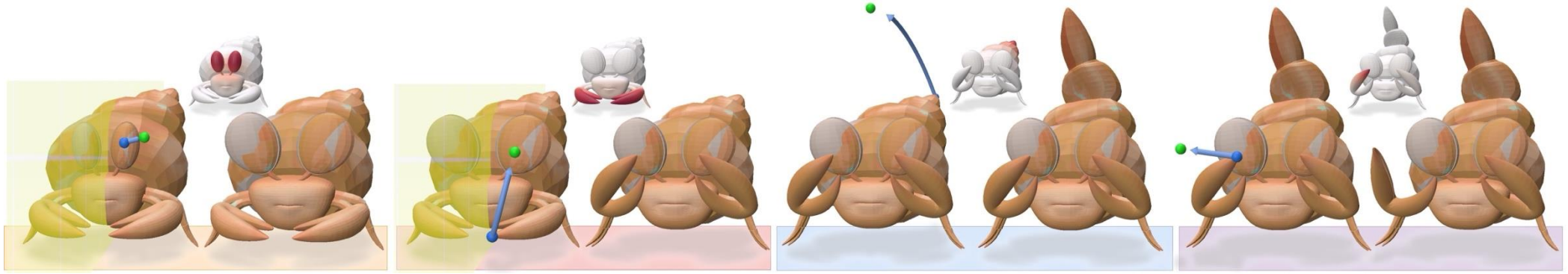
CVPR Highlight

Deep Feature Deformation Weights

Richard Liu, Itai Lang, Rana Hanocka



Background

Traditional handle-based deformation: linear blending with k handles

$$V_i' = \sum_{k=1}^{K} \mathcal{W}_{ik} D_k V_i$$

Background

Traditional handle-based deformation: linear blending with k handles

Fast & fine-grained deformation control

$$V_i' = \sum_{k=1}^{K} \mathcal{W}_{ik} D_k V_i$$

Background

Traditional handle-based deformation: linear blending with k handles

Fast & fine-grained deformation control

$$V_i' = \sum_{k=1}^{K} \mathcal{W}_{ik} D_k V_i$$

Deformed/starting position for vertex i

Background

Traditional handle-based deformation: linear blending with k handles

Fast & fine-grained deformation control

$$V_i' = \sum_{k=1}^{K} \mathcal{W}_{ik} D_k V_i$$

Affine transformation for handle k

Background

Traditional handle-based deformation: linear blending with k handles

Fast & fine-grained deformation control

$$V_i' = \sum_{k=1}^{K} \mathcal{W}_{ik} D_k V_i$$

Blending weight between handle k and vertex i

Background

Traditional handle-based deformation: linear blending with k handles

Fast & fine-grained deformation control

$$V_i' = \sum_{k=1}^{K} \mathcal{W}_{ik} D_k V_i$$

Blending weight between handle k and vertex i

Classical methods solve an optimization problem to obtain W

Background

Traditional handle-based deformation: linear blending with k handles

Fast & fine-grained deformation control

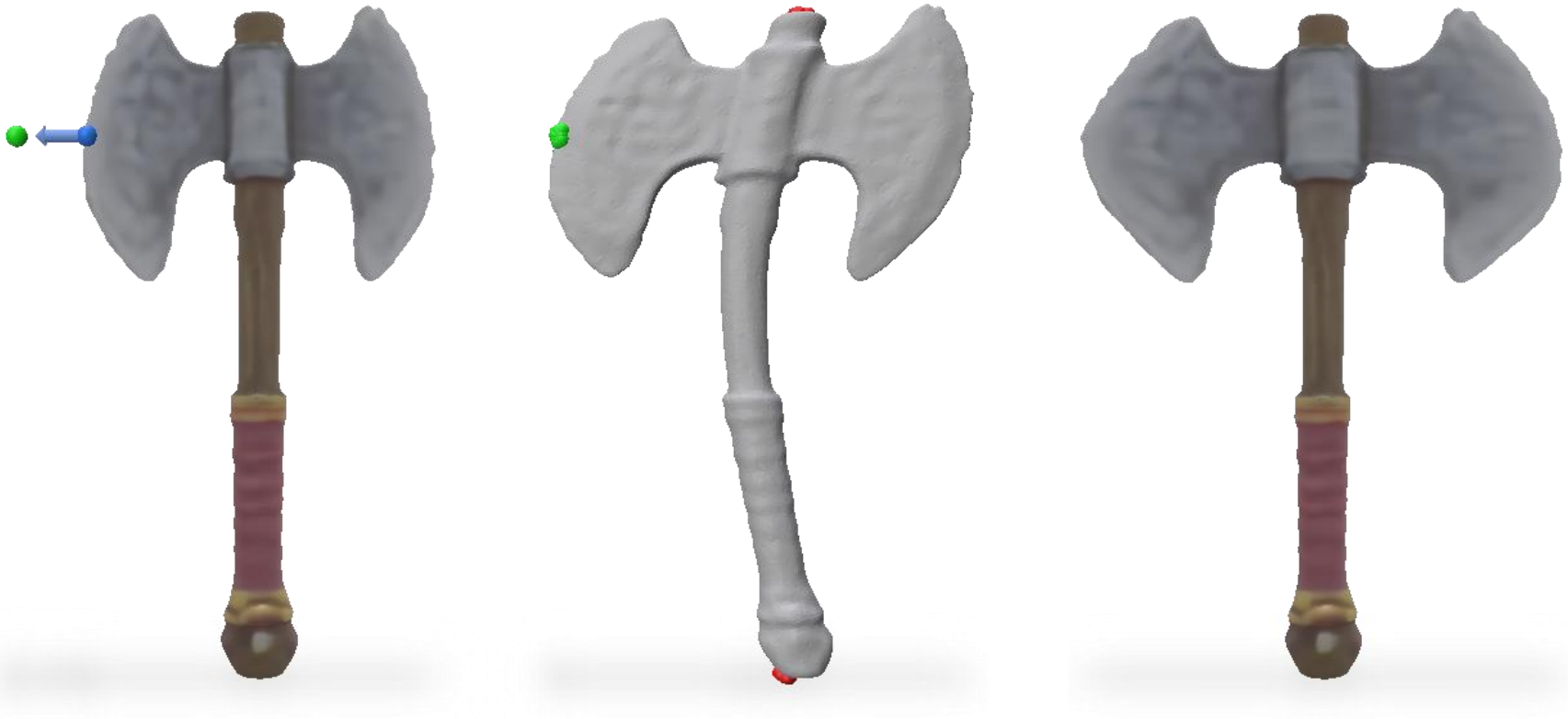
$$V_i' = \sum_{k=1}^{K} \mathcal{W}_{ik} D_k V_i$$

Blending weight between handle k and vertex i

Classical methods solve an optimization problem to obtain W

- Handle set updates require re-solving the optimization
- Optimization aims to minimize surface stretching and global influence

Background



Classical

Ours

Background

Data-driven methods: surface-distorting edits which preserve shape semantics and part relationships

DeepMetaHandles[1]

APAP[2]

[1]Minghua Liu, Minhyuk Sung, Radomir Mech, and Hao Su. Deepmetahandles: Learning deformation meta-handles of 3d meshes with biharmonic coordinates. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12–21, 2021

[2]Seungwoo Yoo, Kunho Kim, Vladimir G Kim, and Minhyuk Sung. As-plausible-as-possible: Plausibility-aware mesh deformation using 2d diffusion priors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4315–4324, 2024

Background

Data-driven methods: surface-distorting edits which preserve shape semantics and part relationships

DeepMetaHandles[1]

APAP[2]

Limitations: expensive pre-training/optimization; no fine-grained deformation control

[1]Minghua Liu, Minhyuk Sung, Radomir Mech, and Hao Su. Deepmetahandles: Learning deformation meta-handles of 3d meshes with biharmonic coordinates. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12–21, 2021

[2]Seungwoo Yoo, Kunho Kim, Vladimir G Kim, and Minhyuk Sung. As-plausible-as-possible: Plausibility-aware mesh deformation using 2d diffusion priors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4315–4324, 2024

Goal

Speed & control

$$V_i' = \sum_{k=1}^{K} \mathcal{W}_{ik} D_k V_i$$

Shape-aware priors

Goal

Speed & control

$$V_i' = \sum_{k=1}^{K} \mathcal{W}_{ik} D_k V_i$$

Shape-aware priors

Our approach: use foundation model features to define blending weights

Method

Given an input mesh, we render and distill image features into a neural field Φ, with **barycentric feature distillation**

Input mesh

Barycentric feature distillation

Neural field (Φ)

Method

Given an input mesh, we render and distill image features into a neural field Φ, with **barycentric feature distillation**

Input mesh

Barycentric feature distillation

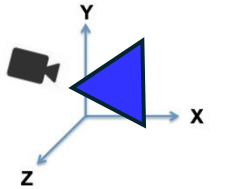
Neural field (Φ)

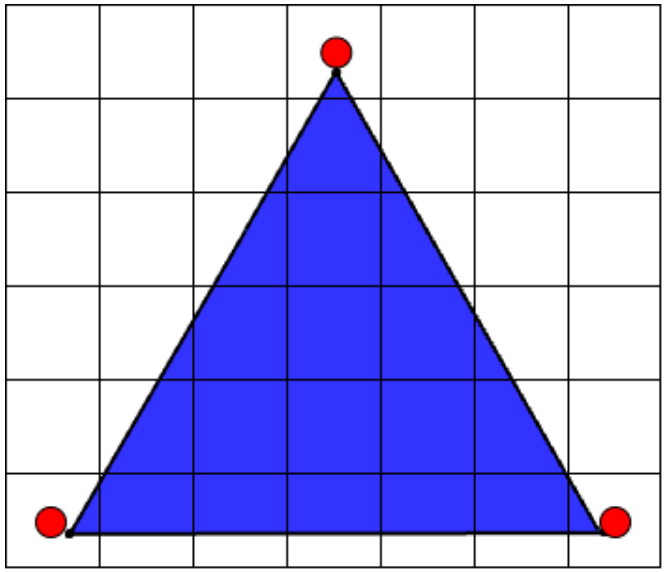
Deep Feature Deformation Weights

$$\mathcal{W}_{ik} = 1 - ||\Phi(V_i) - \Phi(V_k)||$$

Barycentric Feature Distillation

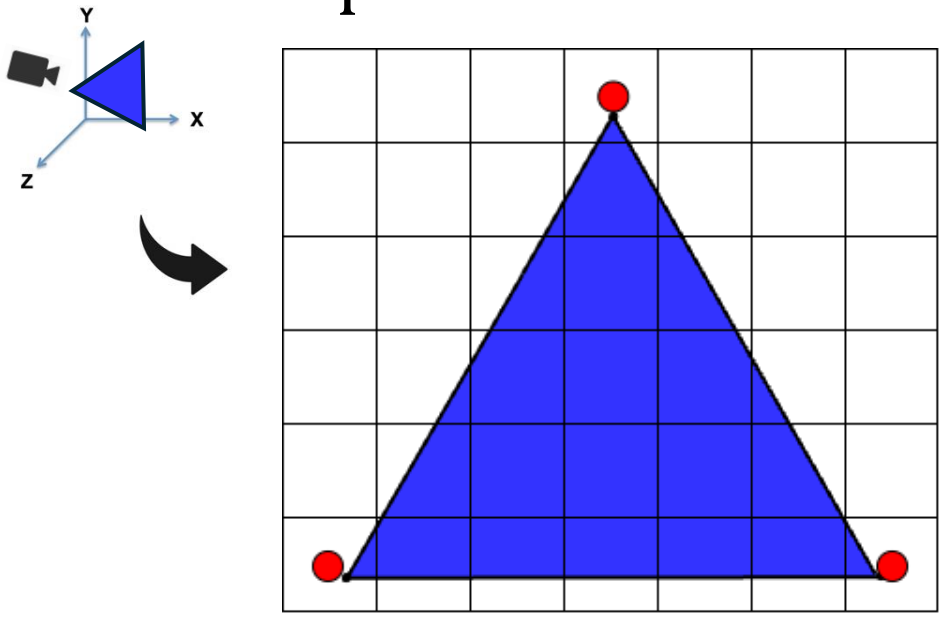
Past distillation methods:
map features to vertices



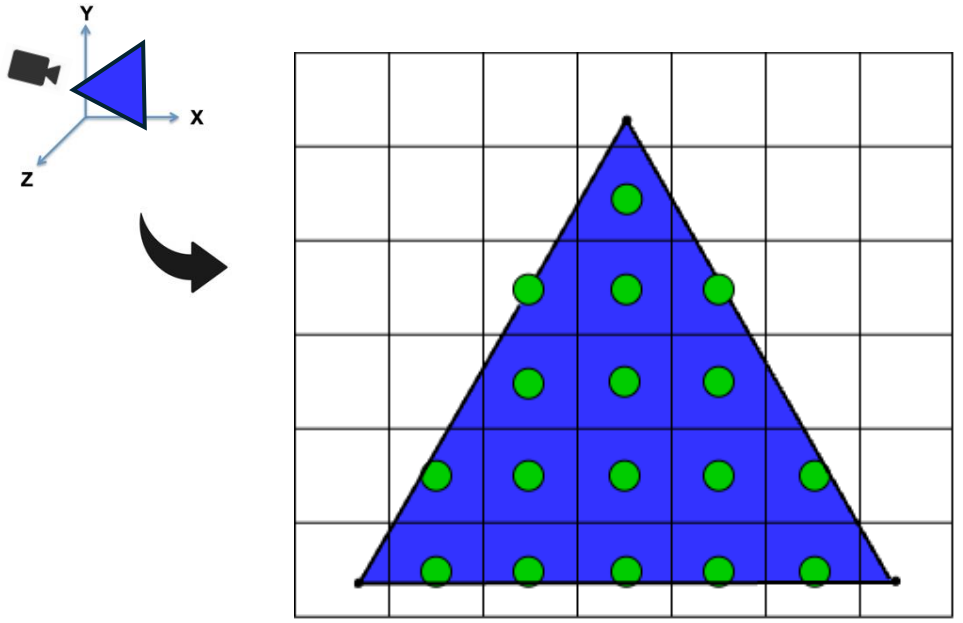


Barycentric Feature Distillation

Past distillation methods:
map features to vertices

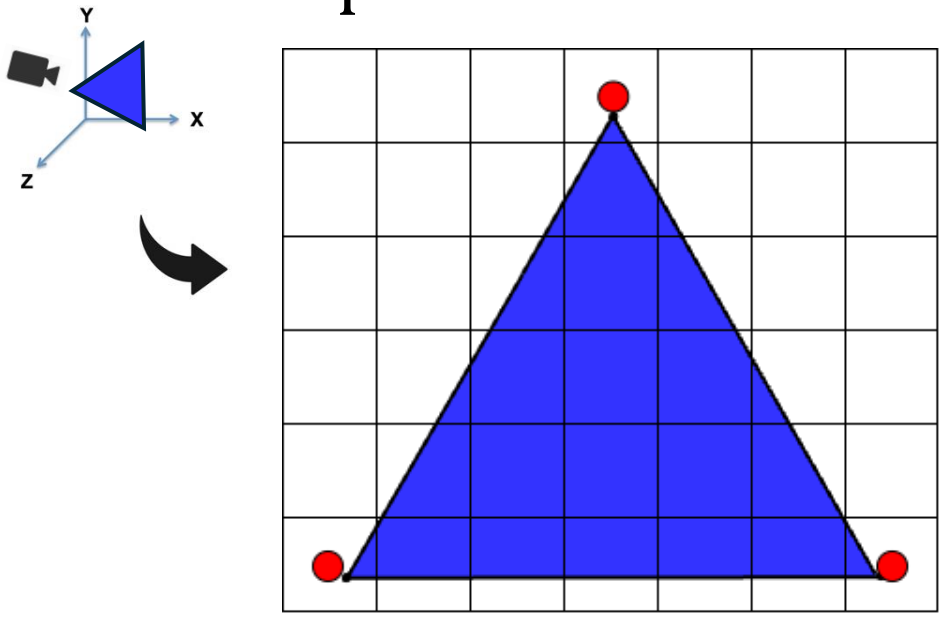


Ours: map features to 3D points

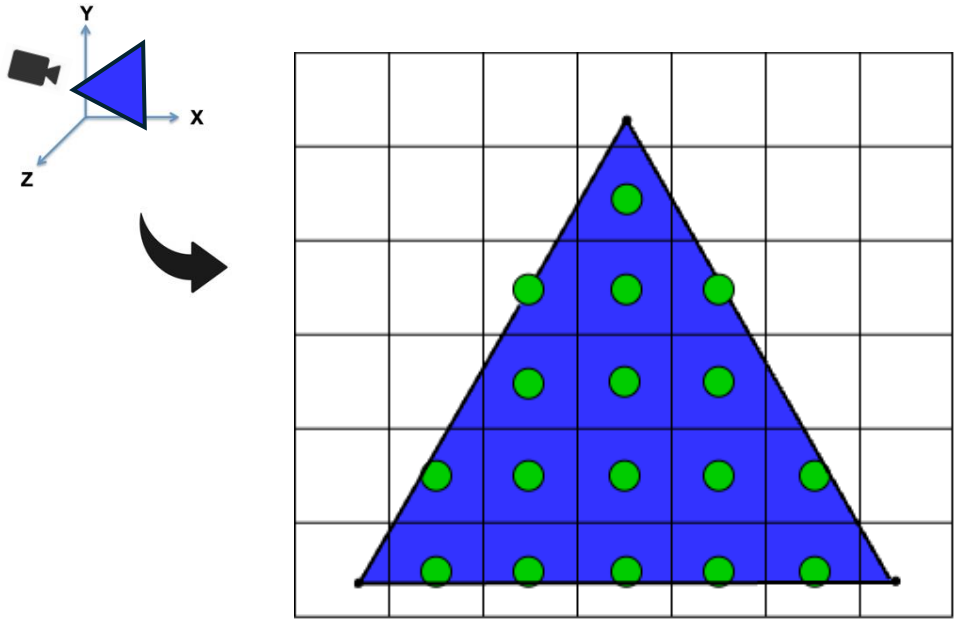


Barycentric Feature Distillation

Past distillation methods:
map features to vertices



Ours: map features to 3D points



We disentangle distillation complexity from mesh complexity

Barycentric Feature Distillation

I/O: 22s
Simplification: 30s
Distillation: 41s

Vertices: 14m
Faces: 28m

~1½ minutes

Barycentric Feature Distillation

I/O: 22s
Simplification: 30s
Distillation: 41s

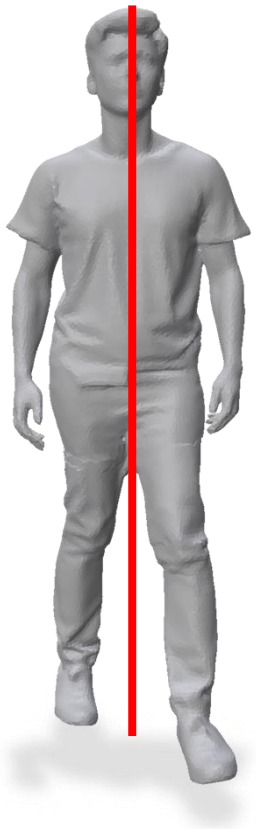
Vertices: 14m
Faces: 28m

~1½ minutes

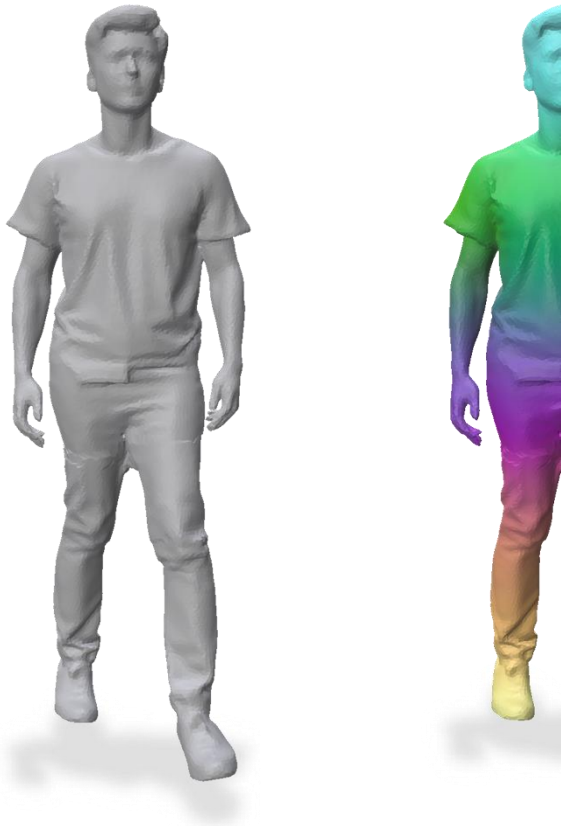
Previous methods take hours/days or hit memory limits

Visual Symmetry Detection

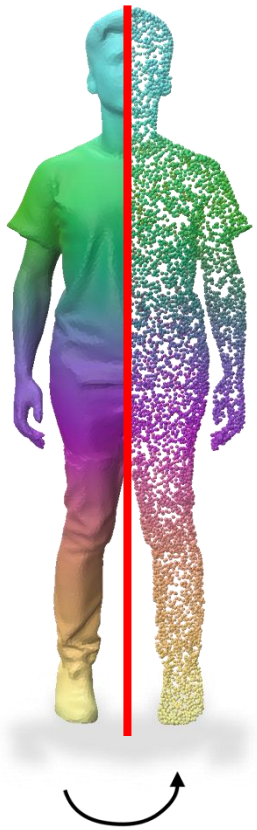
Visual Symmetry Detection



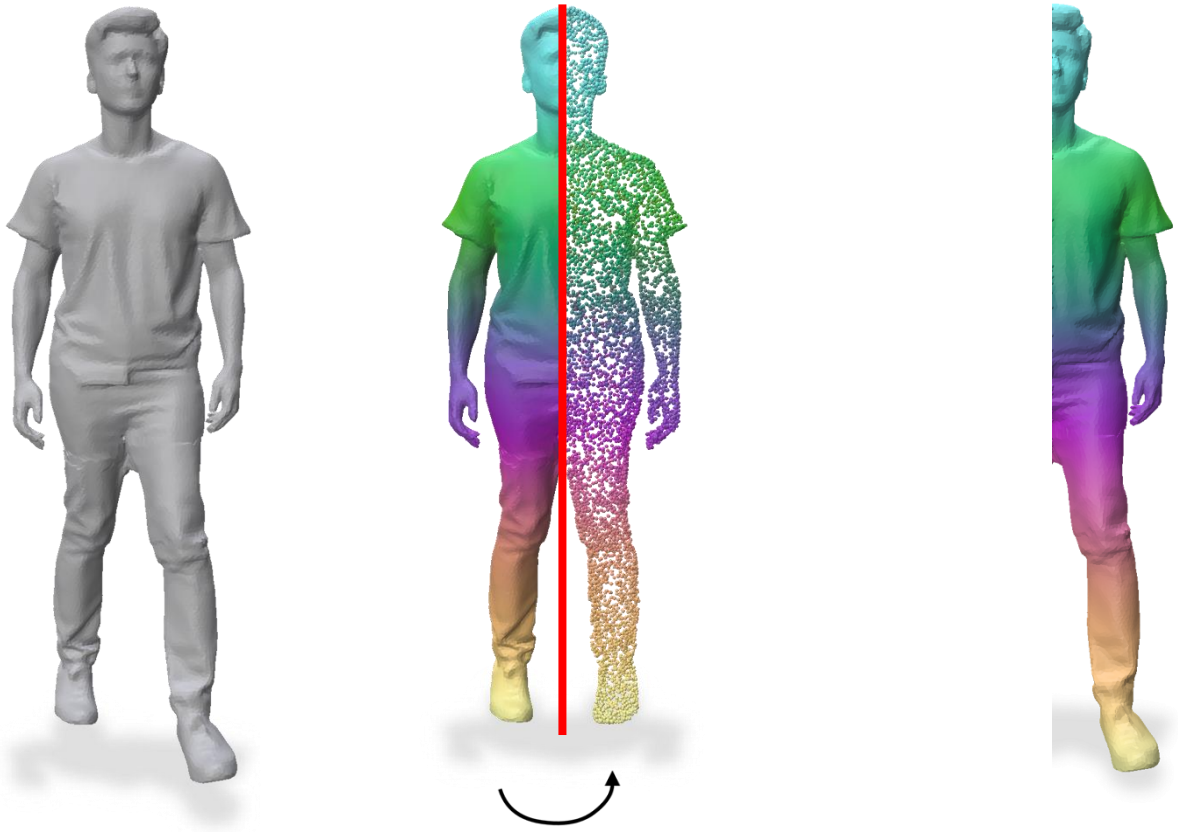
Visual Symmetry Detection



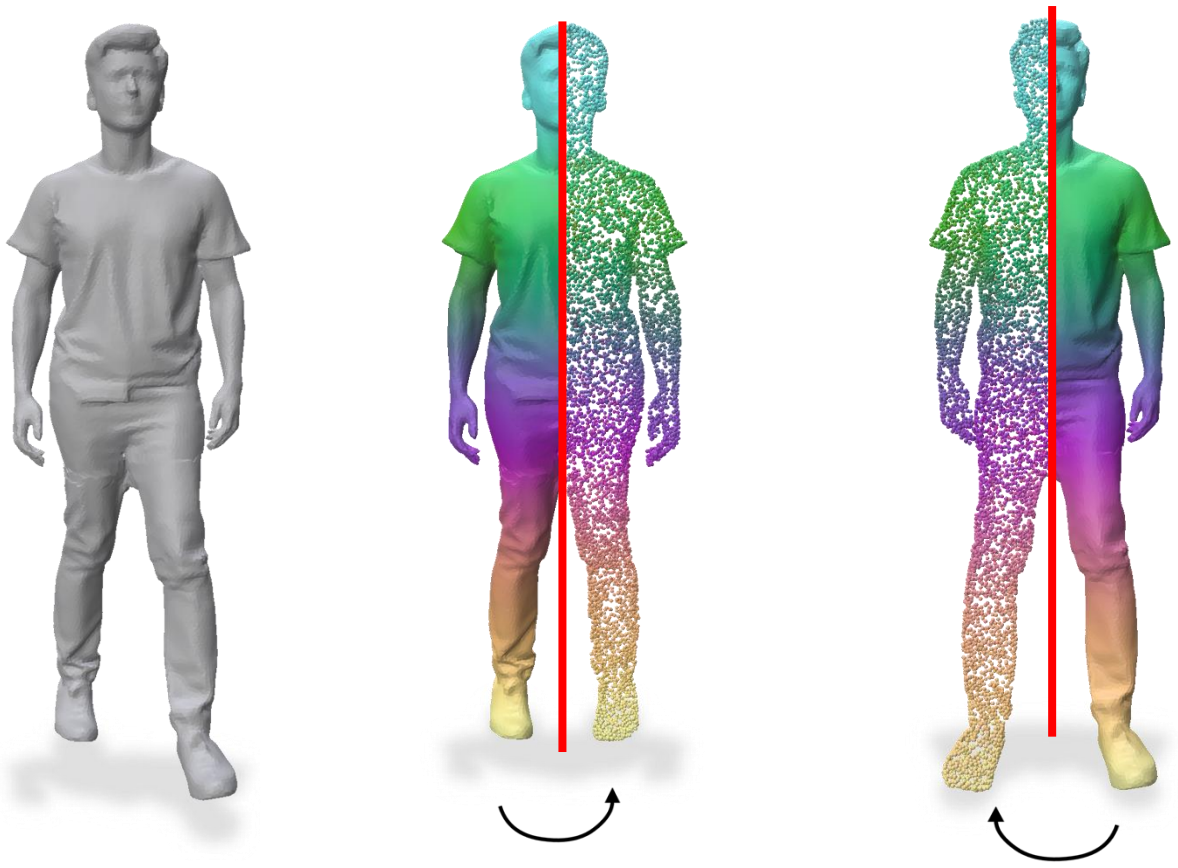
Visual Symmetry Detection



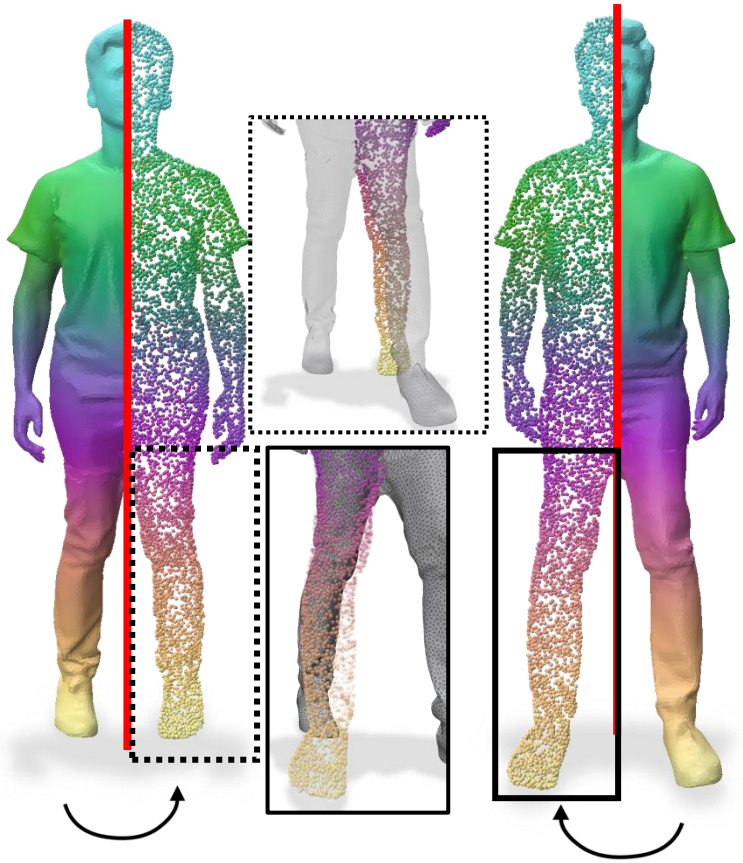
Visual Symmetry Detection



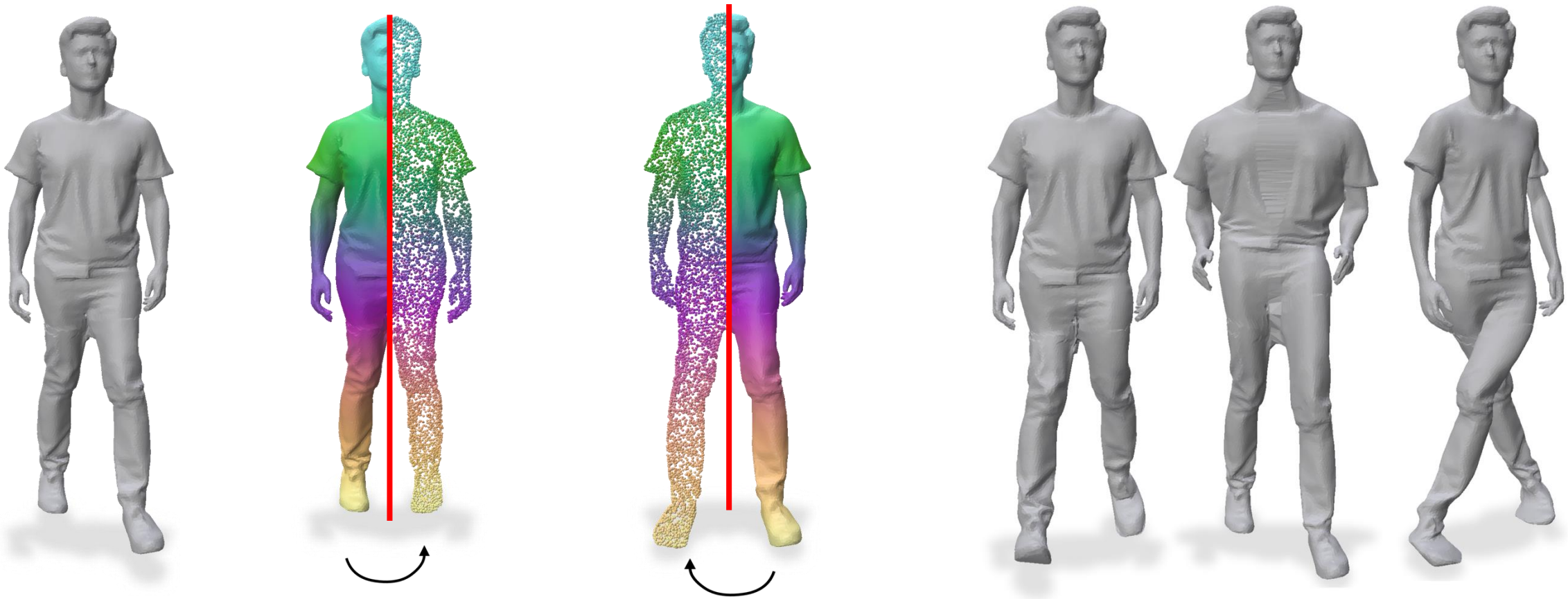
Visual Symmetry Detection



Visual Symmetry Detection

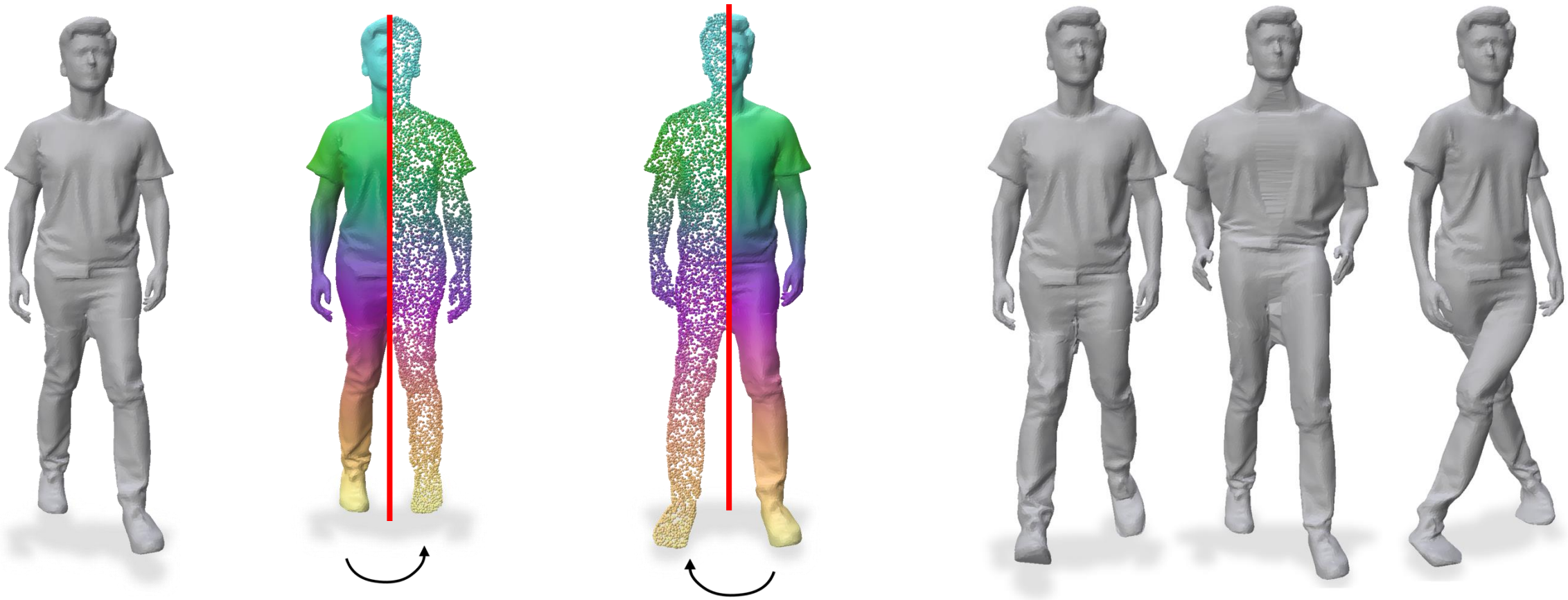


Visual Symmetry Detection



Symmetric Deformations

Visual Symmetry Detection



Symmetric Deformations

Related to Intrinsic Symmetries (Ovsjanikov et al. 08)

Deformation Results







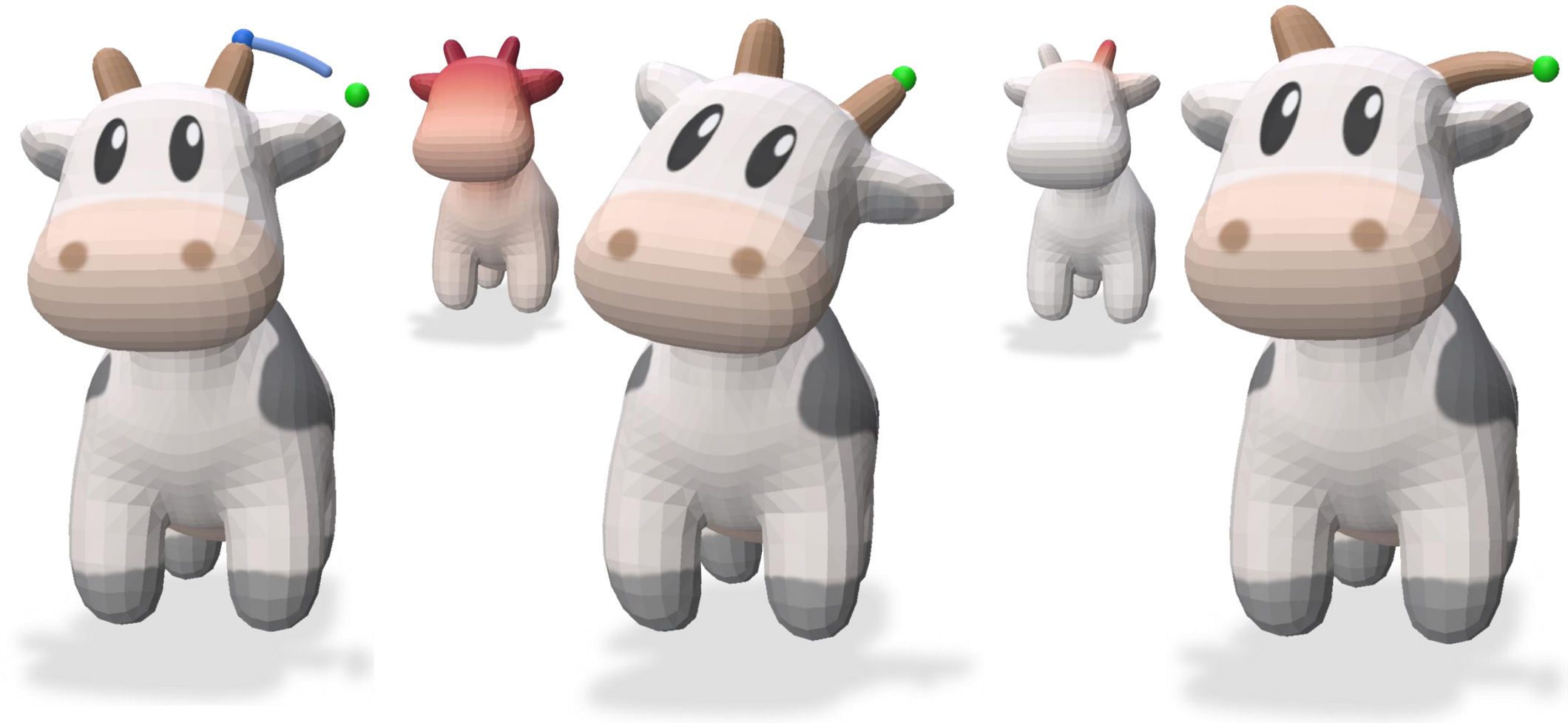





User-defined locality weights

Default Weights

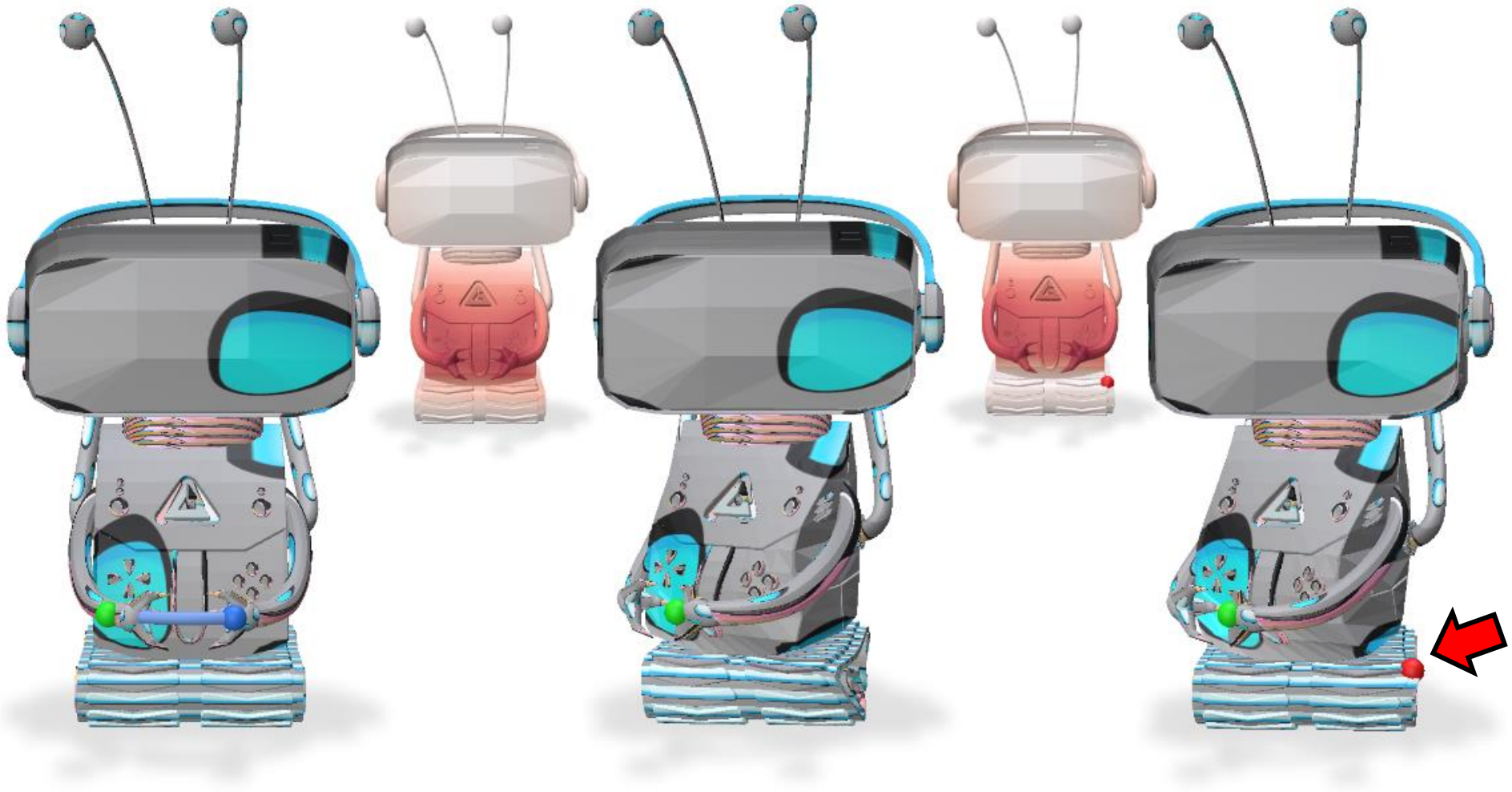
Local Weights



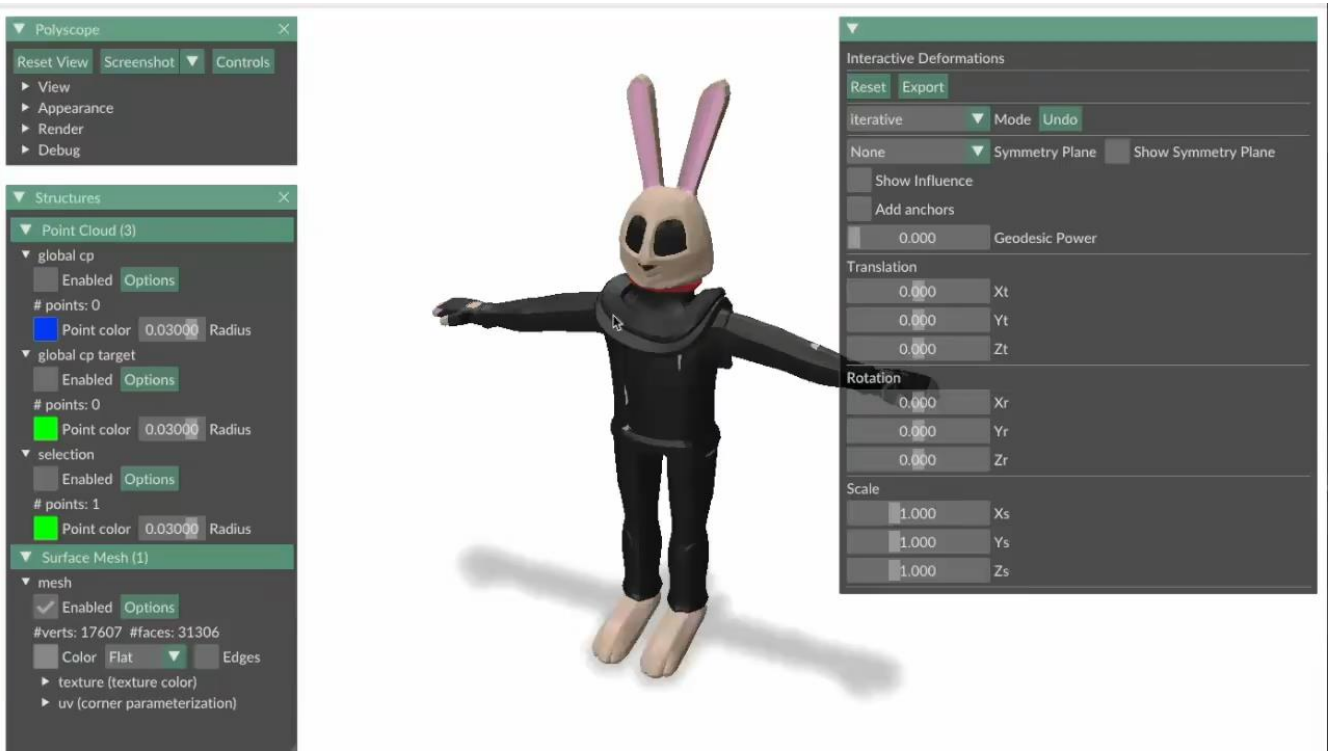
Fixed points act as feature space constraints

Default Weights

With Fixed Point



Code & interactive GUI available online



We’re robust to topological defects!

Project Page

Paper

Code