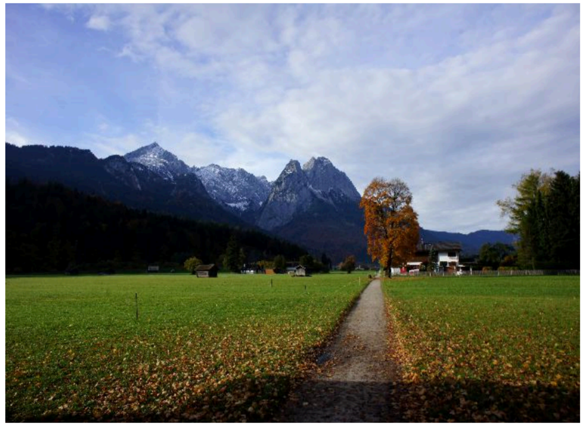
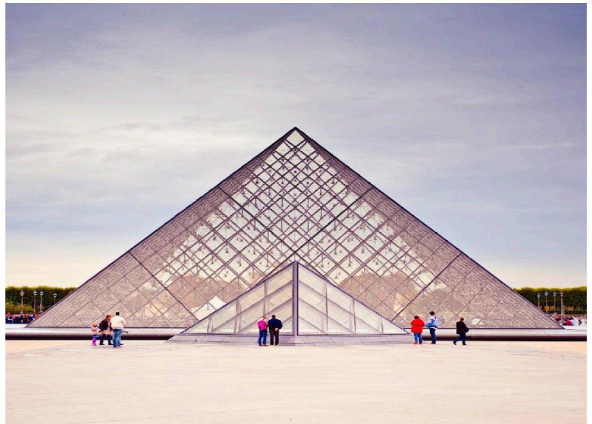
Cubic Stylization

HSUEH-TI DEREK LIU, JACOBSON

Introduction of the problem

2D image stylization:





source: Li Y, Liu M Y, Li X, et al. A closed-form solution to photorealistic image stylization.

Introduction of the problem

3D Mesh stylization: ?



Style of cubic sculptures

Introduction of the problem

3D Mesh stylization:

Input

Output

Approach

- **Energy Function**:

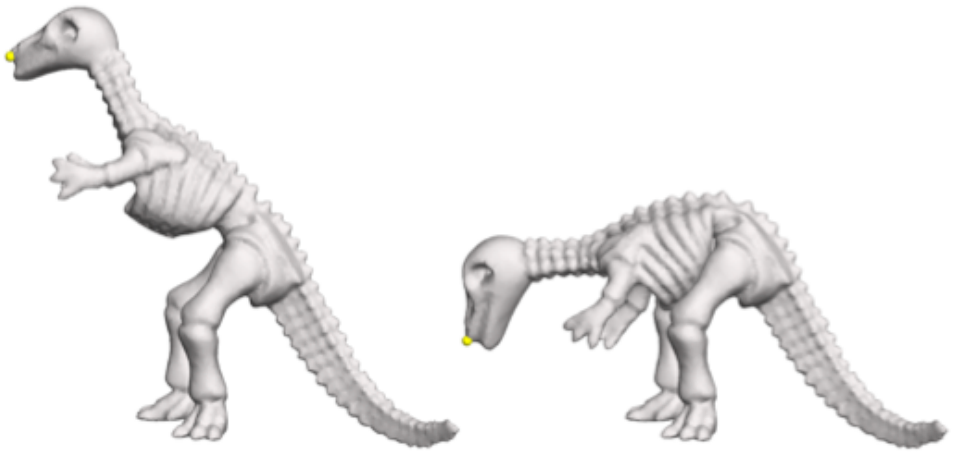
$$\underset{\widetilde{\mathrm{V}},\{\mathrm{R}_i\}}{\text{minimize}} \sum_{i\in V} \sum_{j\in N(i)} \underbrace{\frac{w_{ij}}{2} \left\| \mathrm{R}_i \mathrm{d}_{ij} - \widetilde{\mathrm{d}}_{ij} \right\|_F^2}_{\text{ARAP}} + \underbrace{\lambda a_i \left\| \mathrm{R}_i \hat{\mathrm{n}}_i \right\|_1}_{\text{CUBENESS}}$$

1. As Rigid As Possible (ARAP) : Preserve the local structure. [Sorkine and Alexa 2007]

2. CUBENESS: Minimize the l^1 norm to encourage the rotated normals to align with the axes

ARAP: As-rigid-as-possible

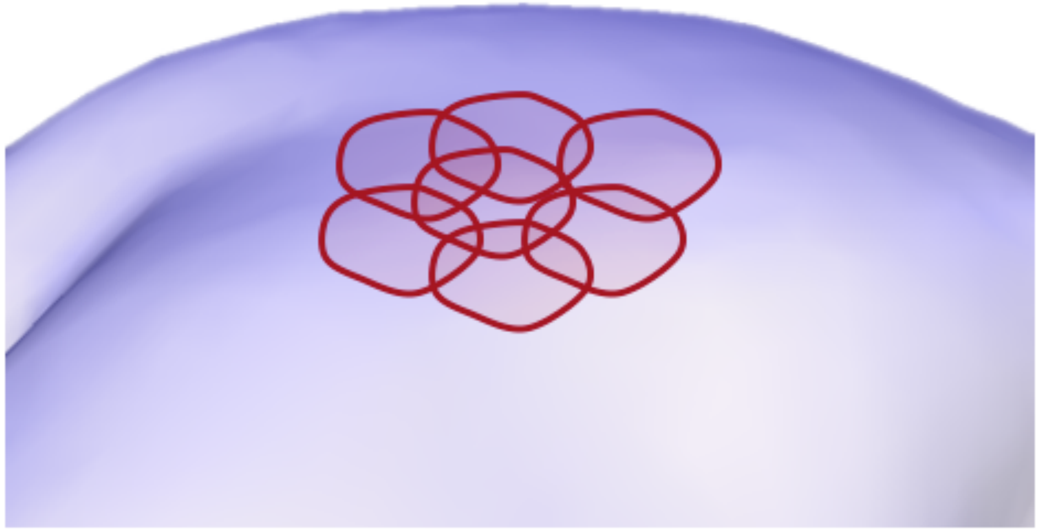
Sorkine, Olga, and Marc Alexa. "As-rigid-as-possible surface modeling."   2007.



source: Sorkine, Olga, and Marc Alexa. "As-rigid-as-possible surface modeling."   2007.

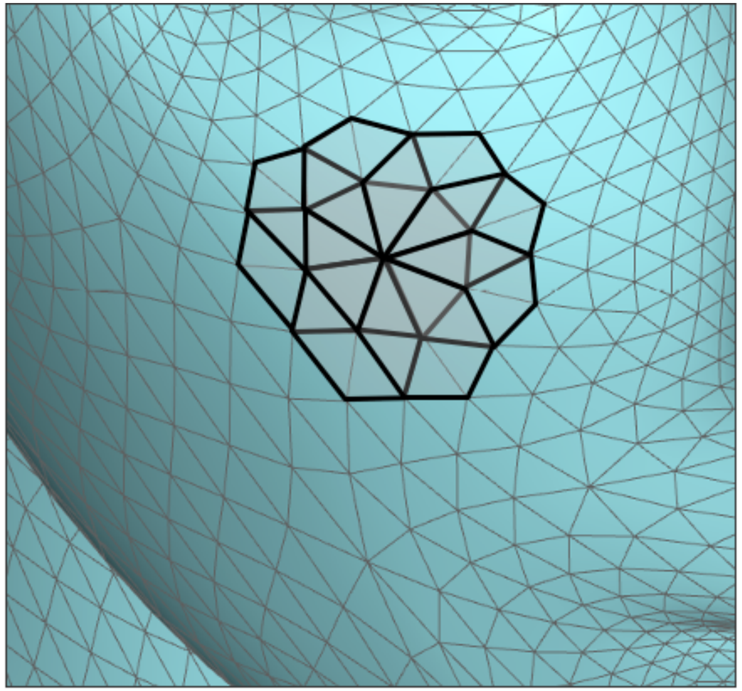
ARAP: As-rigid-as-possible

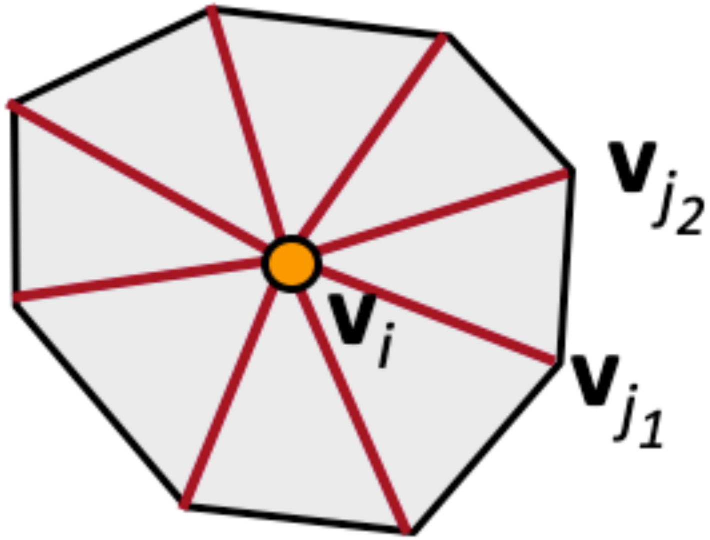
- Preserve small part covering the surface
  - Cells should overlap to prevent shearing at the cells boundaries



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

ARAP: As-rigid-as-possible



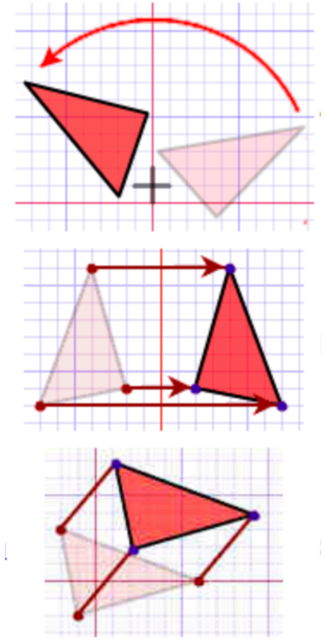


source: https://igl.ethz.ch/projects/ARAP/

How to preserve the details?

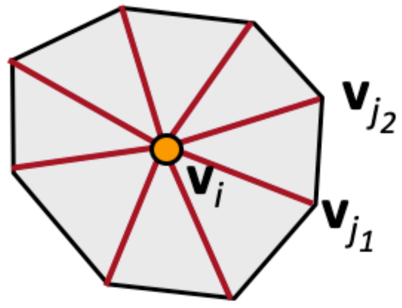
- **Rigid transformation!**

Rigid transformation preserves the distance between every pair of points.



if deformation is rigid:

$$(\mathbf{v}'_i - \mathbf{v}'_j) = R_i(\mathbf{v}_i - \mathbf{v}_j) \ , \ \forall j \in \mathcal{N}(i)$$



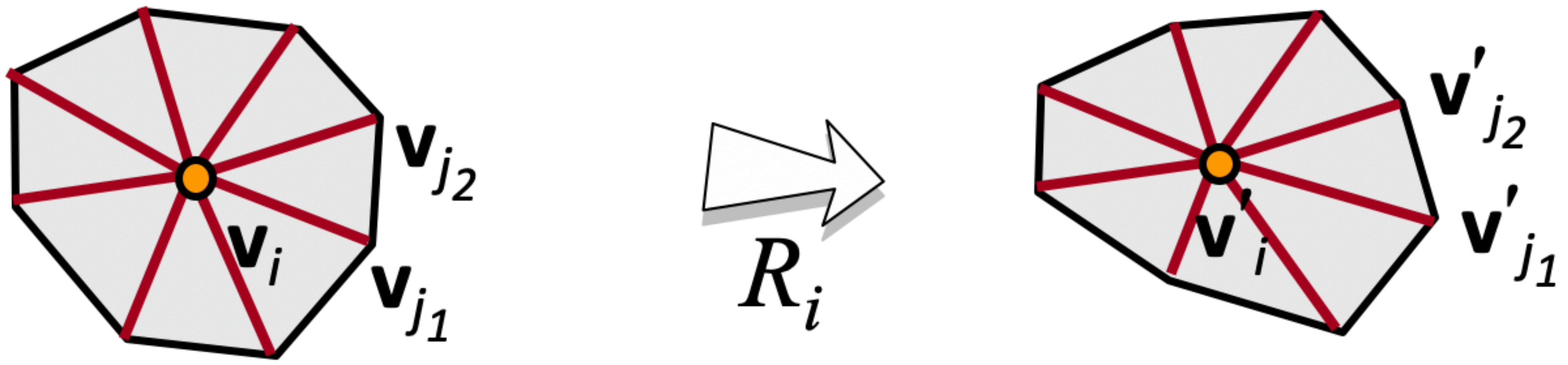
if deformation is not rigid:

$$\min \sum_{j \in N(i)} \left\| (\mathbf{v}'_i - \mathbf{v}'_j) - R_i(\mathbf{v}_i - \mathbf{v}_j) \right\|^2$$

source: https://medium.com/all-math-before-college/transformations-geometry-86643a5cb33b

Energy minimization

If **v**, **v**′ are known then R_i is uniquely defined



$$\min \sum_{j \in N(i)} \left\| (\mathbf{v}'_i - \mathbf{v}'_j) - R_i (\mathbf{v}_i - \mathbf{v}_j) \right\|^2$$

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

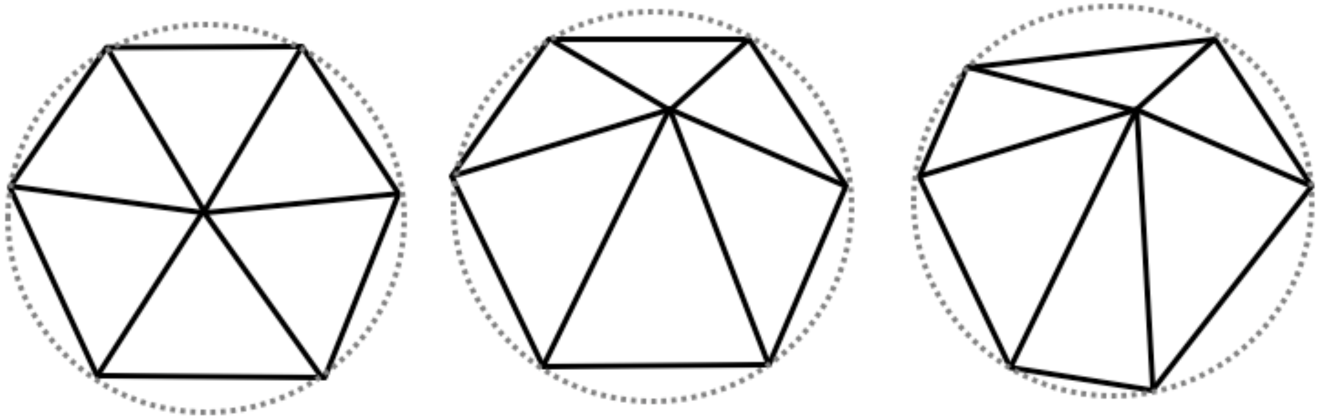
Energy minimization

- **Total deformation energy**

$$\min_{\mathbf{v}'} \sum_{i=1}^{n} \sum_{j \in N(i)} \left\| (\mathbf{v}'_i - \mathbf{v}'_j) - R_i (\mathbf{v}_i - \mathbf{v}_j) \right\|^2$$

- **Alternating iterations of $\boldsymbol{V'}$ and $\boldsymbol{R_i}$**
  - Given initial guess $v'_{0,}$ g find optimal rotations
  - Given the R_i (fixed), minimize the energy by finding new v'

Need appropriate weighting

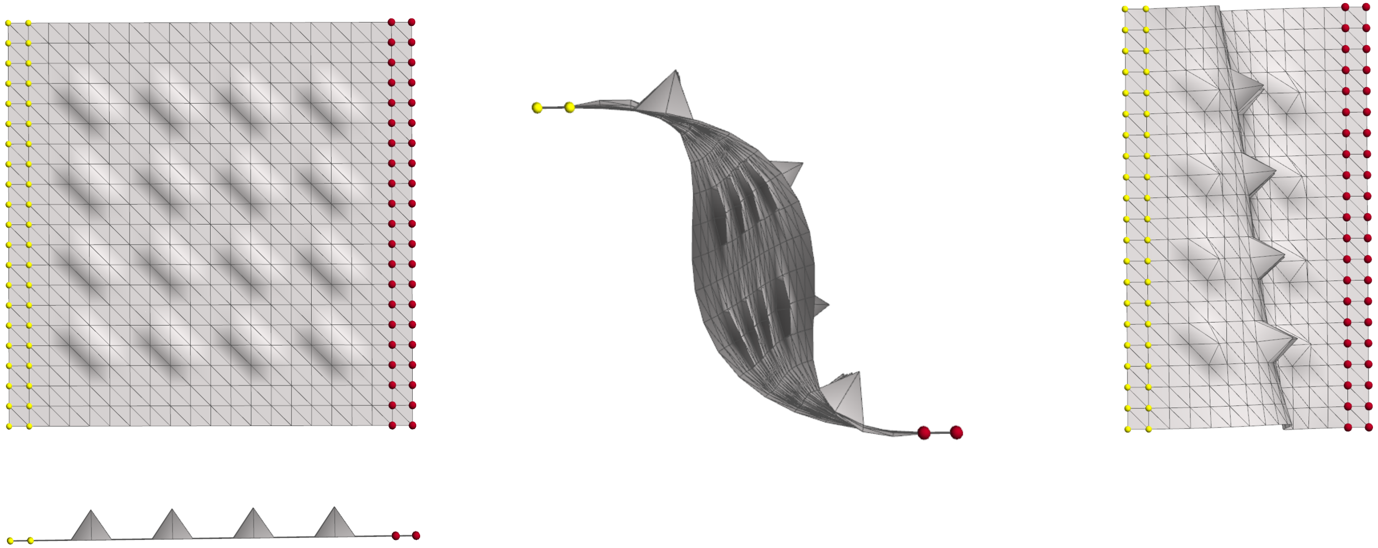


$$E_{cell} = \sum_{j \in N(i)} \left\| (\mathbf{v}'_i - \mathbf{v}'_j) - R_i (\mathbf{v}_i - \mathbf{v}_j) \right\|^2$$

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

Need appropriate weighting

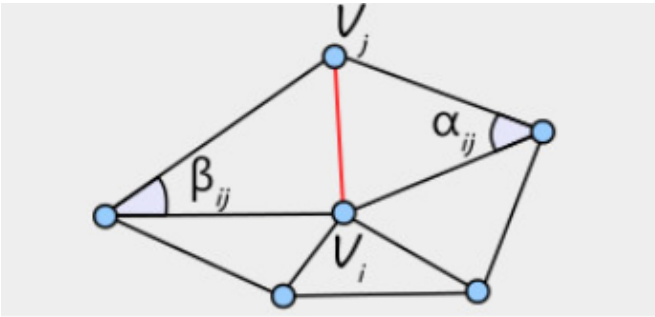
- **Non-symmetric results**



source: Sorkine, Olga, and Marc Alexa. "As-rigid-as-possible surface modeling." 2007.

Need appropriate weighting

$$E_{cell} = \sum_{j \in N(i)} w_{ij} \left\| (\mathbf{v}'_i - \mathbf{v}'_j) - R_i(\mathbf{v}_i - \mathbf{v}_j) \right\|^2$$

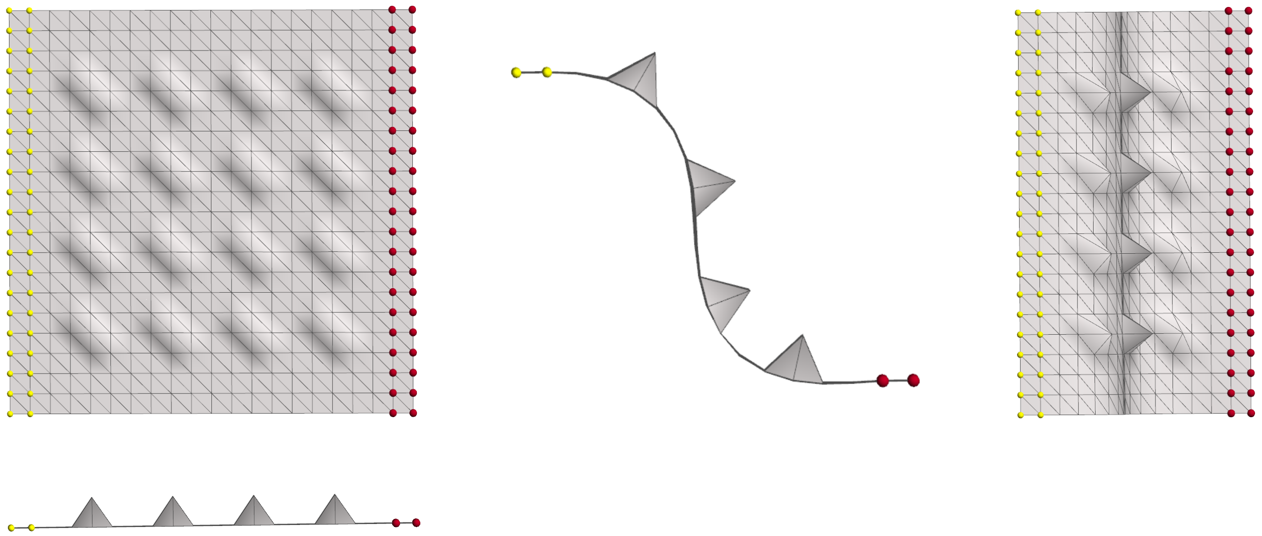


$$w_{ij} = \cot(\alpha_{ij}) + \cot(\beta_{ij})$$

source: http://rodolphe-vaillant.fr/?e=69

Weighted Energy function

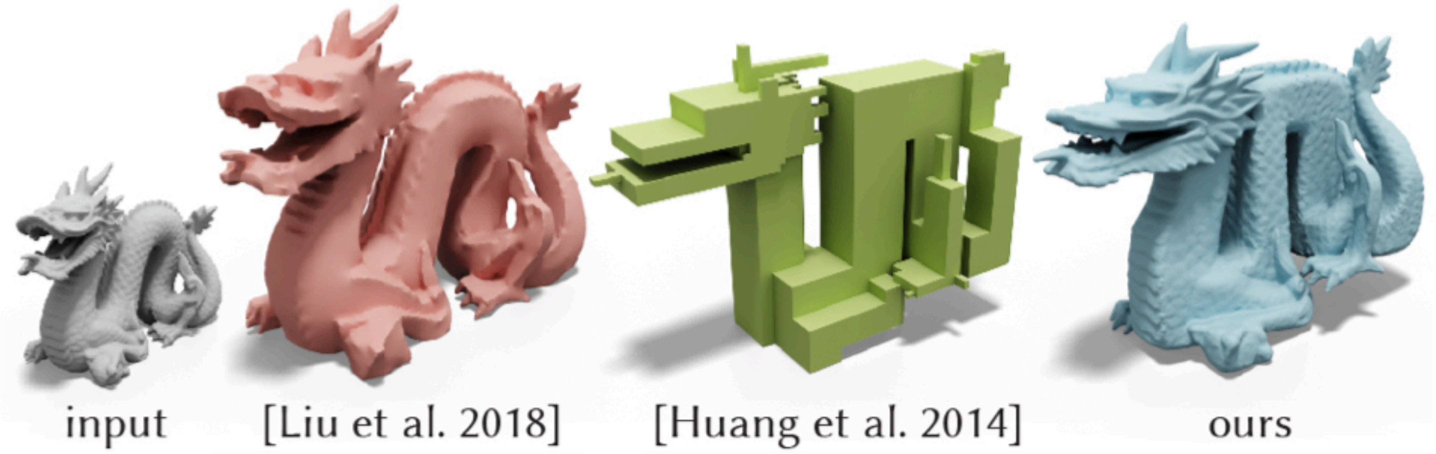
- **Symmetric results**



$$E_{total} = \sum_{i=1}^{n} \sum_{j \in N(i)} w_{ij} \left\| (\mathbf{v}'_i - \mathbf{v}'_j) - R_i(\mathbf{v}_i - \mathbf{v}_j) \right\|^2$$

source: Sorkine, Olga, and Marc Alexa. "As-rigid-as-possible surface modeling." 2007.

ARAP: As-rigid-as-possible

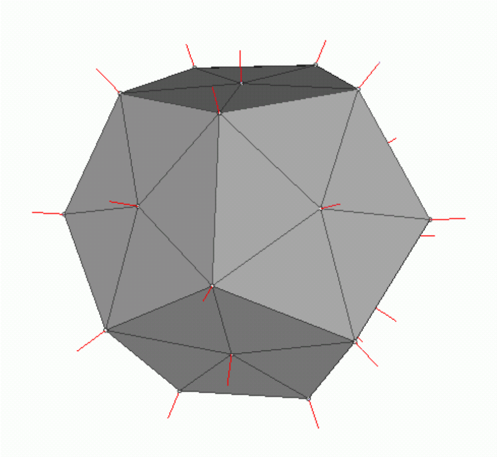


$$\underset{\widetilde{V},\{R_i\}}{\text{minimize}} \sum_{i\in V} \sum_{j\in N(i)} \underbrace{\frac{w_{ij}}{2} \left\| R_i d_{ij} - \widetilde{d}_{ij} \right\|_F^2}_{\text{ARAP}} + \underbrace{\lambda a_i \left\| R_i \hat{n}_i \right\|_1}_{\text{CUBENESS}}$$

CUBENESS:

$$\lambda a_i \, \|\mathrm{R}_i \hat{\mathrm{n}}_i\|_1$$

minimize l1 norm enforce sparsity



[0.5, 0.5, 0.7]

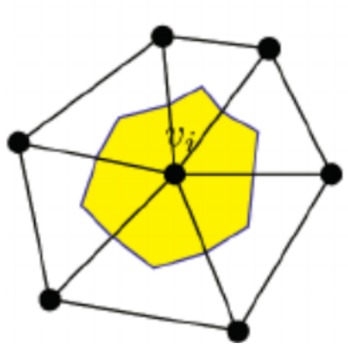
[1, 0, 0]

normal aligned with x axis

source: https://en.wikipedia.org/wiki/Vertex_normal

CUBENESS:

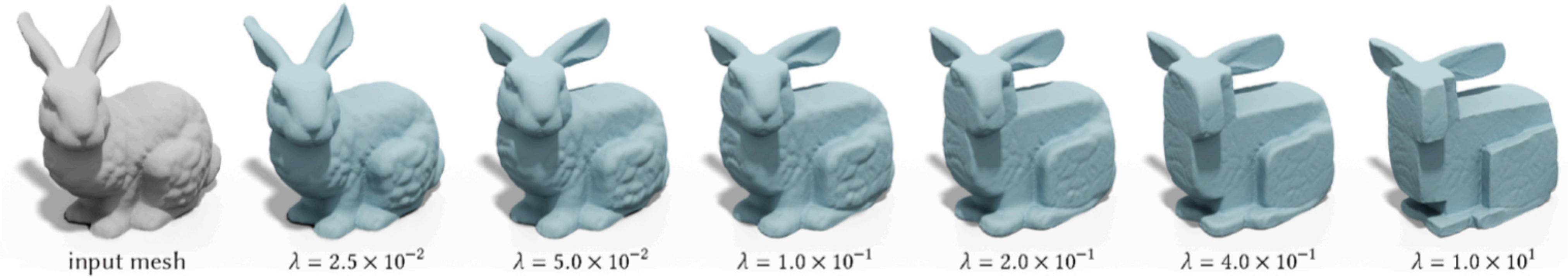
$$\lambda a_i \left\| \mathrm{R}_i \hat{\mathrm{n}}_i \right\|_1$$



source: https://www.researchgate.net/figure/A-mesh-and-its-dual-mesh-a-Barycentric-dual-b-Circumcentric-dual_fig1_5483806 [accessed 2 Oct, 2020]

CUBENESS:

$$\lambda a_i \left\| \mathrm{R}_i \hat{\mathrm{n}}_i \right\|_1$$



How to optimize it?

$$\underset{\widetilde{\mathrm{V}},\{\mathrm{R}_i\}}{\text{minimize}} \sum_{i\in V} \sum_{j\in N(i)} \underbrace{\frac{w_{ij}}{2} \left\|\mathrm{R}_i \mathrm{d}_{ij} - \widetilde{\mathrm{d}}_{ij}\right\|_F^2}_{\text{ARAP}} + \underbrace{\lambda a_i \left\|\mathrm{R}_i \hat{\mathrm{n}}_i\right\|_1}_{\text{CUBENESS}}$$

- **Local-Global**
  - Local: $\mathrm{R}_i^\star = \underset{\mathrm{R}_i\in SO(3)}{\arg\min} \frac{1}{2}\|\mathrm{R}_i \mathrm{D}_i - \widetilde{\mathrm{D}}_i\|_{\mathrm{W}_i}^2 + \lambda a_i \|\mathrm{R}_i \hat{\mathrm{n}}_i\|_1$ → ADMM
  - Global : $\widetilde{\mathrm{V}}^\star = \underset{\widetilde{\mathrm{V}}}{\arg\min} \sum_{i\in V} \frac{1}{2} \left\|\mathrm{R}_i \mathrm{D}_i - \widetilde{\mathrm{D}}_i\right\|_{W_i}^2$ → Linear System

ADMM: Alternating Direction Method of Multipliers

1. Dual ascent

Consider the problem: $\min_x \ f(x) \ \text{ subject to } \ Ax = b$

where f is strictly convex and closed. Denote Lagrangian

$$L(x, u) = f(x) + u^T(Ax - b)$$

decomposable

Dual gradient ascent repeats, for k = 1, 2, 3 ⋯

$$x^{(k)} = \underset{x}{\operatorname{argmin}} \ L(x, u^{(k-1)})$$
$$u^{(k)} = u^{(k-1)} + t_k(Ax^{(k)} - b)$$

Good: decomposable if f is eg. $\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_v} \sum_{i=1}^{v} f_i(\mathbf{x}_i)$

Bad: stringent assumption to ensure convergence

ADMM: Alternating Direction Method of Multipliers

2. Augmented Lagrangian method

Consider the problem: $\min_x f(x)$ subject to $Ax = b$

uses a modified Lagrangian

$$L_\rho(x, u) = f(x) + u^T(Ax - b) + \boxed{\frac{\rho}{2}\|Ax - b\|_2^2}$$

Dual gradient ascent repeats, for k = 1, 2, 3 ···

$$x^{(k)} = \underset{x}{\operatorname{argmin}}\ L_\rho(x, u^{(k-1)})$$
$$u^{(k)} = u^{(k-1)} + \rho(Ax^{(k)} - b)$$

Good: better convergence properties
Bad: loss the decomposability

ADMM: Alternating Direction Method of Multipliers

tries for the best of the both methods:

Consider the problem: $$\min_{x,z} f(x) + g(z) \text{ subject to } Ax + Bz = c$$

uses an augmented Lagrangian

$$L_\rho(x, z, u) = f(x) + g(z) + u^T(Ax + Bz - c) + \boxed{\frac{\rho}{2}\|Ax + Bz - c\|_2^2}$$

Dual gradient ascent repeats, for k = 1, 2, 3 ⋯

$$x^{(k)} = \underset{x}{\operatorname{argmin}}\ L_\rho(x, z^{(k-1)}, u^{(k-1)})$$
$$z^{(k)} = \underset{z}{\operatorname{argmin}}\ L_\rho(x^{(k)}, z, u^{(k-1)})$$
$$u^{(k)} = u^{(k-1)} + \rho(Ax^{(k)} + Bz^{(k)} - c)$$

Local Step: Alternating Direction Method of Multipliers (ADMM)

Local Step: $\mathrm{R}_i^\star = \underset{\mathrm{R}_i \in SO(3)}{\arg\min} \; \frac{1}{2}\|\mathrm{R}_i \mathrm{D}_i - \widetilde{\mathrm{D}}_i\|_{\mathrm{W}_i}^2 + \lambda a_i \|\mathrm{R}_i \hat{\mathrm{n}}_i\|_1$

By setting $z = R_i \tilde{n}_i$, rewrite as

$$\underset{\mathrm{z}, \mathrm{R}_i \in SO(3)}{\text{minimize}} \quad \frac{1}{2}\|\mathrm{R}_i \mathrm{D}_i - \widetilde{\mathrm{D}}_i\|_{\mathrm{W}_i}^2 + \lambda a_i \|\mathrm{z}\|_1$$

$$\text{subject to} \quad \mathrm{z} - \mathrm{R}_i \hat{\mathrm{n}}_i = 0.$$

Local Step: Alternating Direction Method of Multipliers (ADMM)

Local Step: $\mathrm{R}_i^\star = \underset{\mathrm{R}_i \in SO(3)}{\arg\min} \ \frac{1}{2}\|\mathrm{R}_i \mathrm{D}_i - \widetilde{\mathrm{D}}_i\|_{\mathrm{W}_i}^2 + \lambda a_i \|\mathrm{R}_i \hat{\mathrm{n}}_i\|_1$

By setting $z = R_i \tilde{n}_i$, rewrite as

$$\begin{aligned} &\underset{\mathrm{z}, \mathrm{R}_i \in SO(3)}{\text{minimize}} && \frac{1}{2}\|\mathrm{R}_i \mathrm{D}_i - \widetilde{\mathrm{D}}_i\|_{\mathrm{W}_i}^2 + \lambda a_i \|\mathrm{z}\|_1 \\ &\text{subject to} && \mathrm{z} - \mathrm{R}_i \hat{\mathrm{n}}_i = 0. \end{aligned}$$

ADMM problem form ! (f, g convex)

$$\begin{aligned} &\text{minimize} && f(x) + g(z) \\ &\text{subject to} && Ax + Bz = c \end{aligned}$$

Local Step: Alternating Direction Method of Multipliers (ADMM)

$$\underset{z, R_i \in SO(3)}{\text{minimize}} \quad \frac{1}{2}\|R_i D_i - \widetilde{D}_i\|_{W_i}^2 + \lambda a_i \|z\|_1$$
$$\text{subject to} \quad z - R_i \hat{n}_i = 0.$$

ADMM problem form (f, g convex)

$$\text{minimize} \quad f(x) + g(z)$$
$$\text{subject to} \quad Ax + Bz = c$$

Local Step: Alternating Direction Method of Multipliers (ADMM)

$$\underset{z, R_i \in SO(3)}{\text{minimize}} \quad \frac{1}{2}\|R_i D_i - \widetilde{D}_i\|_{W_i}^2 + \lambda a_i \|z\|_1$$
$$\text{subject to} \quad z - R_i \hat{n}_i = 0.$$

ADMM problem form (f, g convex)

$$\text{minimize} \quad f(x) + g(z)$$
$$\text{subject to} \quad Ax + Bz = c$$

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)\|Ax + Bz - c\|_2^2$$

$$\begin{aligned} x^{k+1} &:= \operatorname{argmin}_x L_\rho(x, z^k, y^k) \\ z^{k+1} &:= \operatorname{argmin}_z L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &:= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \end{aligned}$$

Local Step: Alternating Direction Method of Multipliers (ADMM)

$$\underset{z, R_i \in SO(3)}{\text{minimize}} \quad \frac{1}{2}\|R_i D_i - \widetilde{D}_i\|_{W_i}^2 + \lambda a_i \|z\|_1$$

$$\text{subject to} \quad z - R_i \hat{n}_i = 0.$$

ADMM problem form (f, g convex)

$$\text{minimize} \quad f(x) + g(z)$$

$$\text{subject to} \quad Ax + Bz = c$$

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)\|Ax + Bz - c\|_2^2$$

$$x^{k+1} := \operatorname{argmin}_x L_\rho(x, z^k, y^k)$$

$$z^{k+1} := \operatorname{argmin}_z L_\rho(x^{k+1}, z, y^k)$$

$$y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$$

$$R_i^{k+1} \leftarrow \underset{R_i \in SO(3)}{\arg\min} \frac{1}{2}\|R_i D_i - \widetilde{D}_i\|_{W_i}^2 + \frac{\rho^k}{2}\|R_i \hat{n}_i - z^k + u^k\|_2^2$$

$$z^{k+1} \leftarrow \underset{z}{\arg\min}\ \lambda a_i \|z\|_1 + \frac{\rho^k}{2}\|R_i^{k+1}\hat{n}_i - z + u^k\|_2^2$$

$$\tilde{u}^{k+1} \leftarrow u^k + R_i^{k+1}\hat{n}_i - z^{k+1}$$

$$u = (1/p)y$$

Local Step: Alternating Direction Method of Multipliers (ADMM)

$$\mathrm{R}_i^{k+1} \leftarrow \underset{\mathrm{R}_i \in \mathrm{SO}(3)}{\arg\min} \frac{1}{2} \|\mathrm{R}_i \mathrm{D}_i - \widetilde{\mathrm{D}}_i\|_{\mathrm{W}_i}^2 + \frac{\rho^k}{2} \|\mathrm{R}_i \hat{\mathrm{n}}_i - \mathrm{z}^k + \mathrm{u}^k\|_2^2$$

→ Lasso problem

Local Step: Alternating Direction Method of Multipliers (ADMM)

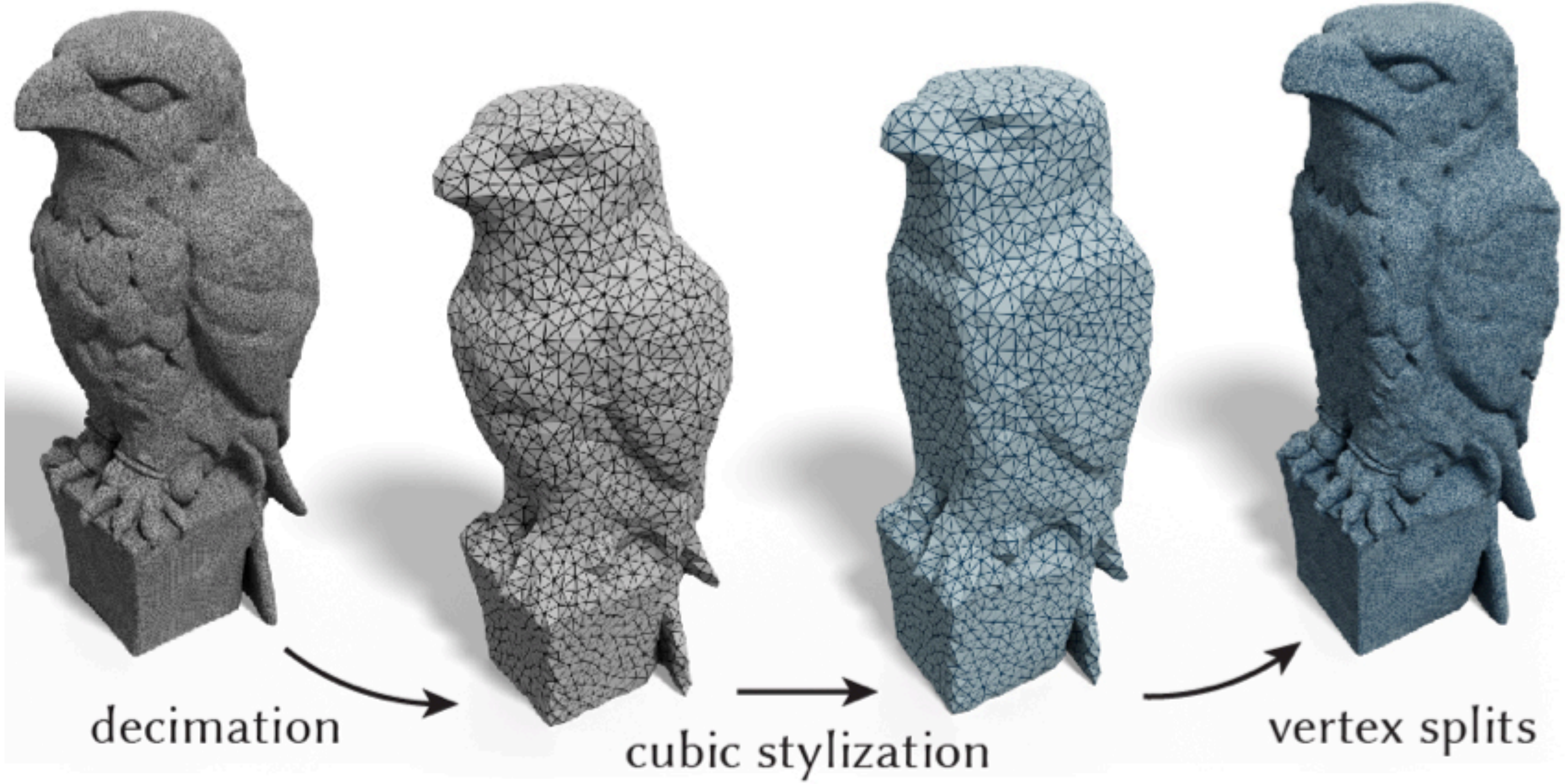
$$R_i^{k+1} \leftarrow \arg\min_{R_i \in SO(3)} \frac{1}{2}\|R_i D_i - \widetilde{D}_i\|_{W_i}^2 + \frac{\rho^k}{2}\|R_i \hat{n}_i - z^k + u^k\|_2^2$$

→ Lasso problem

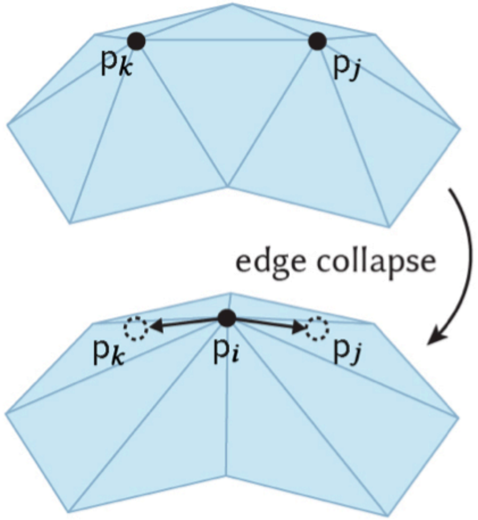
$$z^{k+1} \leftarrow \arg\min_{z} \lambda a_i \|z\|_1 + \frac{\rho^k}{2}\|R_i^{k+1}\hat{n}_i - z + u^k\|_2^2$$

→ Orthogonal Procrustes problem

Affine Progressive Meshes



Affine Progressive Meshes



Store: 1. displacement vectors from p_i to p_j p_k

2. matrix A : $A = (Q_i Q_i^T)^{-1} Q_i$.

Deformation

Recover: $\widetilde{p}_j - \widetilde{p}_i = \widetilde{Q}_i A^T (p_j - p_i)$

Q_i is a $3 \times |N(i)|$ matrix where each column is the vector from pi to one of its one-rings neighbours $N(i)$.

Algorithm

Algorithm 2: *Fast Cube Stylization* (λ, m)

Input : A triangle mesh V, F
Output: Deformed vertex positions $\widetilde{V}$

// pre-processing

1. $m \leftarrow$ target number of faces
2. $V_c, F_c \leftarrow$ *edge collapses* (V, F, m)

// cubic stylization

3. $\widetilde{V}_c \leftarrow V_c$
4. **while** *not converge* **do**
5. $\quad R \leftarrow$ *local-step* $(V_c, \widetilde{V}_c, \lambda)$
6. $\quad \widetilde{V}_c \leftarrow$ *global-step* (R)
7. $\widetilde{V}, F \leftarrow$ *affine vertex splits* $(\widetilde{V}_c, F_c)$

Artistic Controls

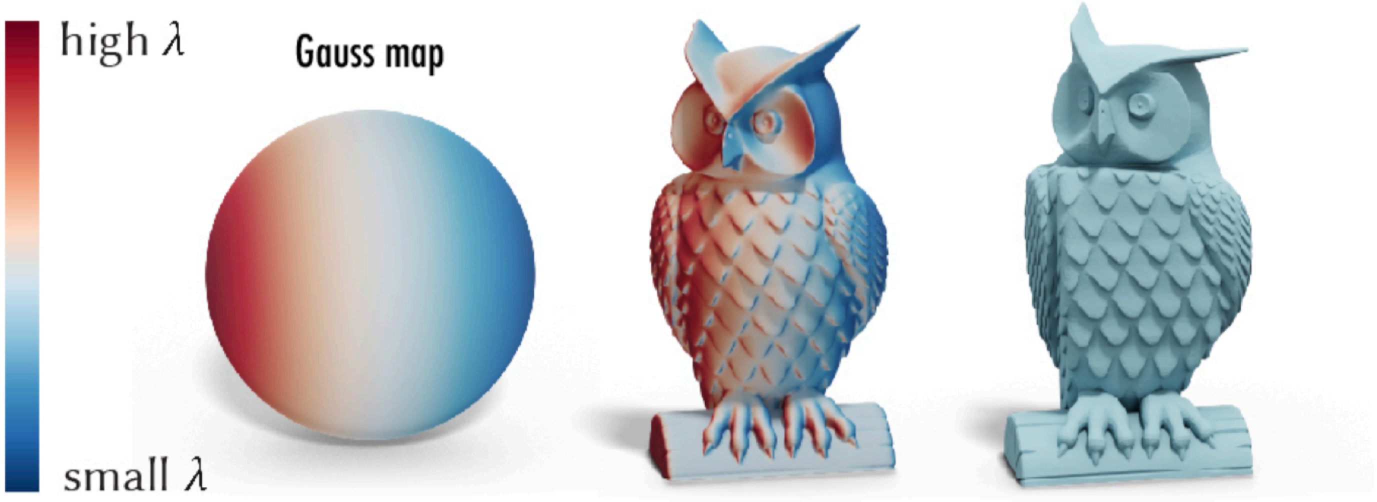
- **Non-uniform cubeness**

Cubeness: $\lambda a_i \left\|\mathrm{R}_i \hat{\mathrm{n}}_i\right\|_1$

Artistic Controls

- **Non-uniform cubeness**

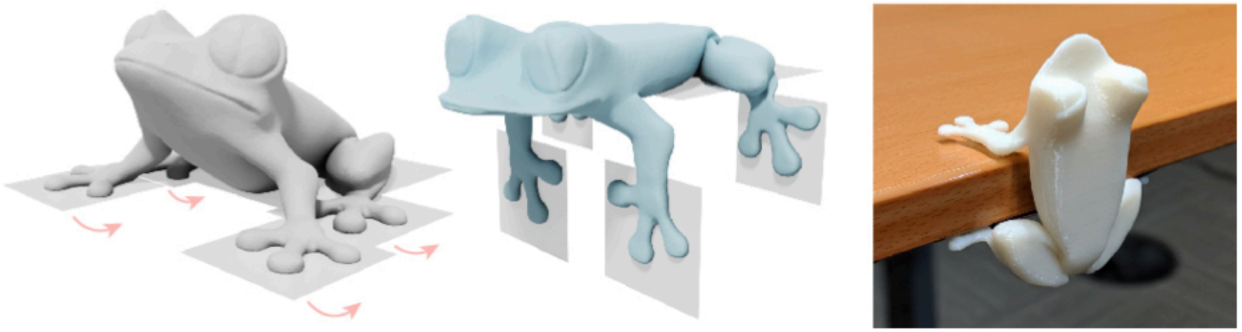
Cubeness: $\lambda a_i \left\| \mathrm{R}_i \hat{\mathrm{n}}_i \right\|_1$



Artistic Controls

- **Fix certain parts of the shape:** add constraint in global step

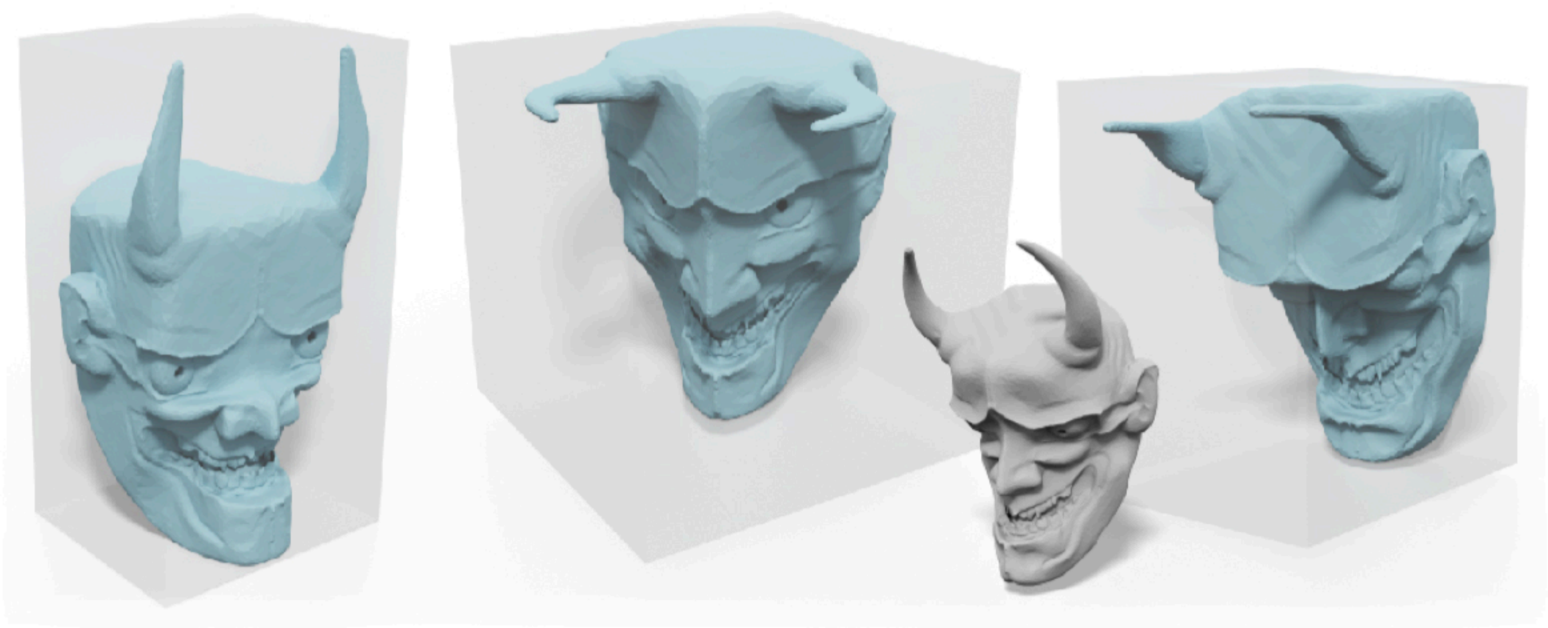
$$\widetilde{\mathrm{V}}^{\star} = \underset{\widetilde{\mathrm{V}}}{\arg\min} \sum_{i \in V} \frac{1}{2} \left\| \mathrm{R}_i \mathrm{D}_i - \widetilde{\mathrm{D}}_i \right\|_{W_i}^2$$



Artistic Controls

- **Apply different Rotations before stylization**

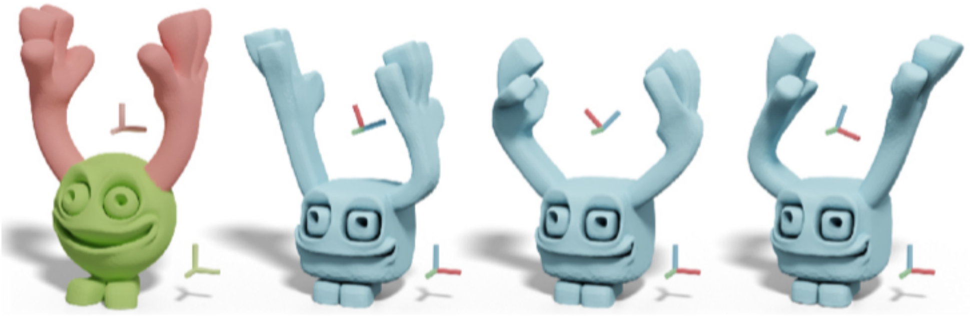
Cubeness: $\lambda a_i \left\| \mathrm{R}_i \hat{\mathrm{n}}_i \right\|_1$



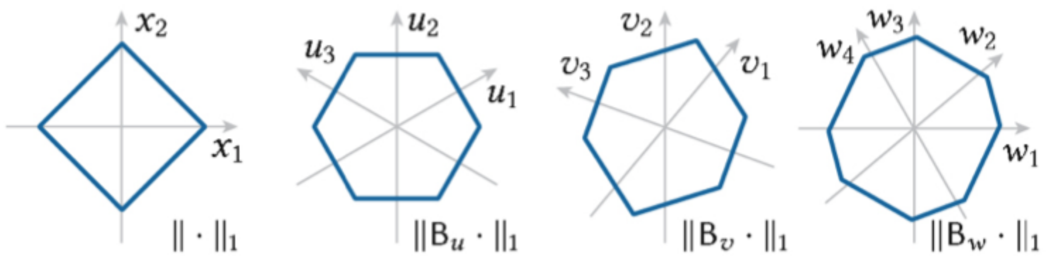
Artistic Controls

- **Apply different coordinate system**

Cubeness: $\lambda a_i \|R_i \hat{n}_i^{\text{local}}\|_1$



Artistic Controls



- **Polyhedral stylization**

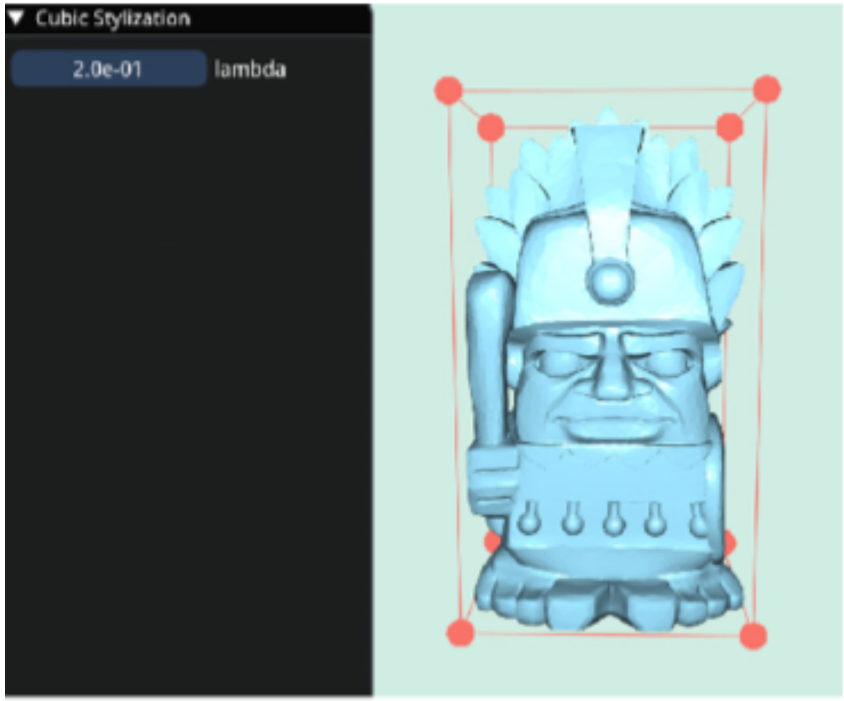
Cubeness: $\lambda a_i \|B R_i \hat{n}_i\|_1$

Runtime

1 2 3 4 5 6

Model	$\|F\|$	λ	m	*Iters.*	*Pre.*	*Runtime*
1	39K	0.20	n/a	106	n/a	**5.08s**
2	41K	0.20	n/a	93	n/a	**4.50s**
3	21K	0.4	n/a	86	n/a	**2.26s**
4	2018K	0.20	20K	83	64.19*s*	**3.93s**
5	346K	0.40	20K	222	10.69*s*	**4.59s**
6	811K	0.30	40K	173	30.44*s*	**8.38s**

User study


Cubic Stylization
2.0e-01
lambda


input scene
cubified scene

Conclusion

- 3D stylization algorithm that can turn an input shape into the style of a cube while maintaining the content of the original shape.
- No mesh surgery
- Artistic Control
- Generalize to polyhedral stylization

Thank you !