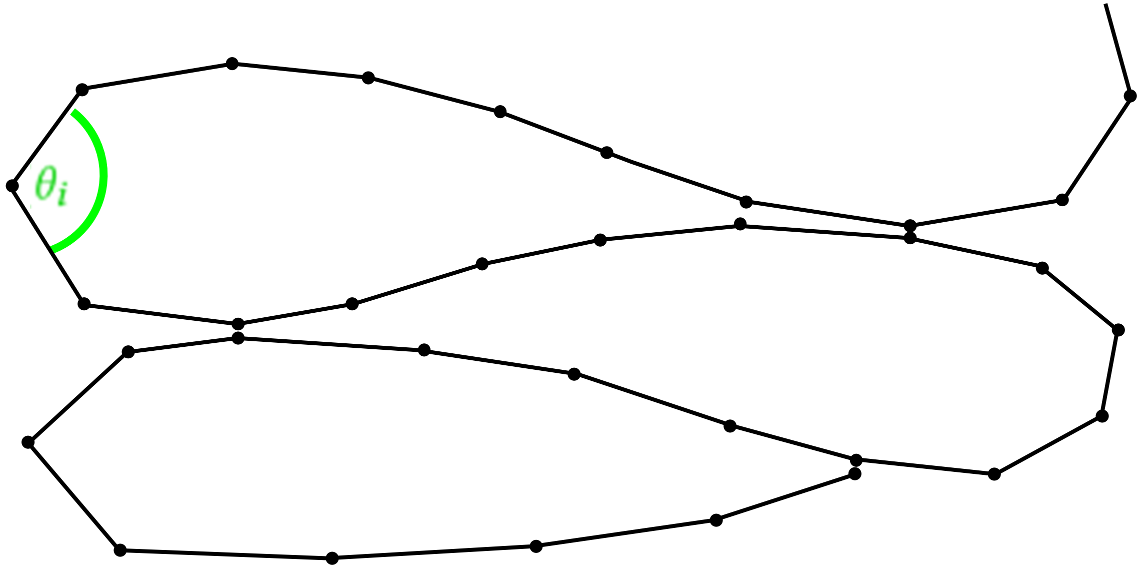


We want to get a small energy when the paper is at rest *(flat)*, and a higher number the more we deform it. How would you model this smoothness energy?

<30sec brainstorming>

properties → energies

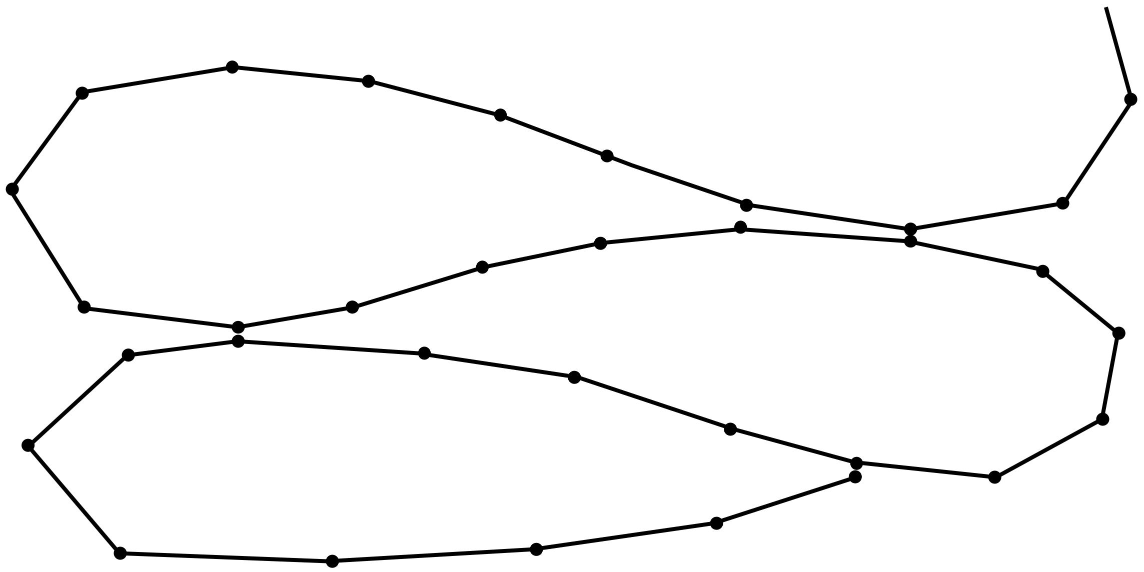
1. The ruffle deforms **smoothly** at any given point.



$$E_B(\mathbf{x}) = \sum_{i=2}^{N-1} (\theta_i - \pi)^2 w_i / \bar{\ell}_i$$

properties → energies

2. The ruffle does **not stretch**, like paper.

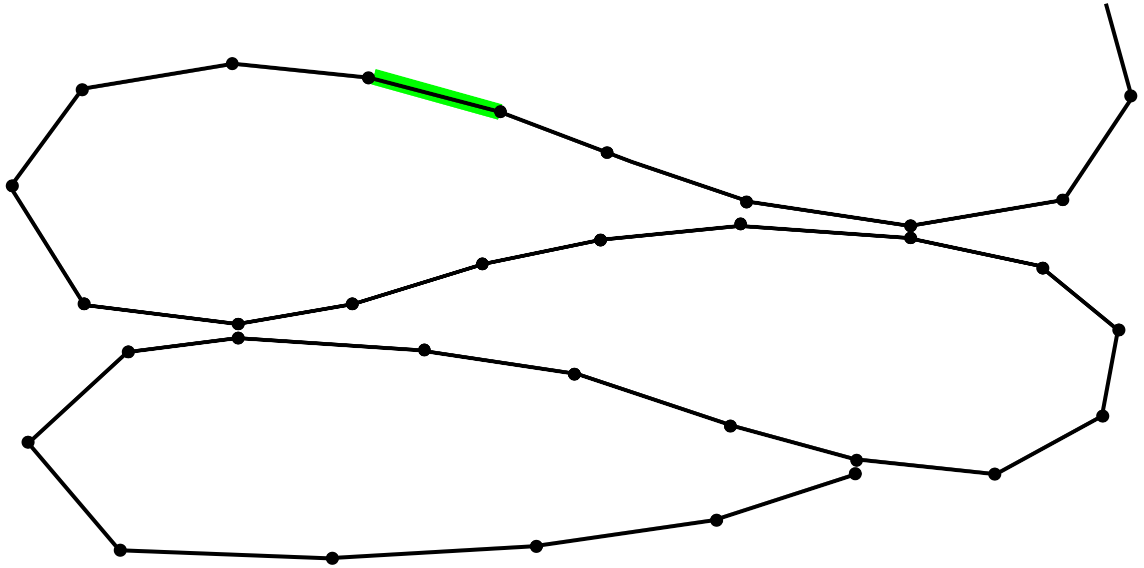


We want: small energy when the *lengths* stay the same, higher energy when deformed. How would you model this stretch energy?

<30sec brainstorming>

properties → energies

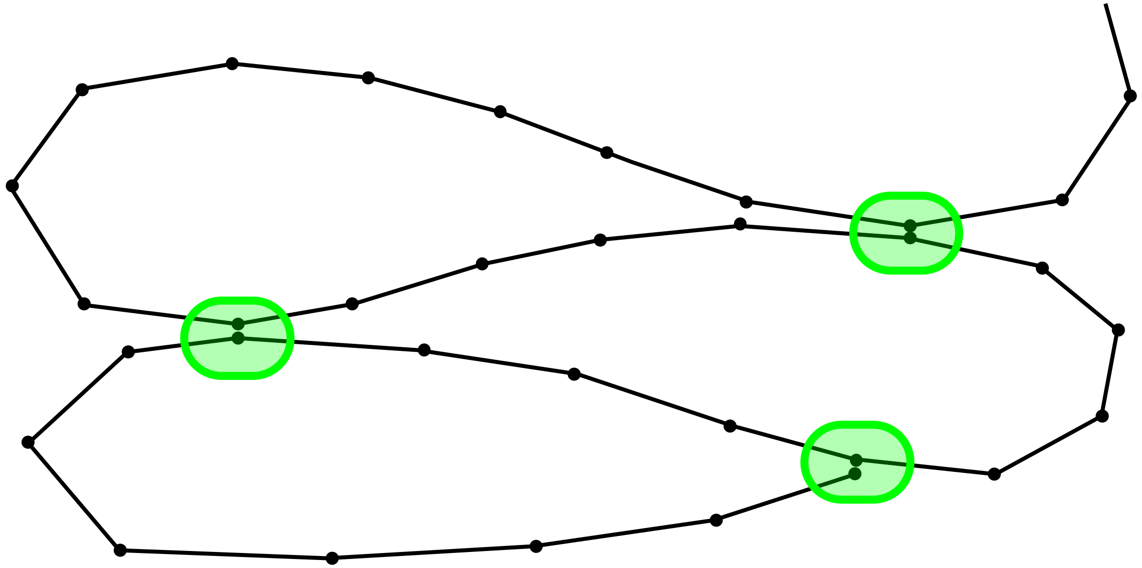
2. The ruffle does **not stretch**, like paper.



$$E_M(\mathbf{x}) = \sum_{i=1}^{N-1} (\|\mathbf{x}_{i+1} - \mathbf{x}_i\| - \ell_i)^2$$

properties → energies

3. The ruffle **connections** meet in a smooth manner.

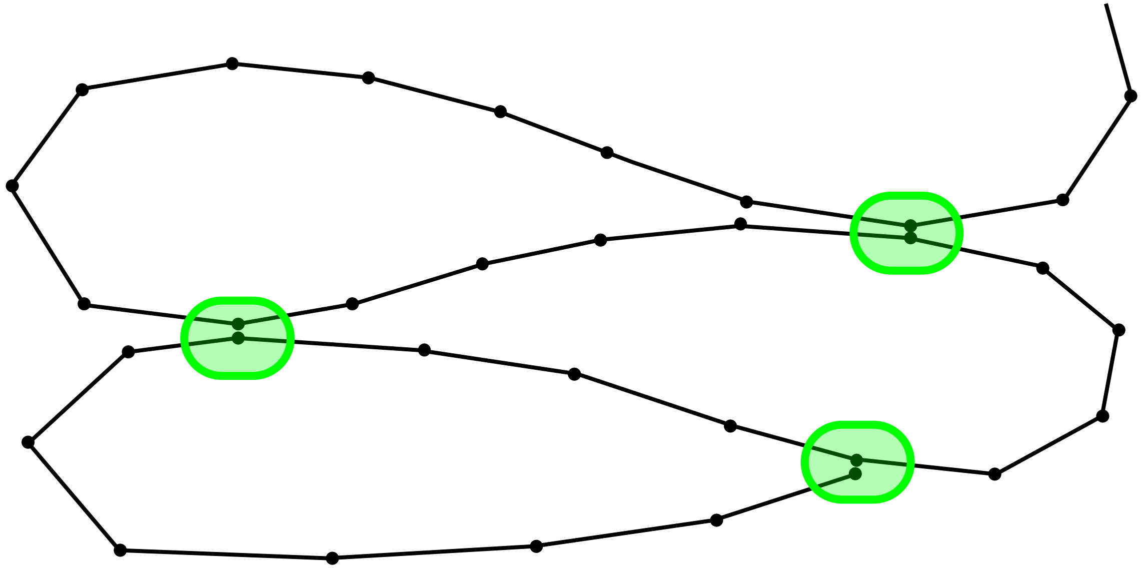


We want: small energy when the *tangents* are the same, higher energy when different. How would you model this stretch energy?

<30sec brainstorming>

properties → energies

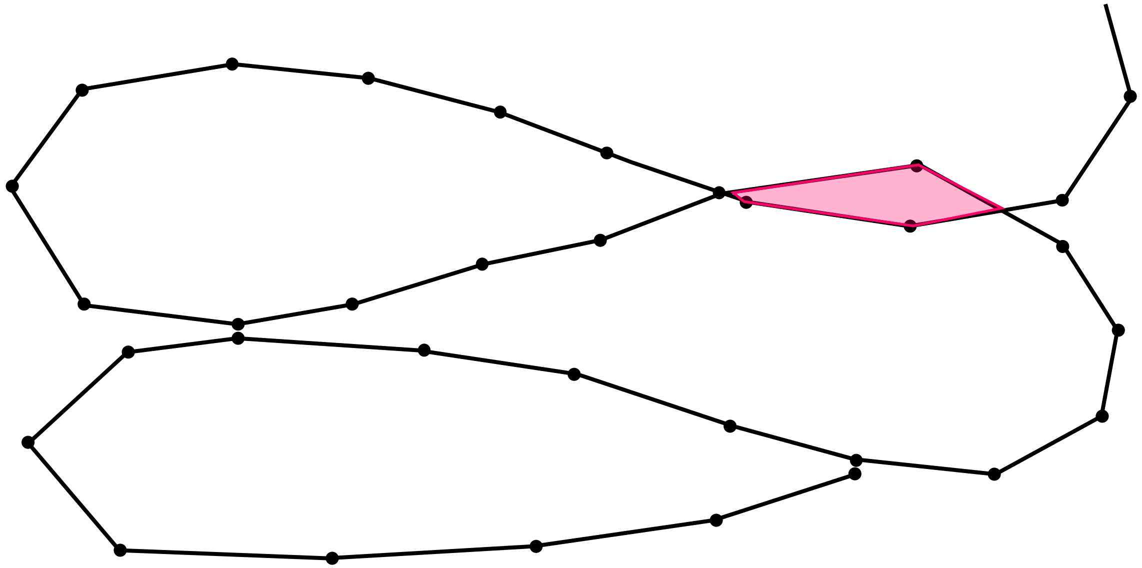
3. The ruffle **connections** meet in a smooth manner.



$$E_C(\mathbf{x}) = \sum_{(i,j)\in C} (\theta_{ij}^{+} - \pi)^2 w_i / \bar{\ell}_{ij}^{+} + (\theta_{ij}^{-} - \pi)^2 w_i / \bar{\ell}_{ij}^{-}$$

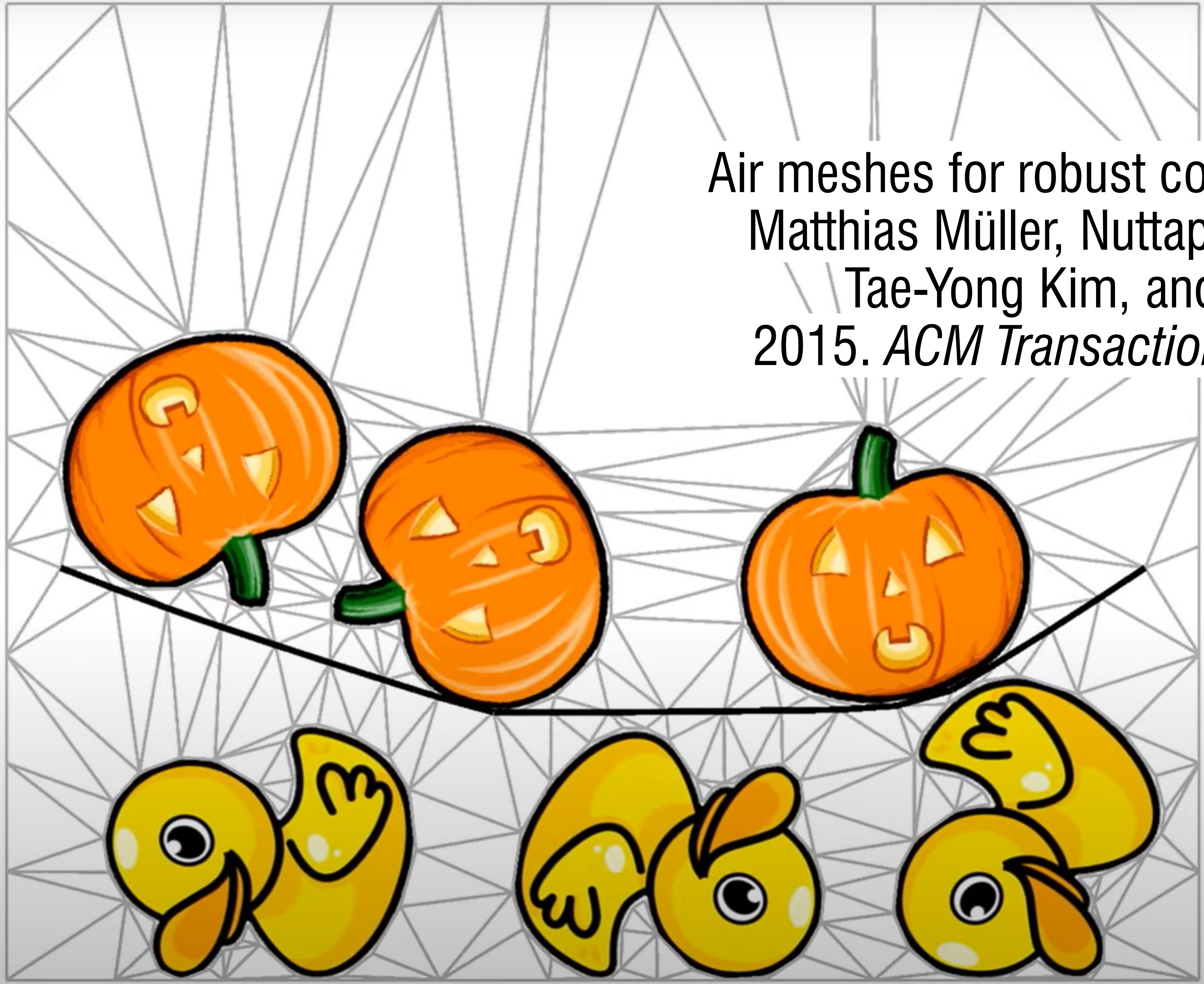
properties → energies

4. The ruffle does not have **self-intersections**.



We want: small energy at no intersection, high energy otherwise. This is a long-standing challenge in engineering.

We use...



Video

Air meshes for robust collision handling.
Matthias Müller, Nuttapong Chentanez,
Tae-Yong Kim, and Miles Macklin.
2015. *ACM Transactions on Graphics.*

Simulation

Putting it all together

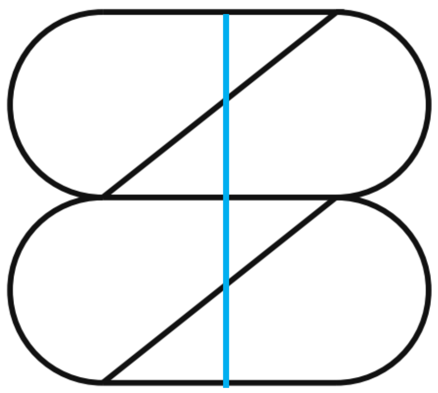
$$E(\mathbf{x}) = k_M E_M(\mathbf{x}) + k_B(E_B(\mathbf{x}) + E_C(\mathbf{x})) + k_{AM} E_{AM}(\mathbf{x})$$

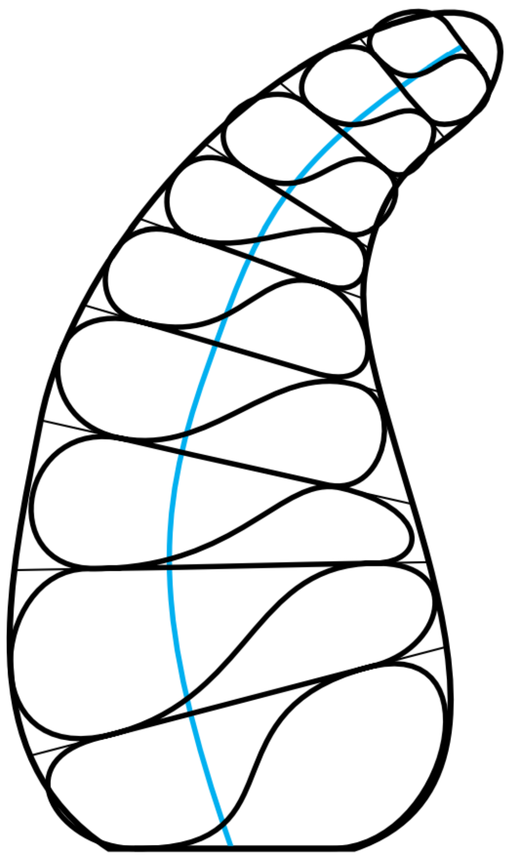
and solve for

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} E(\mathbf{x})$$

Initial guess

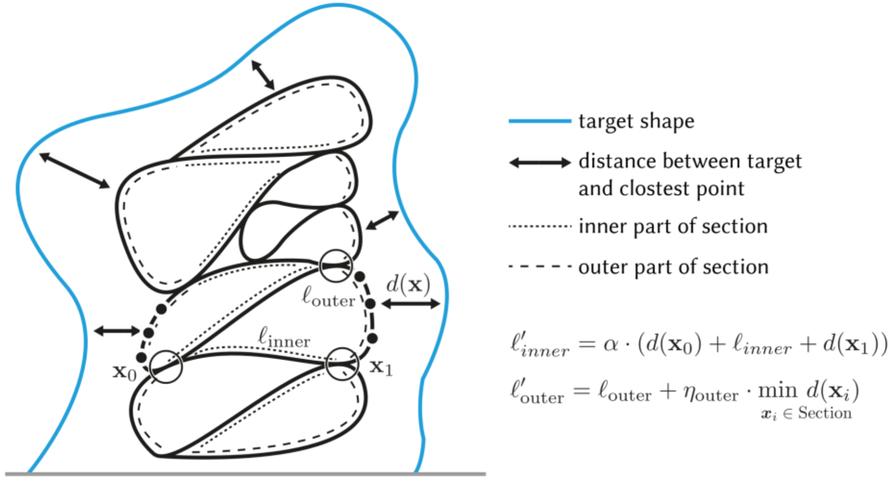
Ruffle construction by *generator curve*



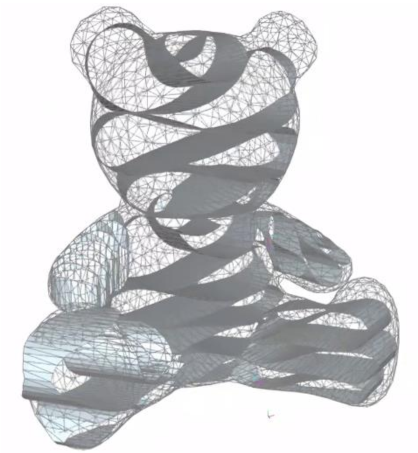


outer: optimization

update shape



simulation



properties

1. smooth deformation
   based on Discrete Shells [Grispun et al. 2003]
2. smooth connections
3. no self-intersection
   extening AirMeshes [Müller et al. 2015]

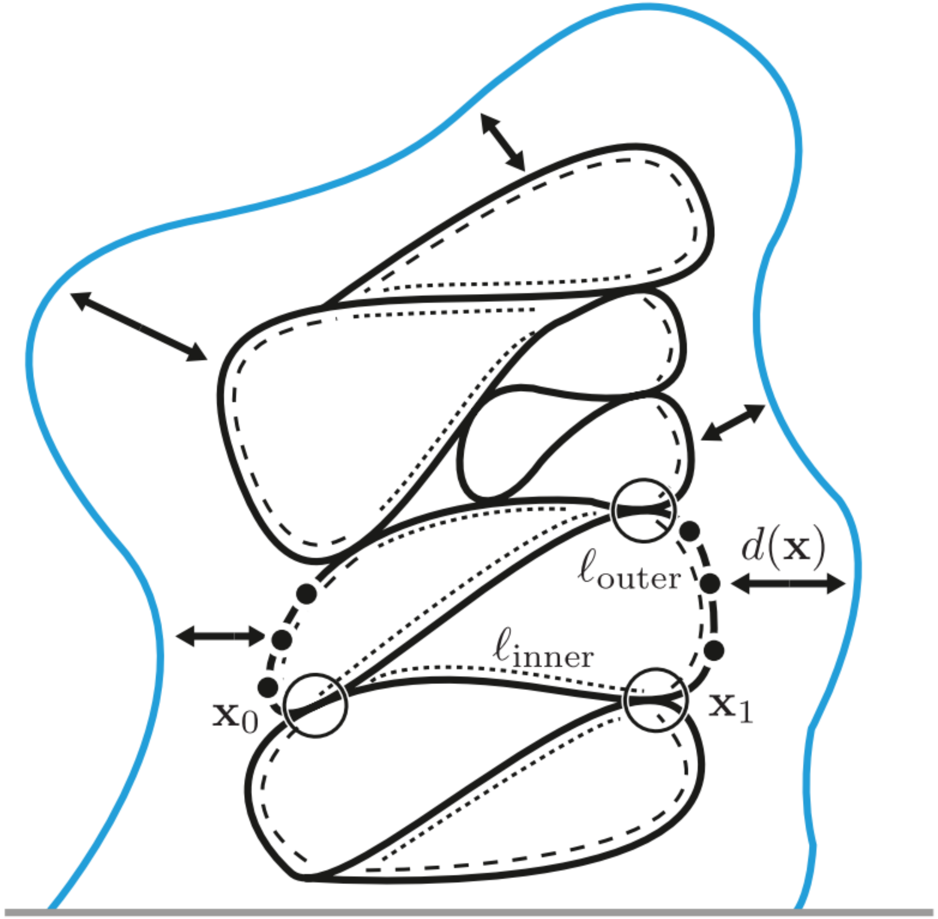
error < ε ?

no

yes

final result

shape approximation



target shape

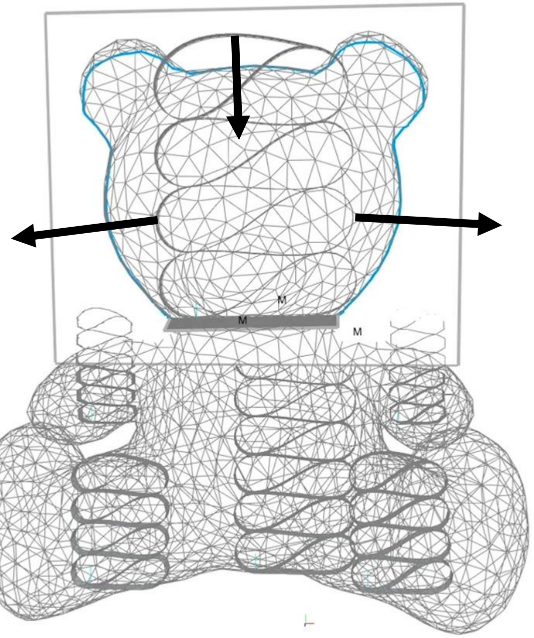
distance between target and clostest point

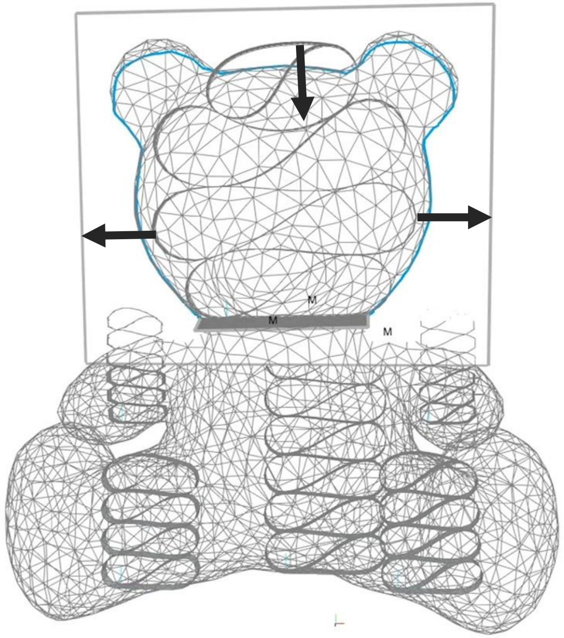
inner part of section

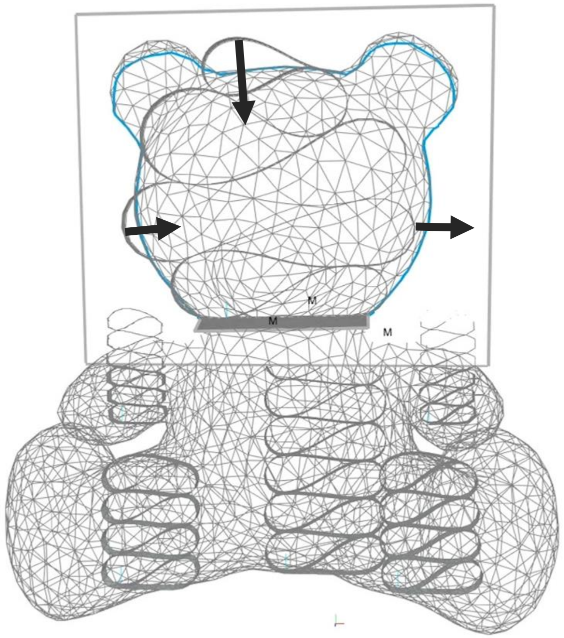
outer part of section

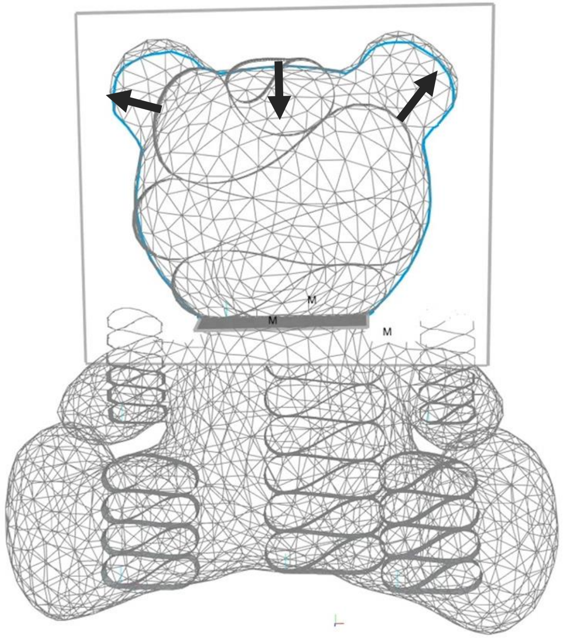
$$\ell'_{inner} = \alpha \cdot (d(\mathbf{x}_0) + \ell_{inner} + d(\mathbf{x}_1))$$

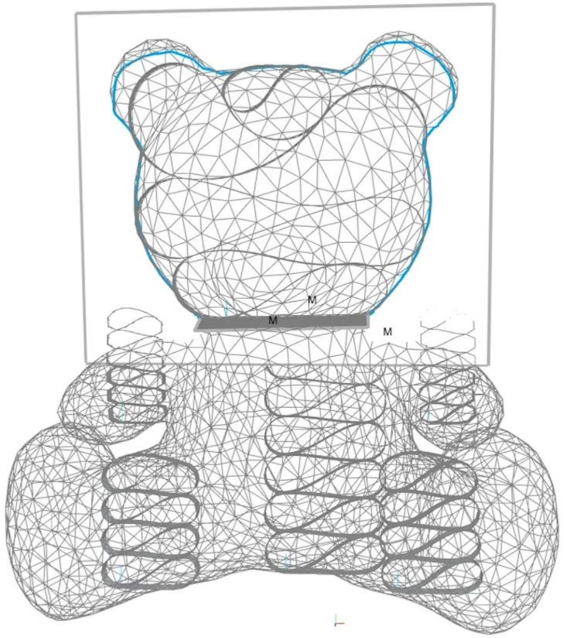
$$\ell'_{\text{outer}} = \ell_{\text{outer}} + \eta_{\text{outer}} \cdot \min_{\boldsymbol{x}_i \in \text{Section}} d(\mathbf{x}_i)$$


M
M
M


M
M
M


M
M
M


M
M
M


M
M
M

scope

current

optimize shape

→

next

optimize for **forces**

summary

Summary

Geometry is awesome

Generate & edit meshes, send to 3D printer → physical objects!

Generate complex materials

Useful tools & data

MeshLab

Open Flipper

Meshes: https://github.com/libigl/libigl-tutorial-data end