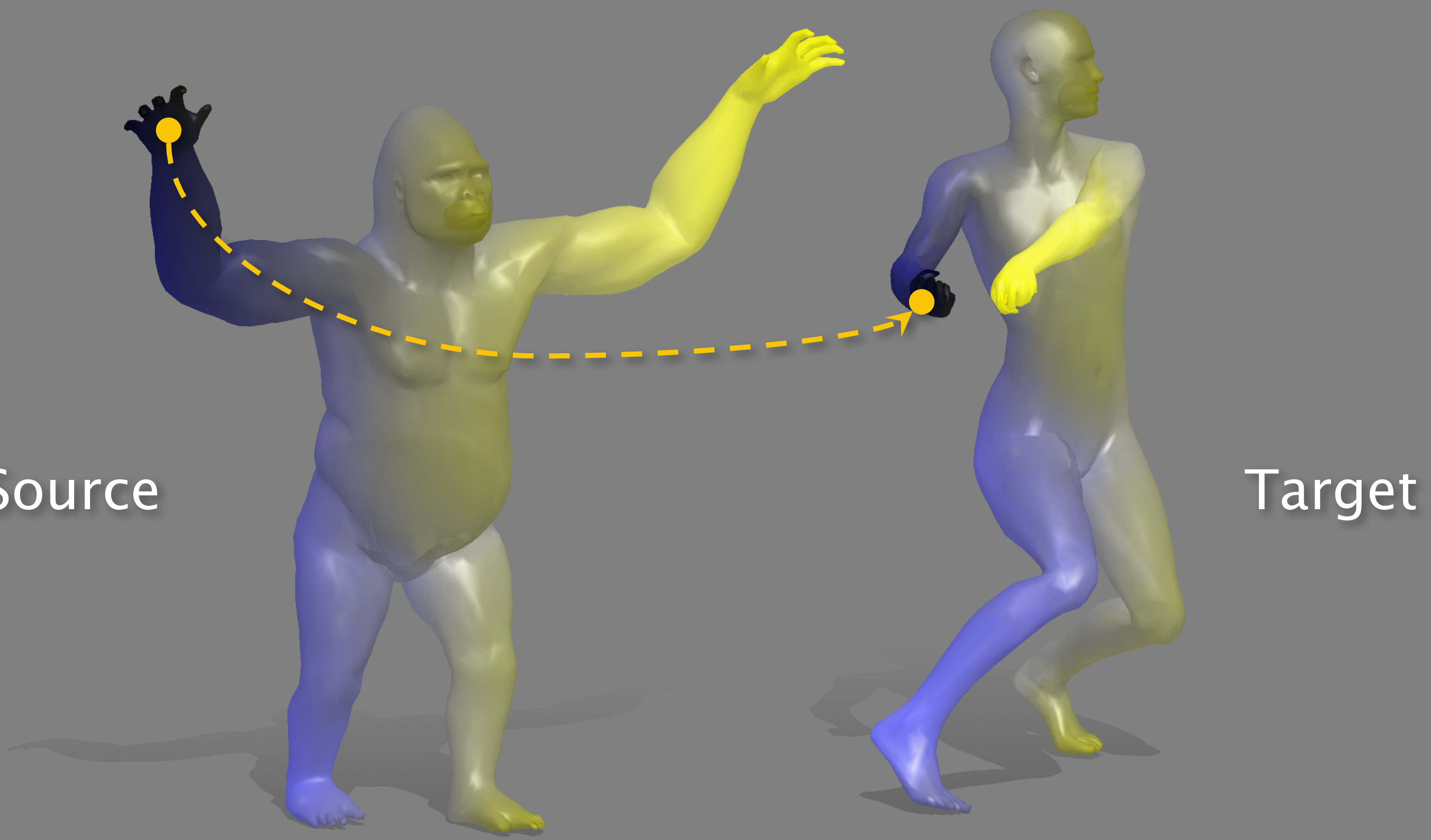
Continuous and Orientation-preserving Correspondences via Functional Maps

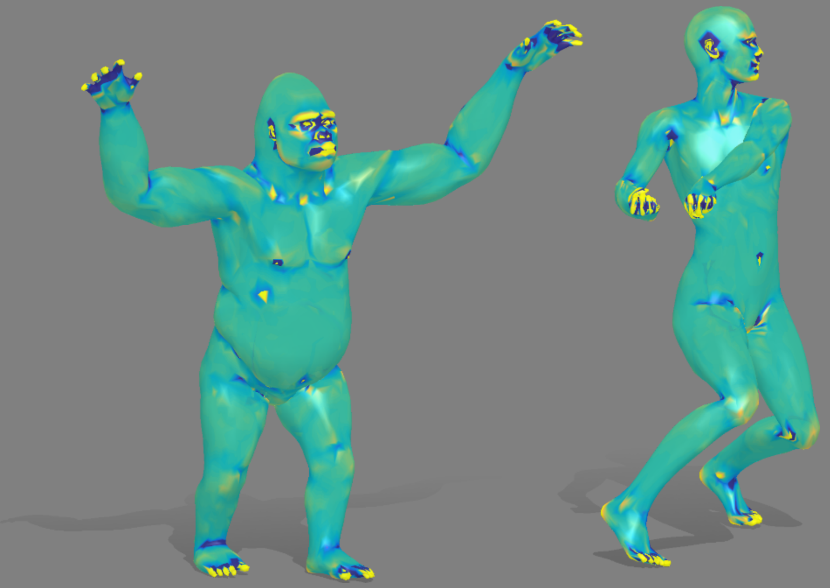
Jing Ren, KAUST
Adrien Poulenard, École Polytechnique
Peter Wonka, KAUST
Maks Ovsjanikov, École Polytechnique

Shape matching



Corresponding descriptors

Gaussian curvatures

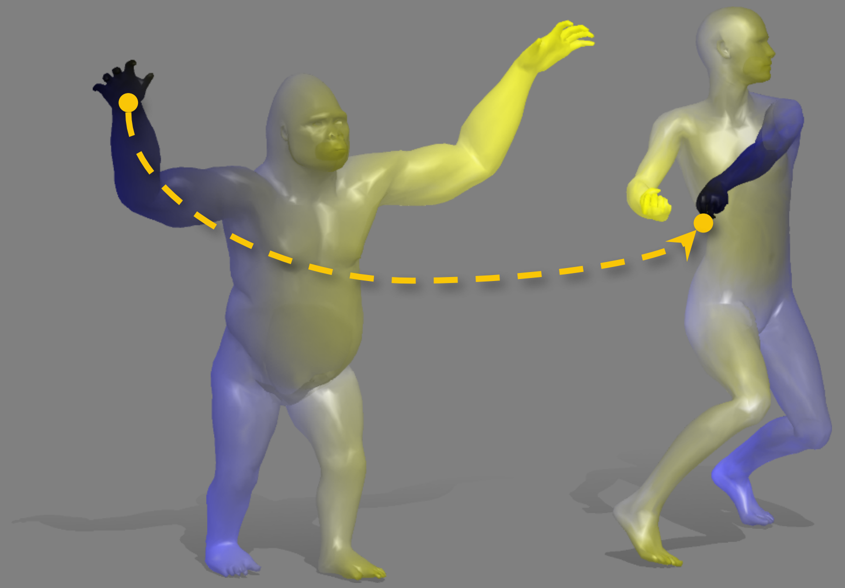


Joint segmentation

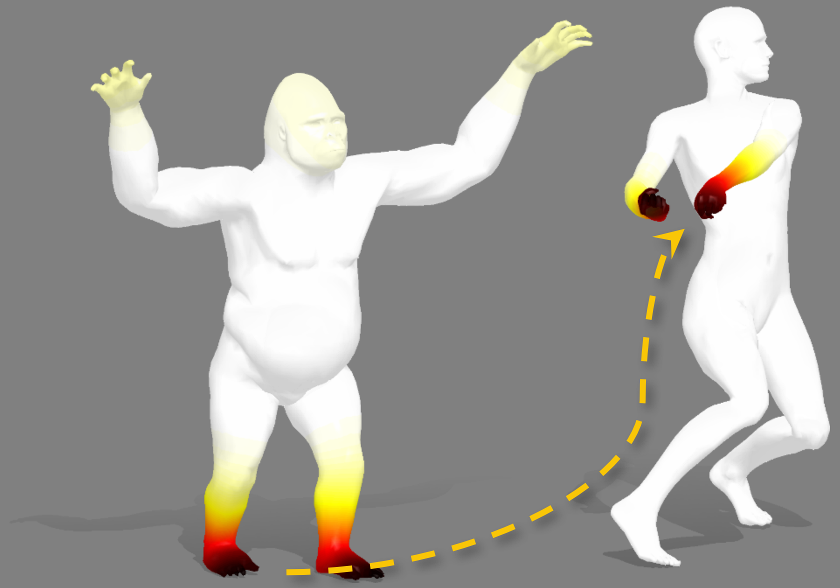
Wave kernel signature

Problems

Left-right ambiguity



Leg-arm ambiguity



How to break the symmetry ambiguity?

Our solution: define orientation-preserving operators in the functional map framework.

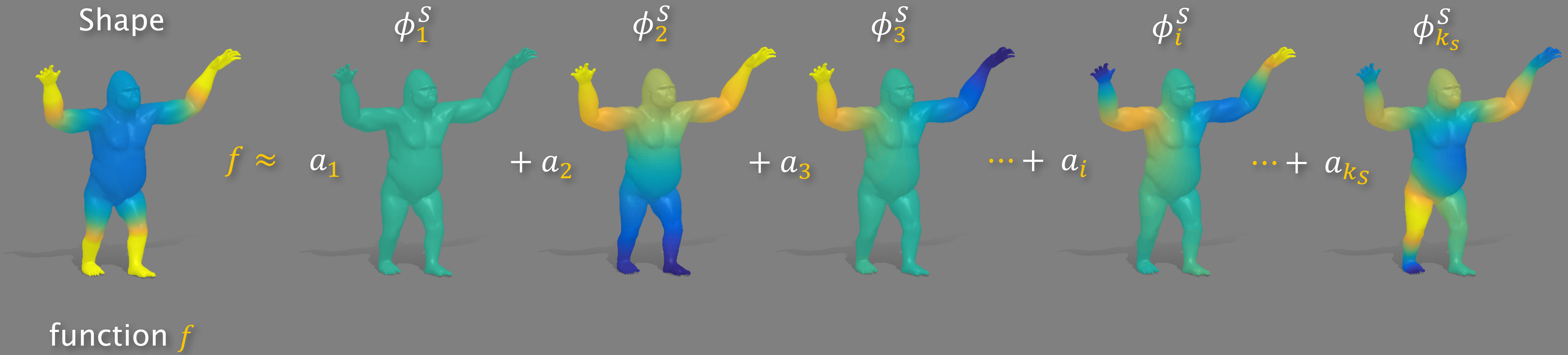
Q1: what is the standard functional map pipeline to solve the shape matching problem?

Q2: how to define the orientation-preserving operator? Why does it work?

Q1: Functional map pipeline

Laplacian–Beltrami operator

Q1: Functional map pipeline — Laplacian–Beltrami operator



Q1: Functional map pipeline

Source

$f \approx \Phi^S a$

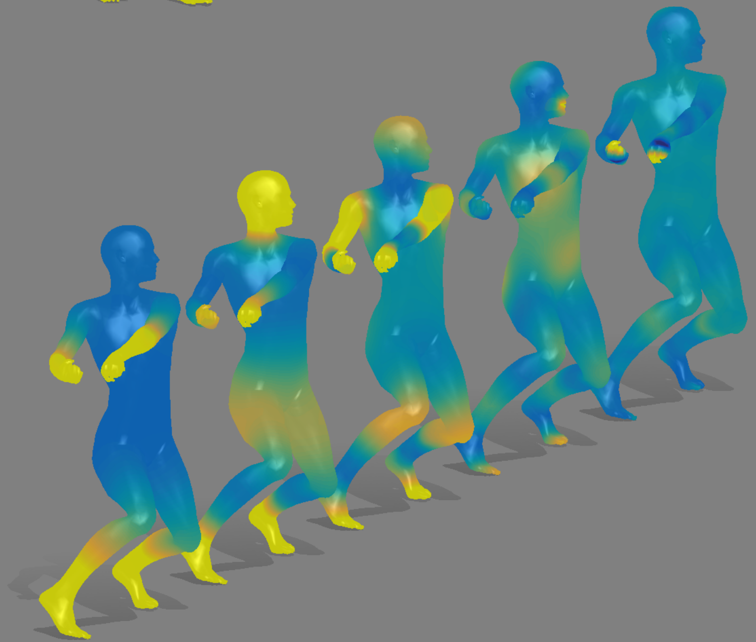
$$\{f_i\}_{i=1}^k \xrightarrow{\Phi^S} \{a_i\}_{i=1}^k$$

$$C a_i = b_i \quad \forall i = 1, \cdots k$$

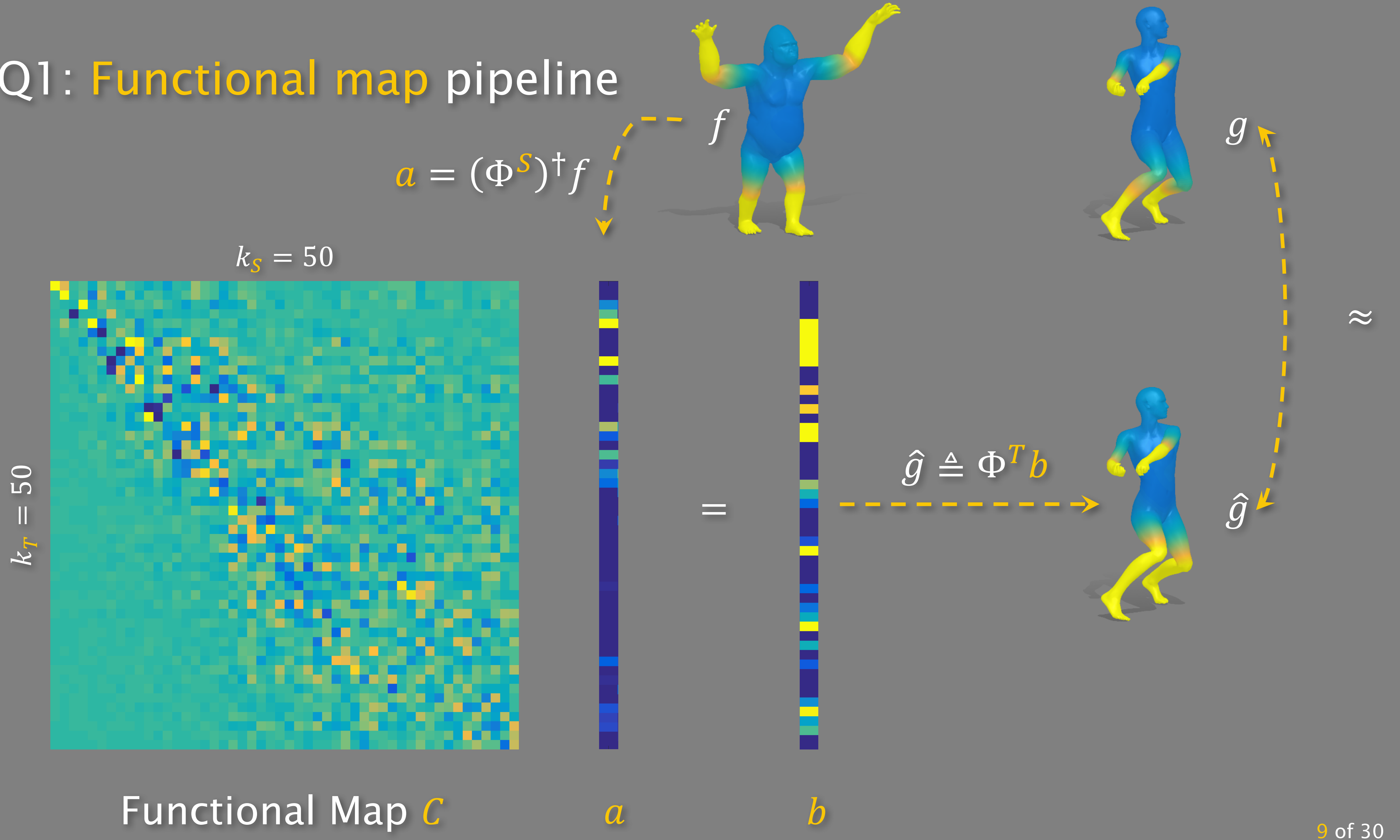
$$\{g_i\}_{i=1}^k \xrightarrow{\Phi^T} \{b_i\}_{i=1}^k$$

Target

$g \approx \Phi^T b$



Q1: Functional map pipeline



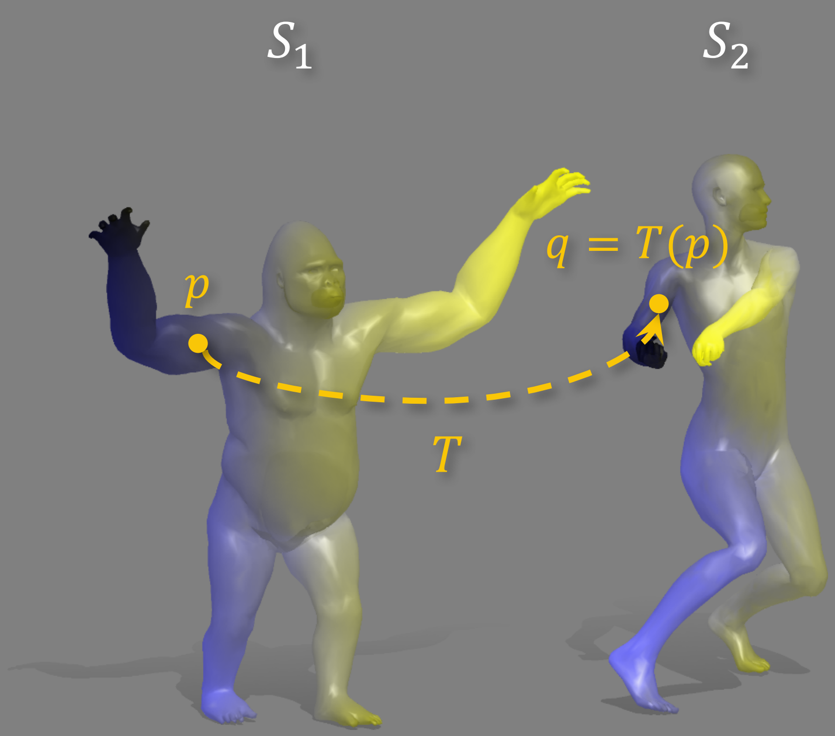
Q1: Functional map pipeline

How to find C ?

Given a pair of shapes S_1, S_2, with Laplacian operators $\Delta_{S_1}, \Delta_{S_2}$

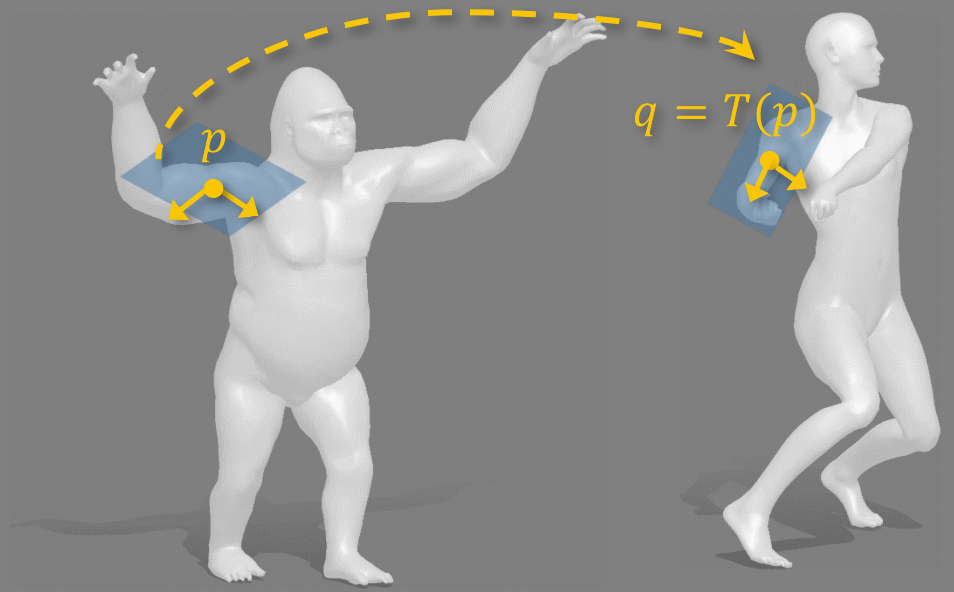
- Compute the functional bases on the two shapes. Store them in matrices Φ^{S_1}, Φ^{S_2}
- Project the corresponding descriptors $\{f_i, g_i\}_{i=1}^{k}$ into the functional space. Store the coefficients in matrices $A = [a_1 \cdots a_k], B = [b_1 \cdots b_k]$
- Solve $C^* = \underset{C \in R^{k_T \times k_S}}{\operatorname{argmin}} E(C) = \|CA - B\|^2 + \left\|C\Delta_{S_1} - \Delta_{S_2} C\right\|^2$
- Orientation-preservation term $\sum_{i=1}^{k} \left\|C\Omega_{f_i} - \Omega_{g_i} C\right\|^2$

Q2: Orientation-preserving operators



Map differential

$dT: \mathcal{T}_p(S_1) \to \mathcal{T}_q(S_2)$



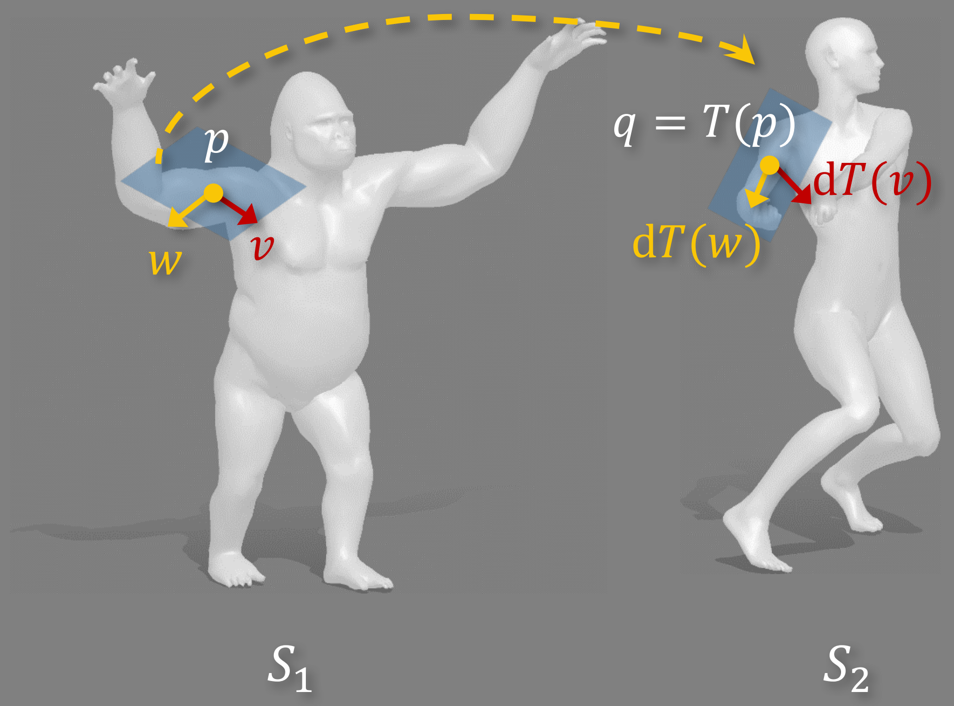
$\mathcal{T}_x(S_i)$: tangent space at $x \in S_i$

A smooth map T is orientation preserving if and only if dT is orientation preserving

Q2: Orientation-preserving operators

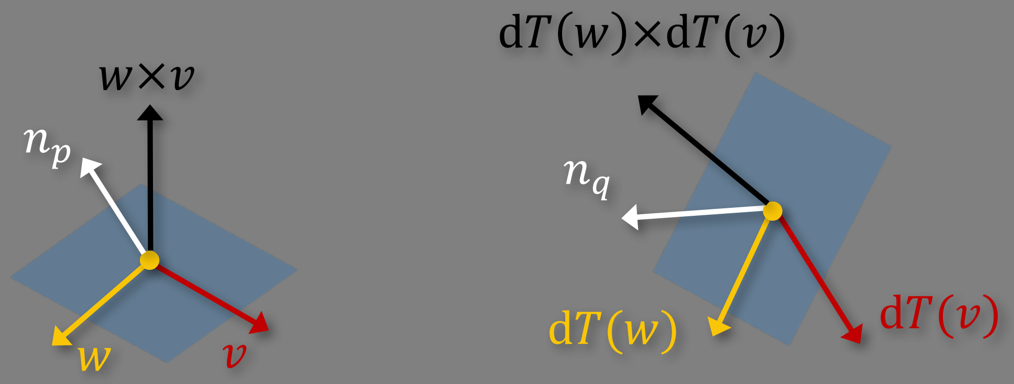
Map differential

$dT: \mathcal{T}_p(S_1) \to \mathcal{T}_q(S_2)$



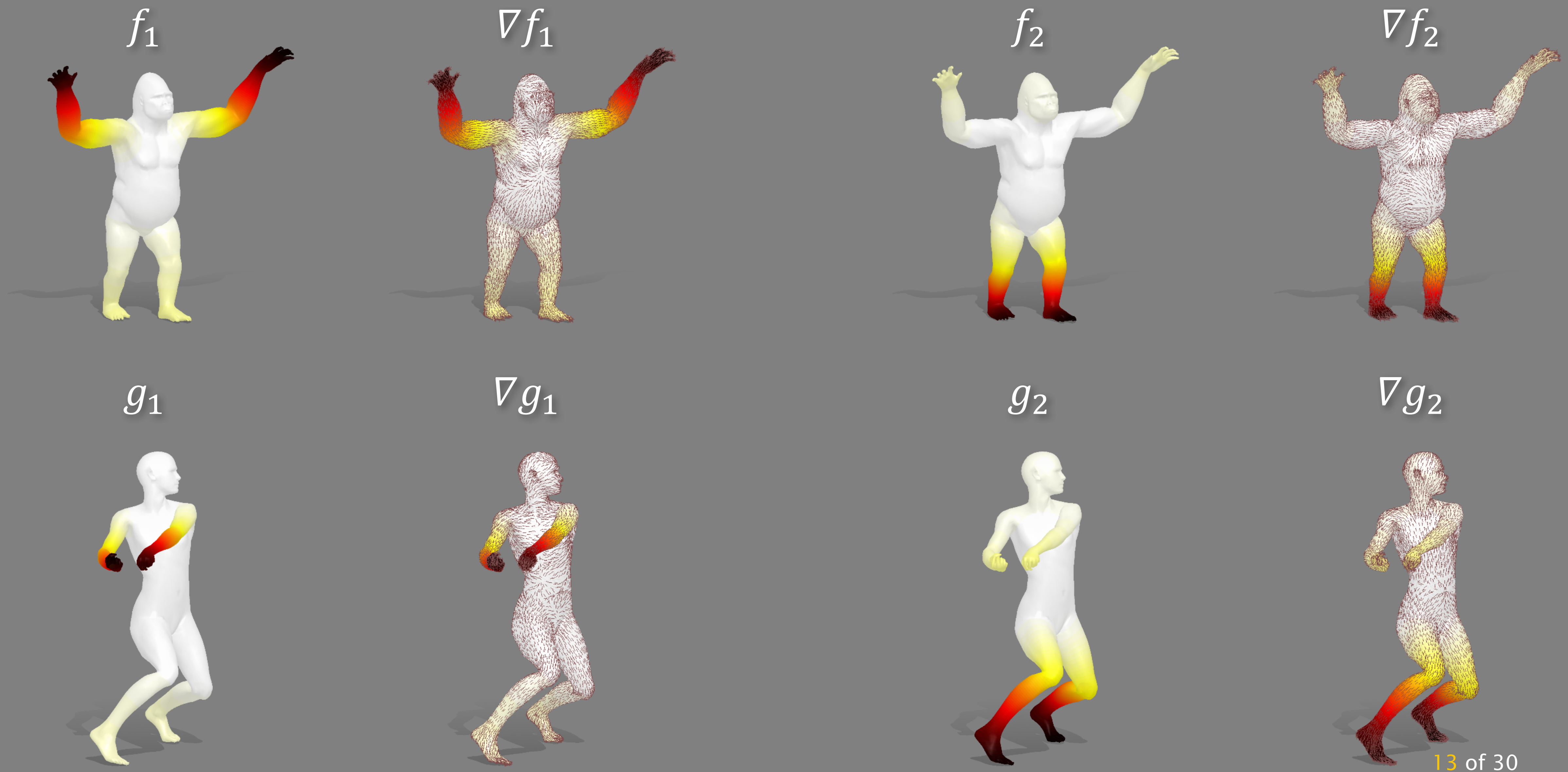
Frame at p: (w, v, n_p)

Frame at $q = T(p)$: $\left(dT(w), dT(v), n_{T(p)}\right)$

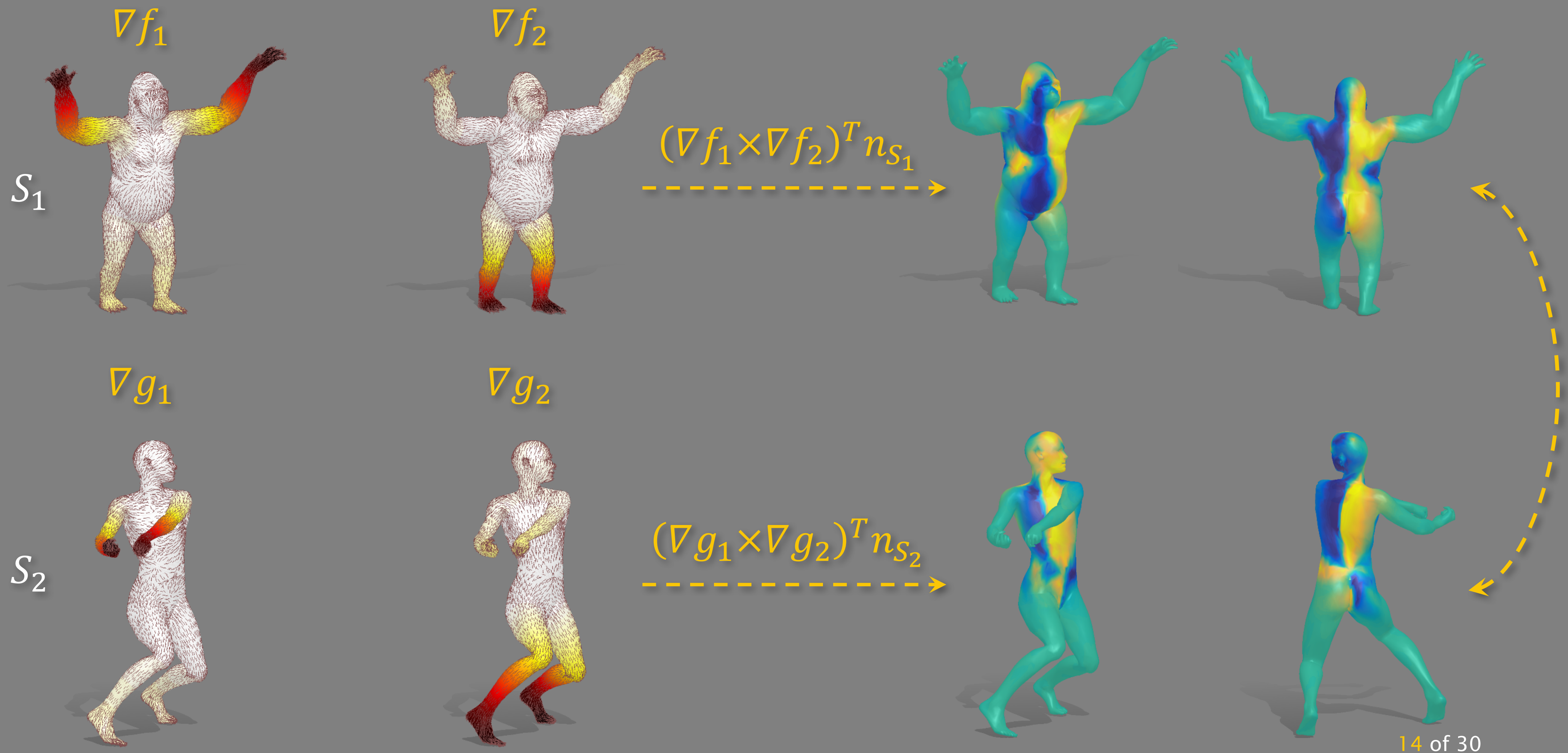


$(w \times v)^T n_p$ should have the same sign as $\left(dT(w) \times dT(v)\right)^T n_{T(p)}$

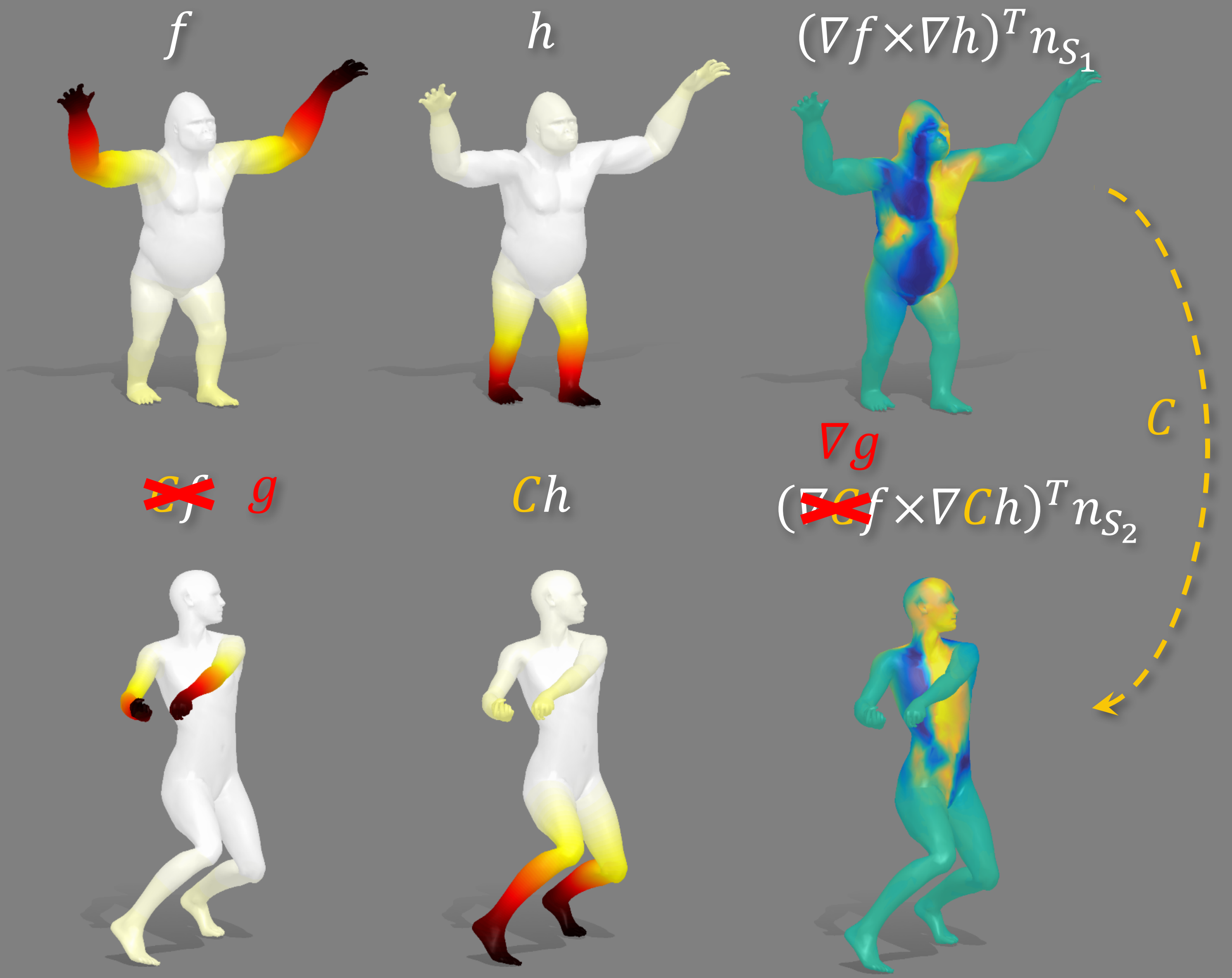
Q2: Orientation-preserving operators



Q2: Orientation-preserving operators



Q2: Orientation-preserving operators



non-linear

linear! – $\Omega(\cdot)$

Q2: Orientation-preserving operators

Shape

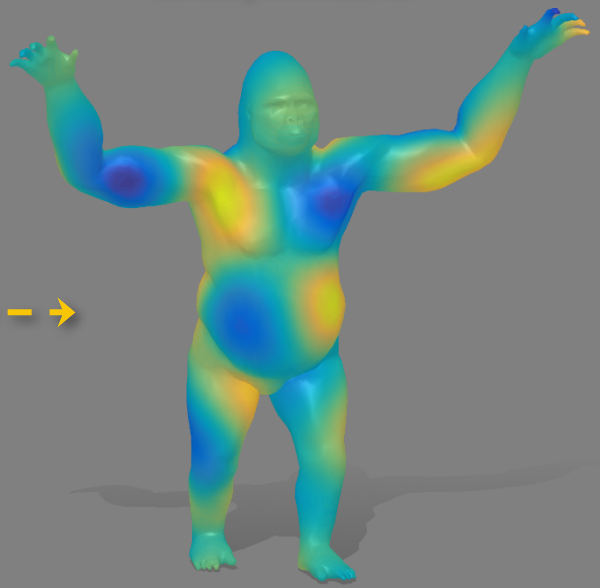
A descriptor function f

With symmetry ambiguity

$\Omega(\cdot)$

$\Omega(f)$

antisymmetric



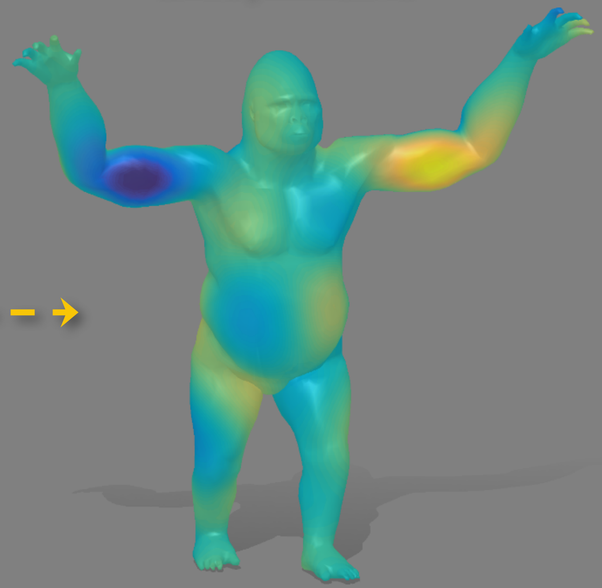
Q2: Orientation-preserving operators

Shape

A descriptor function f
With symmetry ambiguity

$\Omega(\cdot)$

$\Omega(f)$
antisymmetric



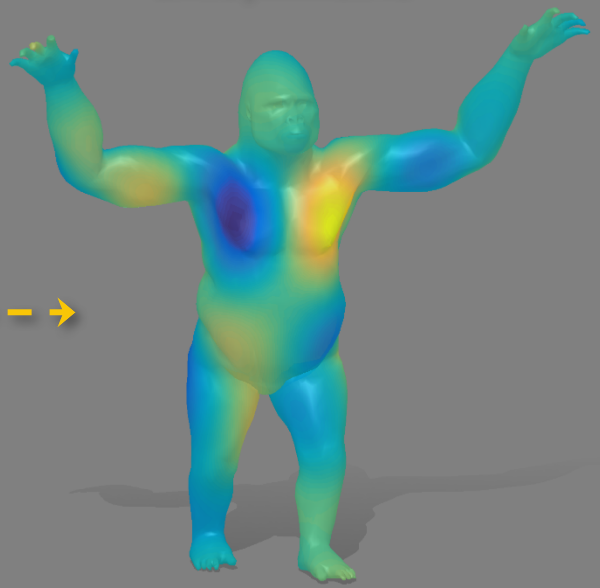
Q2: Orientation-preserving operators

Shape

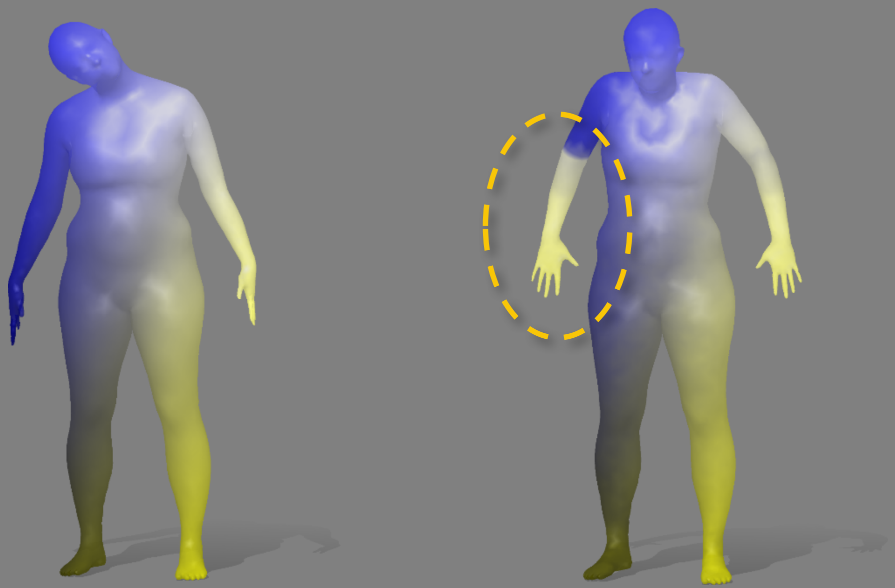
A descriptor function f
With symmetry ambiguity

$\Omega(\cdot)$

$\Omega(f)$
antisymmetric



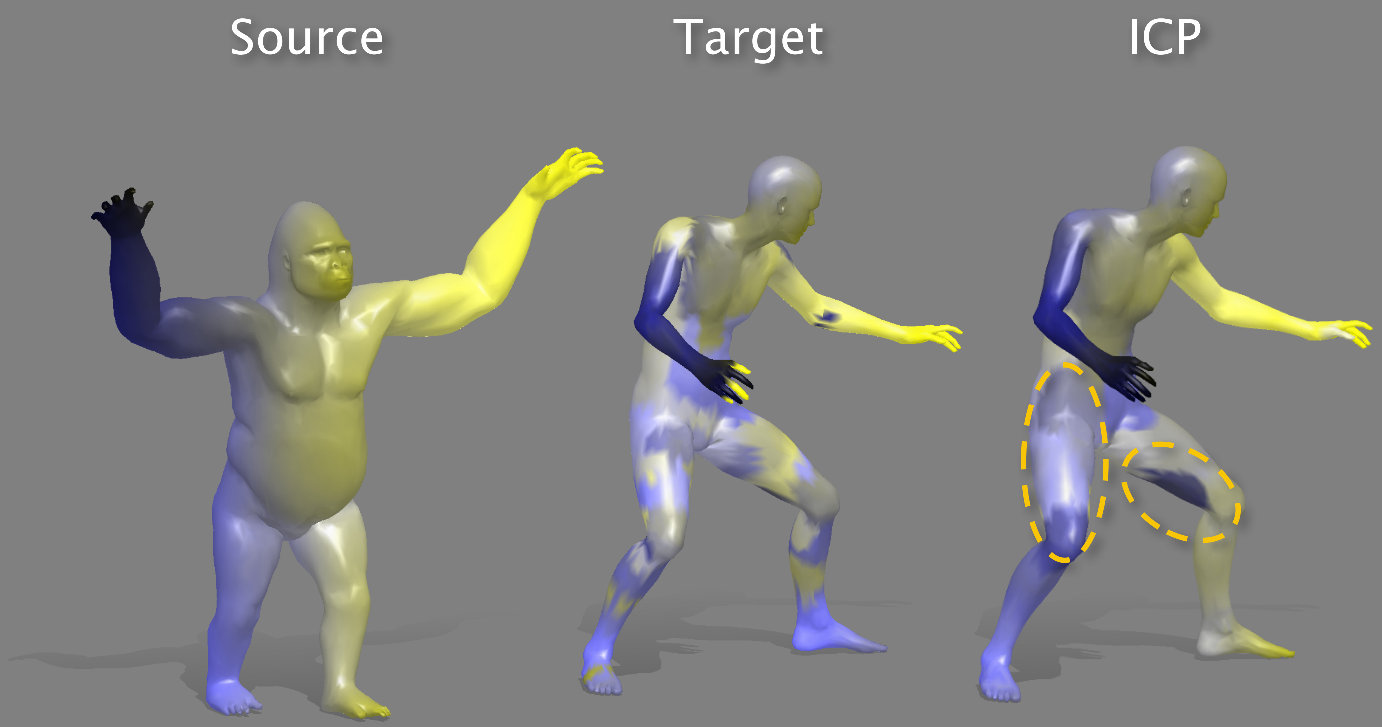
Orientation-preserving operators – defects



Intrinsic – no guarantee

Induced – depends on the descriptors

Refinement! – iterative closest point (ICP)?



Problems of ICP:
- discontinuous
- Only in the spectral domain – no spatial info

A better refinement to improve the quality?

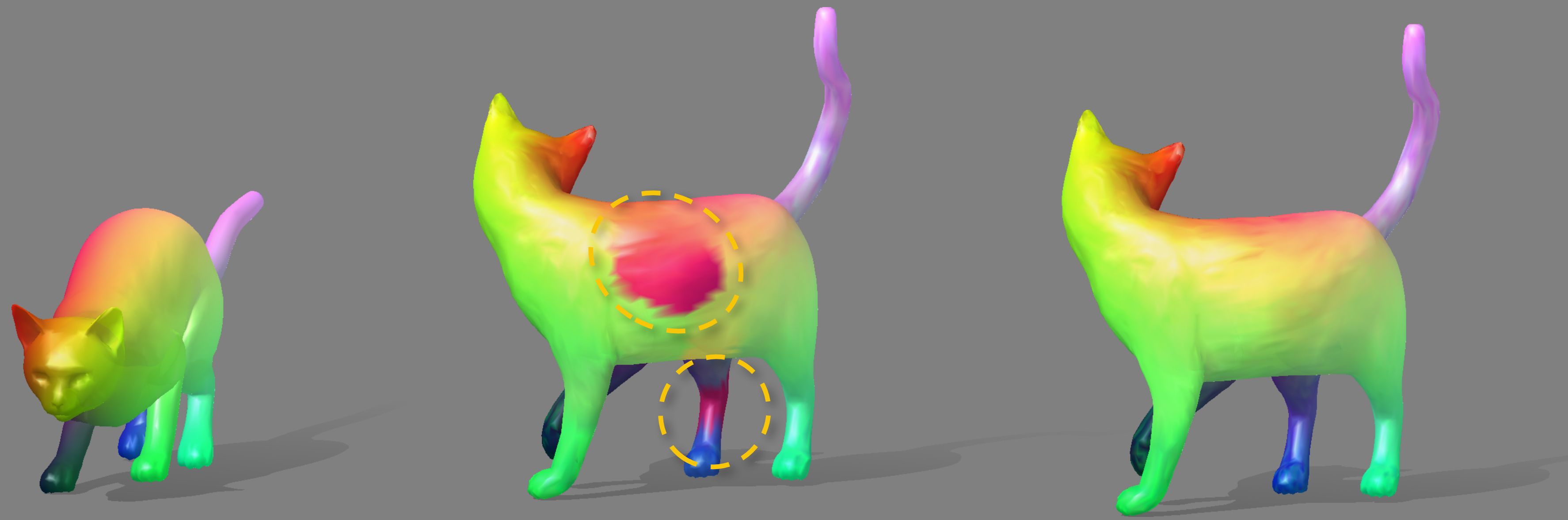
Without ground-truth correspondences,
how do we measure the quality of a map?

Refinement! – improve smoothness

smoother – better!

Refinement! – remove outliers

Source | Target: map 01 | Target: map 02



smoother (no outliers)
– better!

Refinement! – improve coverage

Source | Target: map 01 | Source | Target: map 02

Black region: not covered (no correspondence)

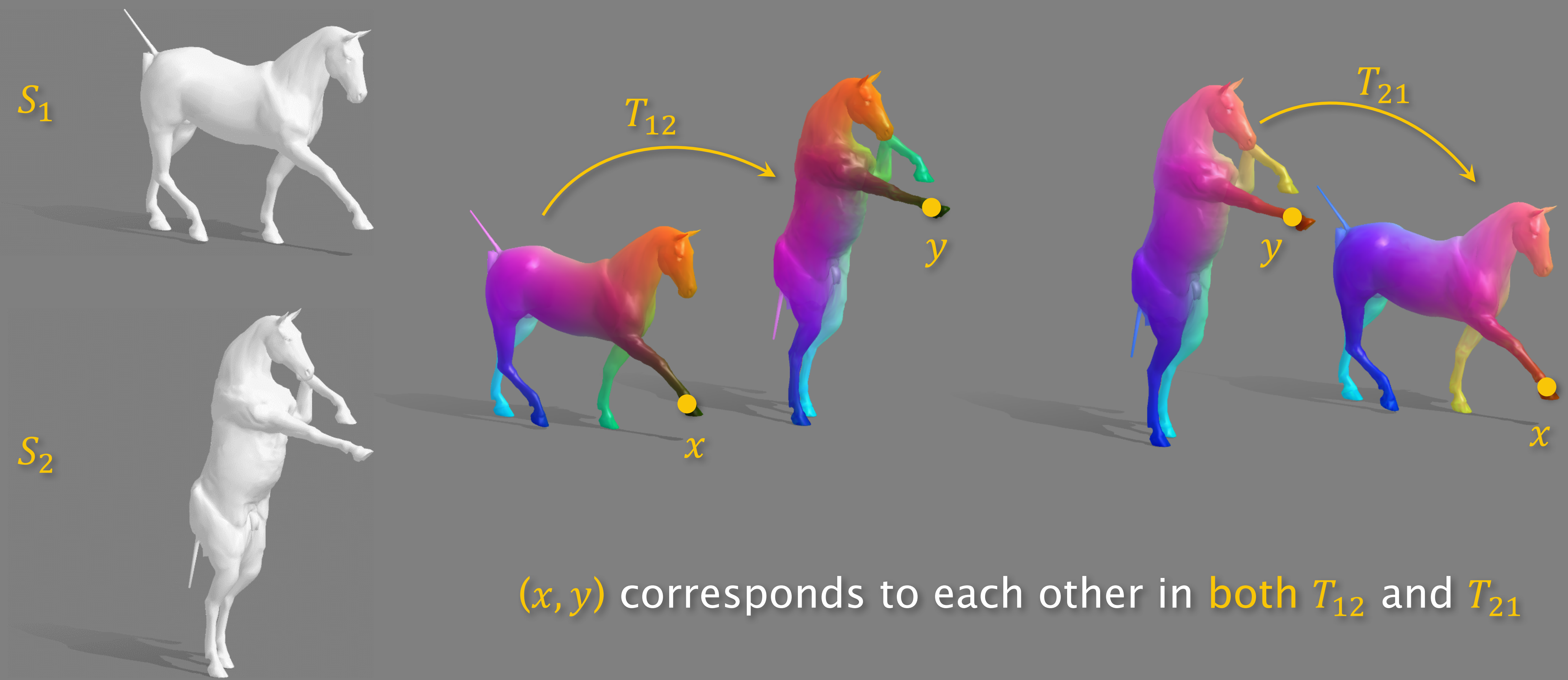
Coverage: 48.9%

Coverage: 81.4%

Coverage: #vertex (or surface area) % covered by the map

More vertices covered – better!

Refinement! – improve bijectivity



(x, y) corresponds to each other in both T_{12} and T_{21}

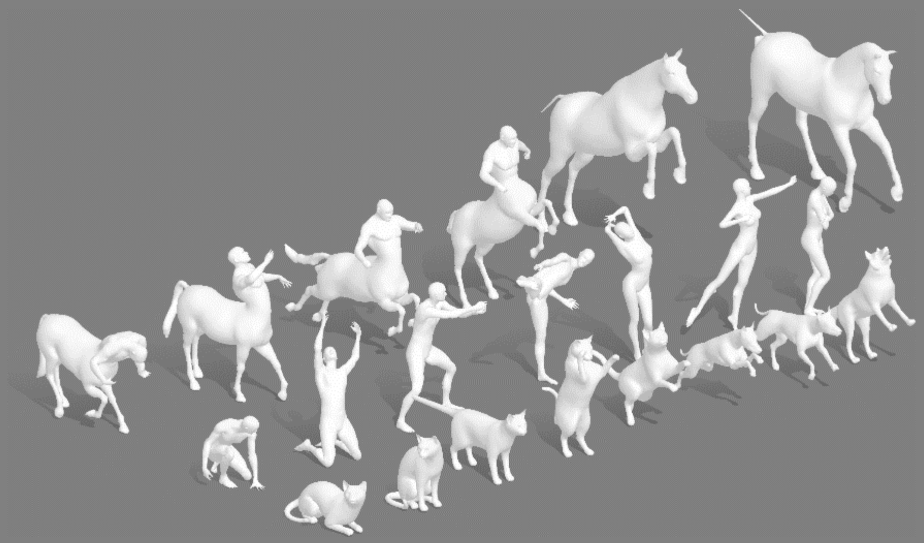
Bijective and Continuous ICP (BCICP)

code

- Bijectivity
  - Soft constraints: $T_{21} \circ T_{12}$ and $T_{12} \circ T_{21}$ are close to identity
- Continuity
  - Smooth the displacement vector field
- Coverage
  - Find the nearest neighbor with the largest preimage size
- Outliers
  - Detected by edge distortion and fixed by the nearest neighbor classified as an inlier

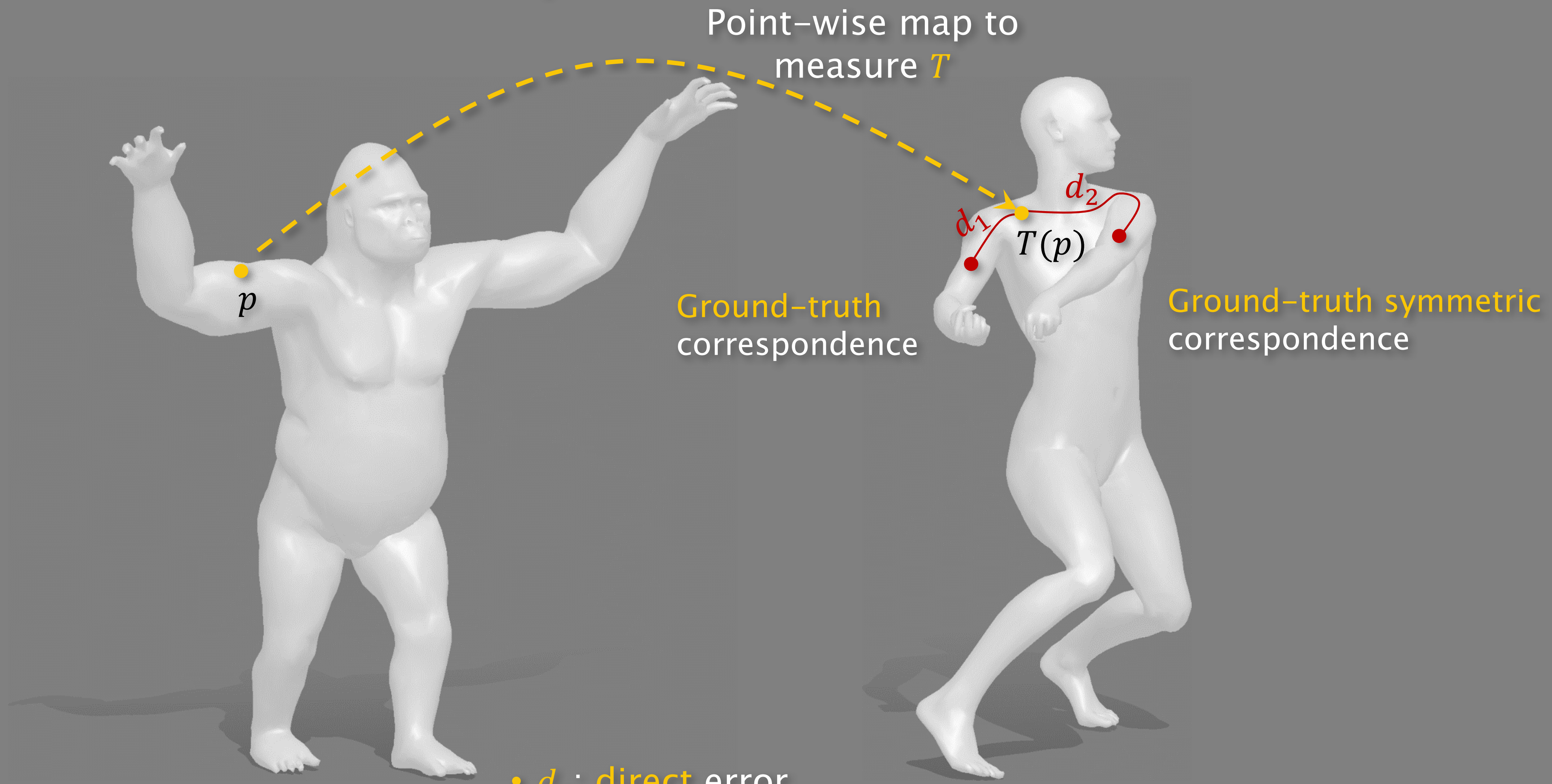
Benchmark datasets

FAUST dataset [Bogo et al. 2014]
- 10 different humans in 10 different poses
- We tested on
  - 200 isometric pairs
  - 400 non-isometric pairs



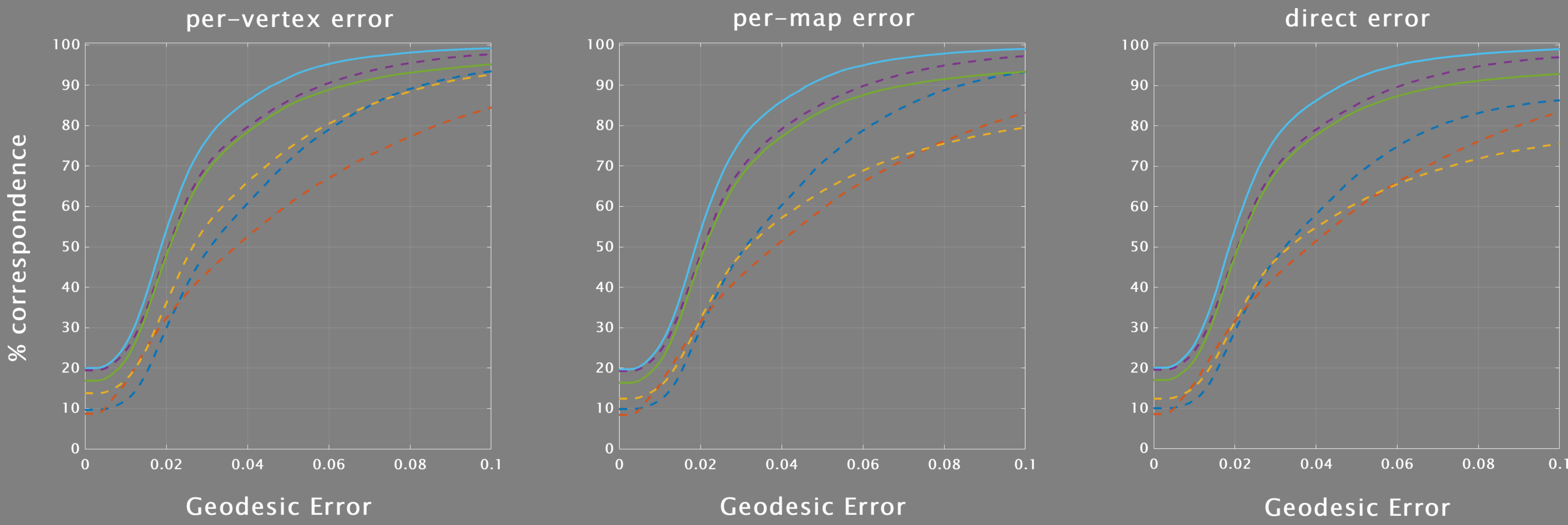
TOSCA dataset [Bronstein et al 2008]
- 80 different humans and animals in 9 categories
- We tested on
  - 568 isometric pairs
  - 190 non-isometric pairs

Measurement – Accuracy



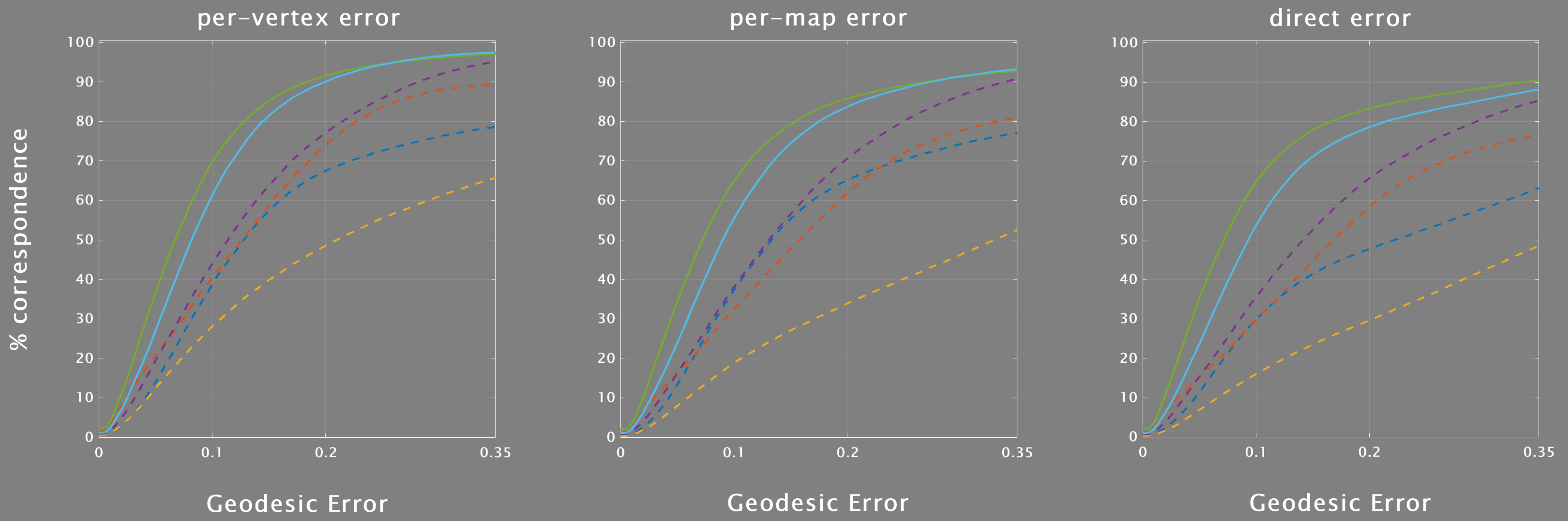
- d_1: direct error
- d_2: symmetric error
- $\min(d_1, d_2)$: per-vertex error

Results – FAUST Isometric dataset (200 pairs)



Solid lines: our methods (with different descriptors)
Dashed lines: state-of-the-art methods

Results – TOSCA non-Isometric dataset (190 pairs)



Solid lines: our methods (with different descriptors)
Dashed lines: state-of-the-art methods

Summary

- Introduce orientation-preserving operator into functional map framework
- Propose a refinement technique, BCICP, which improves the bijectivity, continuity, and coverage
- Verify the usefulness of the orientation-preserving operator and the BCICP refinement on large datasets, w.r.t. different measurements

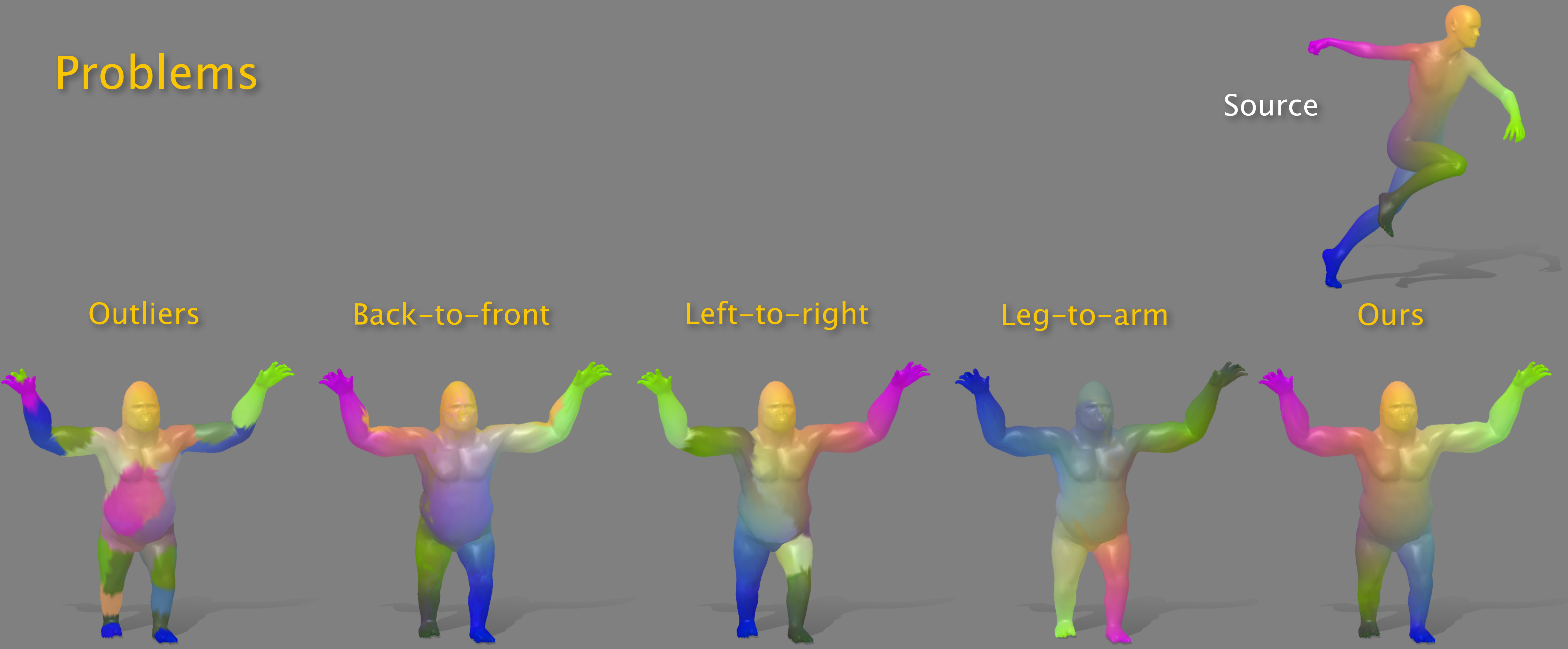
Thanks for your attention ☺

Continuous and Orientation-preserving Correspondences via Functional Maps

Jing Ren, KAUST
Adrien Poulenard, École Polytechnique
Peter Wonka, KAUST
Maks Ovsjanikov, École Polytechnique

Supplementary

Problems



Results – Summary

Dataset (#pairs) / Measurement	FAUST		TOSCA	
	Isometric 200	Non-isometric 400	Isometric 586	Non-isometric 190
Per-vertex	17.8%	24.1%	31.4%	38.3%
Per-map	17.5%	18.4%	14.6%	37.1%
Direct	17.5%	18.4%	38.8%	43.5%

Relative improvement of our method (directOp + BCICP)
over the best baseline methods
w.r.t. the average error of three measurements

Q2: Orientation-preserving operators

Encode into functional map framework:

- Given a functional map C maps function on S_1 to function on S_2
- For every pair of corresponding descriptor $(f, g), f \in \mathcal{F}(S_1), g \in \mathcal{F}(S_2)$
- We can add the orientation-preserving constrain $C\left((\nabla f \times \nabla h)^T n_{S_1}\right) = \left(\nabla g \times \nabla C(h)\right)^T n_{S_2}$
  - $\nabla f, \nabla g$: vector field on the source and target mesh resp.
  - $h, C(h)$: functions defined on the source and target resp.
  - $\nabla h, \nabla C(h)$: vector field
  - $(\nabla f \times \nabla h)^T n_{S_1}$ defines a function on the source
  - $\left(\nabla g \times \nabla C(h)\right)^T n_{S_2}$ defines a function on the target
  - they should correspond to each other! – use functional map to transport the function
- $C\left((\nabla f \times \nabla h)^T n_{S_1}\right) = \left(\nabla g \times \nabla C(h)\right)^T n_{S_2}$
- $\text{OrientationPreserving}(C)$ for every pair of corresponding descriptors (f, g)

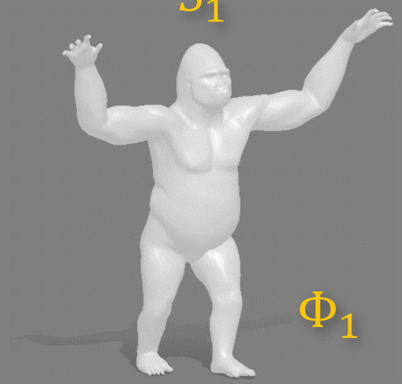
Q2: Orientation-preserving operators

- Define $\omega(w, v, n) = (w \times v, n)$
- If T is orientation preserving, we have $\text{sign}\left(\omega(w, v, n_p)\right) = \text{sign}\left(\omega(\mathrm{d}T(w), \mathrm{d}T(v), n_{T(p)})\right)$ for any pair of $w, v \in \mathcal{T}_p(S_1)$, and for any vertex p

Encode into functional map framework:

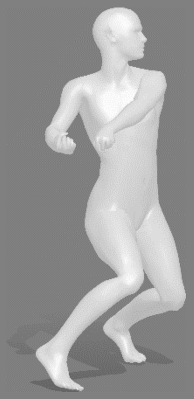
- Given a pair of corresponding descriptor $(f, g), f \in \mathcal{F}(S_1), g \in \mathcal{F}(S_2)$
- $\forall\ h \in \mathcal{F}(S_1)$, it is mapped to $C(h) \in \mathcal{F}(S_2)$ via a functional map C
- Orientation-preserving:
  - For any point $p \in S_1$ (corresponds to $q \in S_2$ via C)
  - $\omega(\nabla f(p), \nabla h(p), n_p) \approx \omega\left(\nabla g(q), \nabla\left(C(h)\right)(q), n_q\right)$
  - Define a function $\Omega(\cdot,\cdot,\cdot) \in \mathcal{F}(S_1)$ such that $\Omega(p) = \omega(\cdot_p, \cdot_p, \cdot_p)$
  - $C\left(\Omega(\nabla f, \nabla h, n_{S_1})\right) = \Omega(\nabla g, \nabla C(h), n_{S_2})$
- $\text{OrientationPreserving}(C)$ for every pair of corresponding descriptors (f, g)

BCICP refinement – improve bijectivity



Notation:

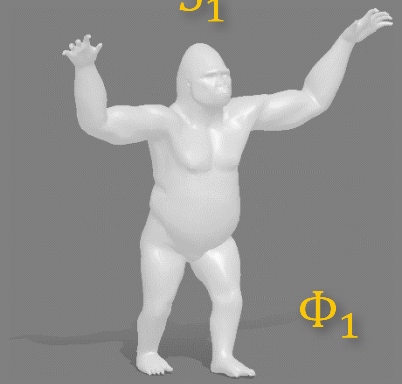
- Point-wise map (vector!) T_{12}: i–th vertex on S_1 is mapped to $T_{12}(i)$–th vertex on S_2
- Shape S_i has Laplacian–Beltrami Basis Φ_i
- Functional map (matrix!) C_{12}: maps the functional space $\mathcal{F}(S_1|\Phi_1)$ to $\mathcal{F}(S_2|\Phi_2)$

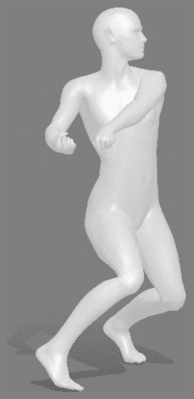


Recall ICP:

- C_{12} is associated with T_{21}
  - Arbitrary function $f \in \mathcal{F}(S_1)$ can be transported to S_2 in two ways:
    - Using the point–wise map directly, i.e., $g = f(T_{21}) \in \mathcal{F}(S_2)$
    - Use the functional map, i.e., $g = \Phi_2\left(C_{12}\left(\Phi_1^{\dagger} f\right)\right)$
- Two transportations should give similar result – for any f
- Minimize $\|\Phi_2 C_{12} - \Phi_1(T_{21},:)\|^2$
  - ICP: alternatively solve for C_{12} and T_{21}

BCICP refinement – improve bijectivity





Bijective in the spectral domain:

- $\|\Phi_2 C_{12} - \Phi_1(T_{21},:)\|^2 + \|\Phi_2 C_{21} - \Phi_1(T_{12},:)\|^2$

Bijective in the spatial domain:

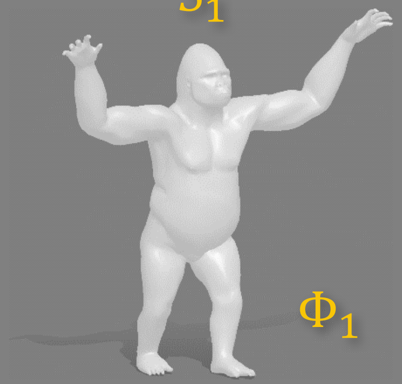
- $T_{21} \circ T_{12}$: maps S_1 to itself (S_1)
- We can add a similar term $\|\Phi_1 C_{11} - \Phi_1(T_{21} \circ T_{12},:)\|^2$
  - where C_{11} is an auxiliary variable, maps the functional space $\mathcal{F}(S_1|\Phi_1)$ to itself
- We can similarly define the energy for $T_{12} \circ T_{21}$

New energy

$$\begin{aligned} &\lambda_1\|\Phi_2 C_{12} - \Phi_1(T_{21},:)\|^2 \\ &+ \lambda_2\|\Phi_2 C_{21} - \Phi_1(T_{12},:)\|^2 \\ &+ \lambda_3\|\Phi_1 C_{11} - \Phi_1(T_{21} \circ T_{12},:)\|^2 \\ &+ \lambda_4\|\Phi_2 C_{22} - \Phi_2(T_{12} \circ T_{21},:)\|^2 \end{aligned}$$

BCICP refinement – improve bijectivity

S_1

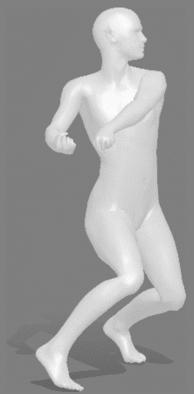


Φ_1

ICP energy

$$\|\Phi_2 C_{12} - \Phi_1(T_{21},:)\|^2$$

S_2

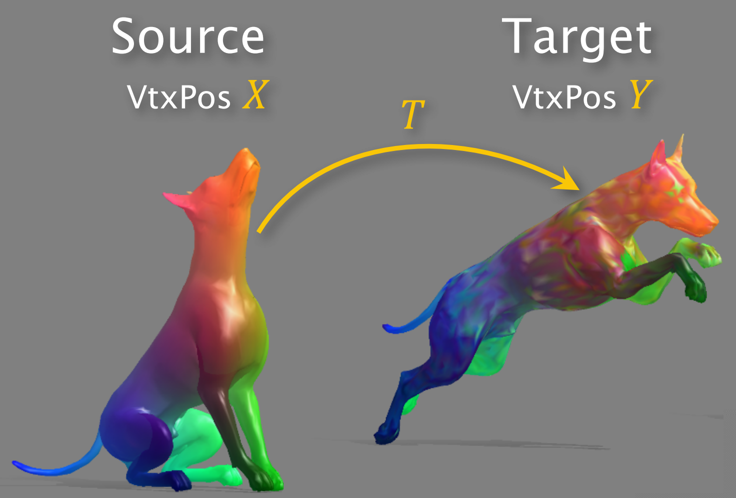


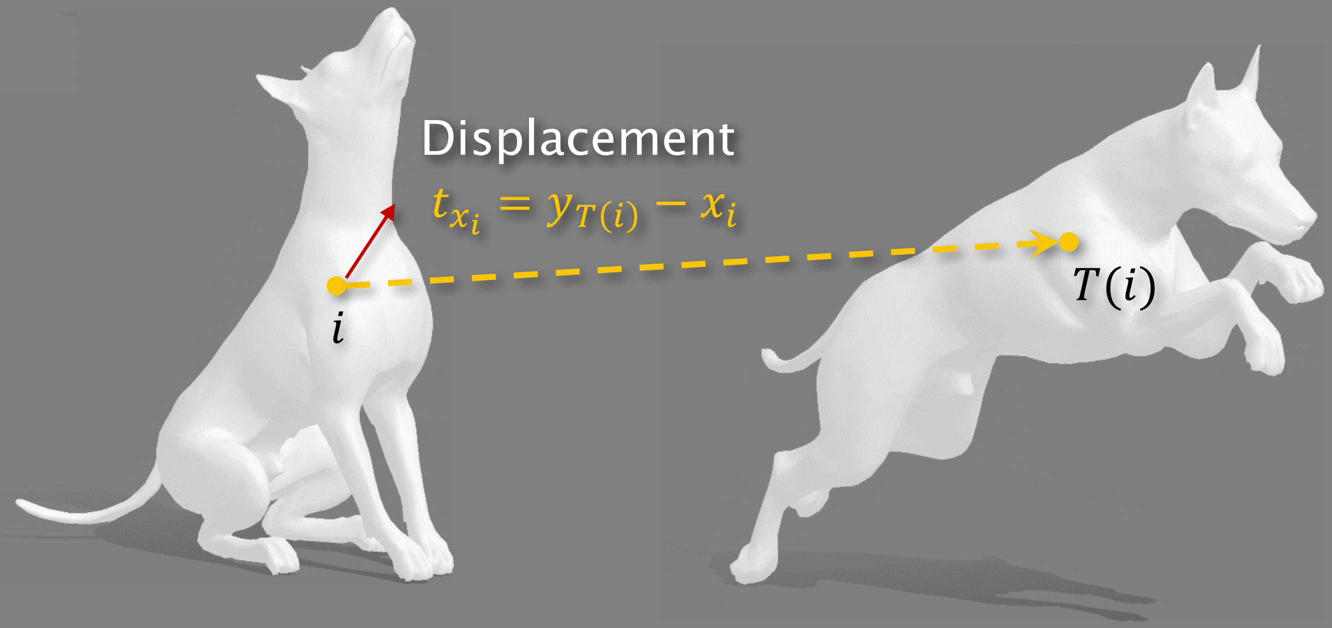
Φ_2

New energy

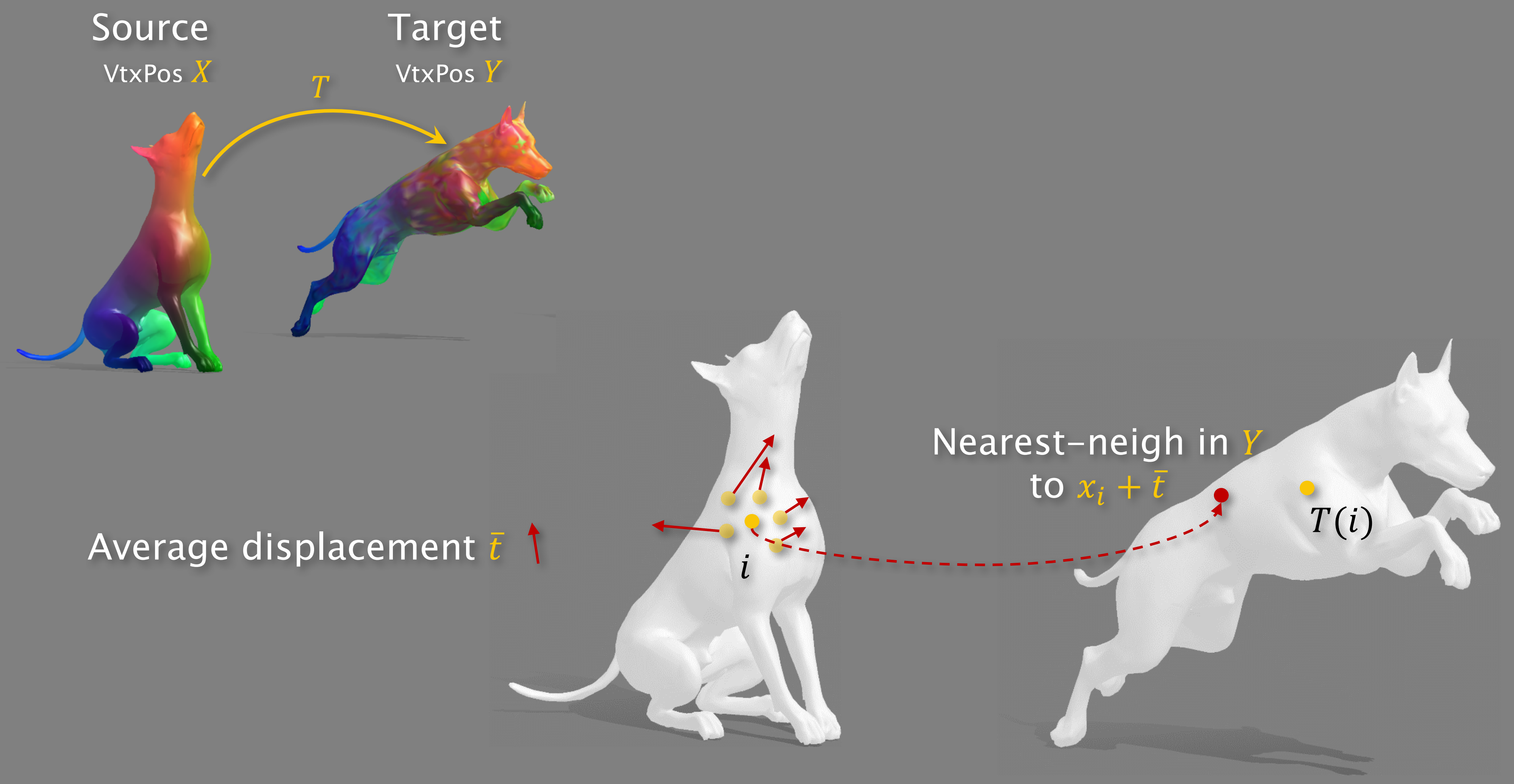
$$\begin{aligned} &\lambda_1\|\Phi_2 C_{12} - \Phi_1(T_{21},:)\|^2 \\ +\, &\lambda_2\|\Phi_2 C_{21} - \Phi_1(T_{12},:)\|^2 \\ +\, &\lambda_3\|\Phi_1 C_{11} - \Phi_1(T_{21} \circ T_{12},:)\|^2 \\ +\, &\lambda_4\|\Phi_2 C_{22} - \Phi_2(T_{12} \circ T_{21},:)\|^2 \end{aligned}$$

BCICP refinement – improve smoothness



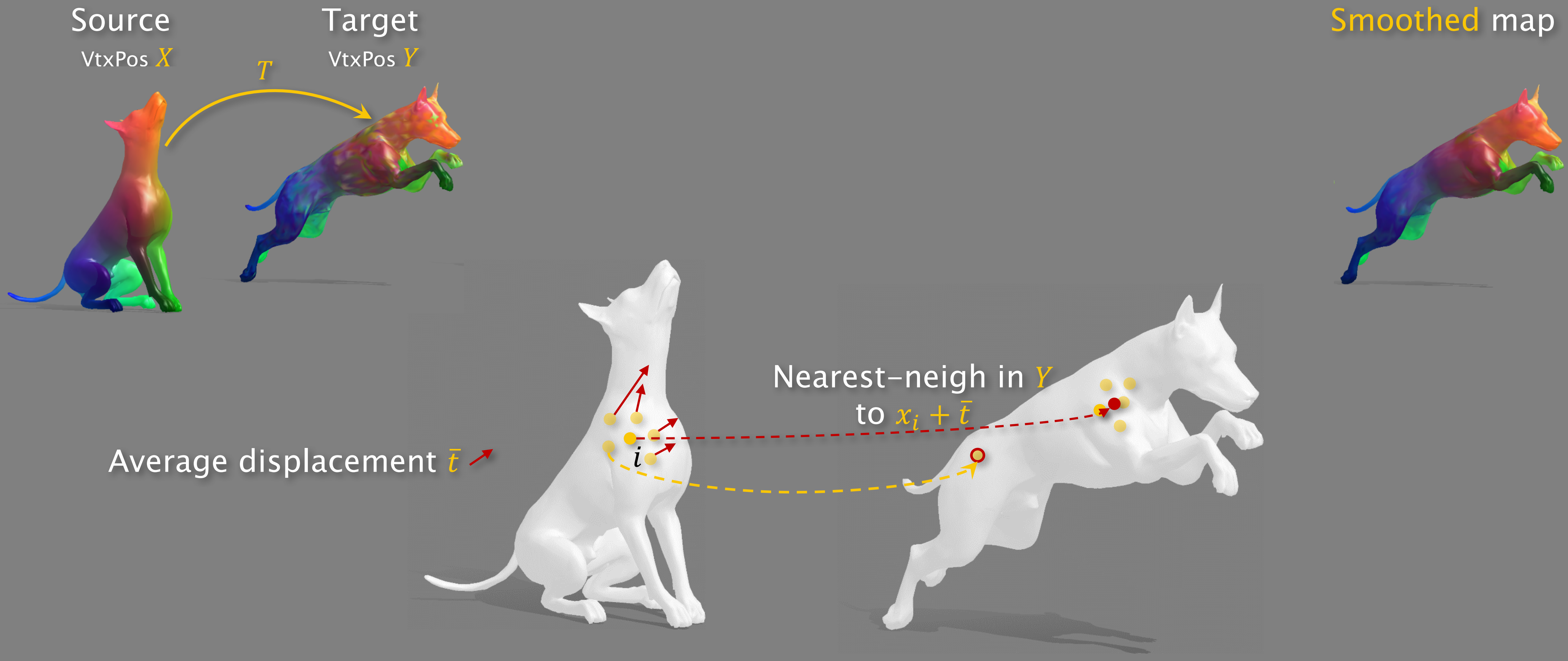


BCICP refinement – improve smoothness



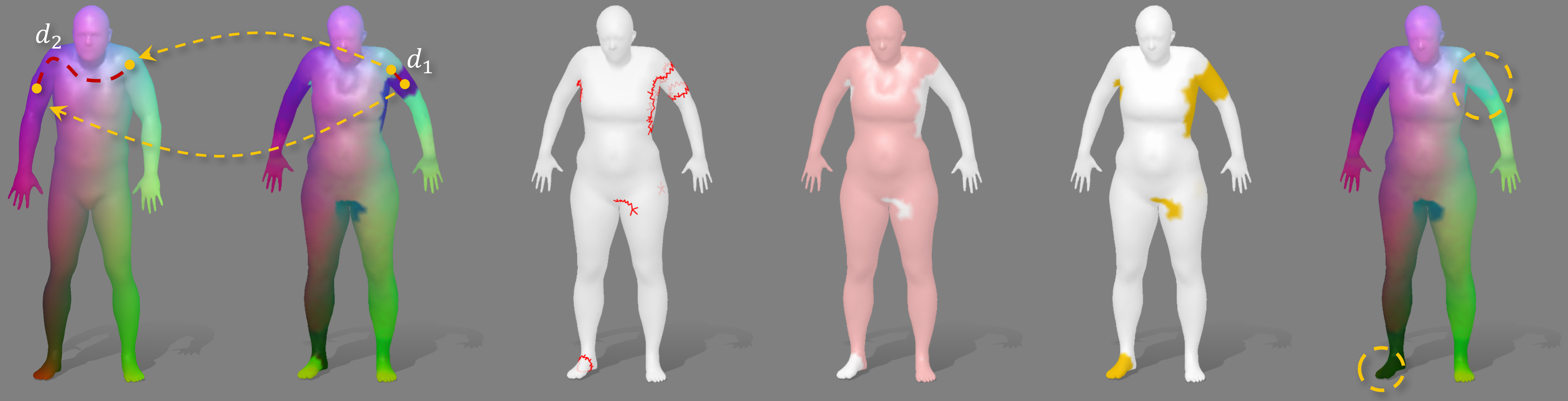
[Papazov and Burschka 2011]

BCICP refinement – improve smoothness



BCICP refinement – remove outliers

Source | Target | Edges with large distortion | Trustworthy vertices | Outlier vertices | Final map



Compute the edge distortion $r = d_2/d_1$

Remove these edges from the adjacency matrix of the mesh

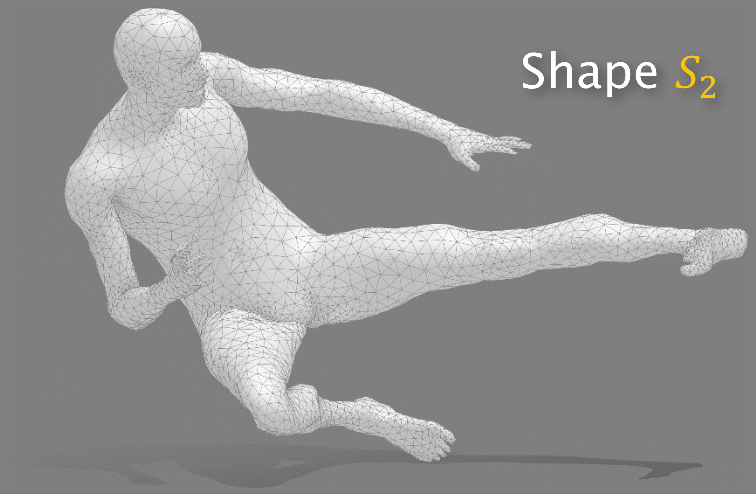
Find the largest connected component

Find the outlier region

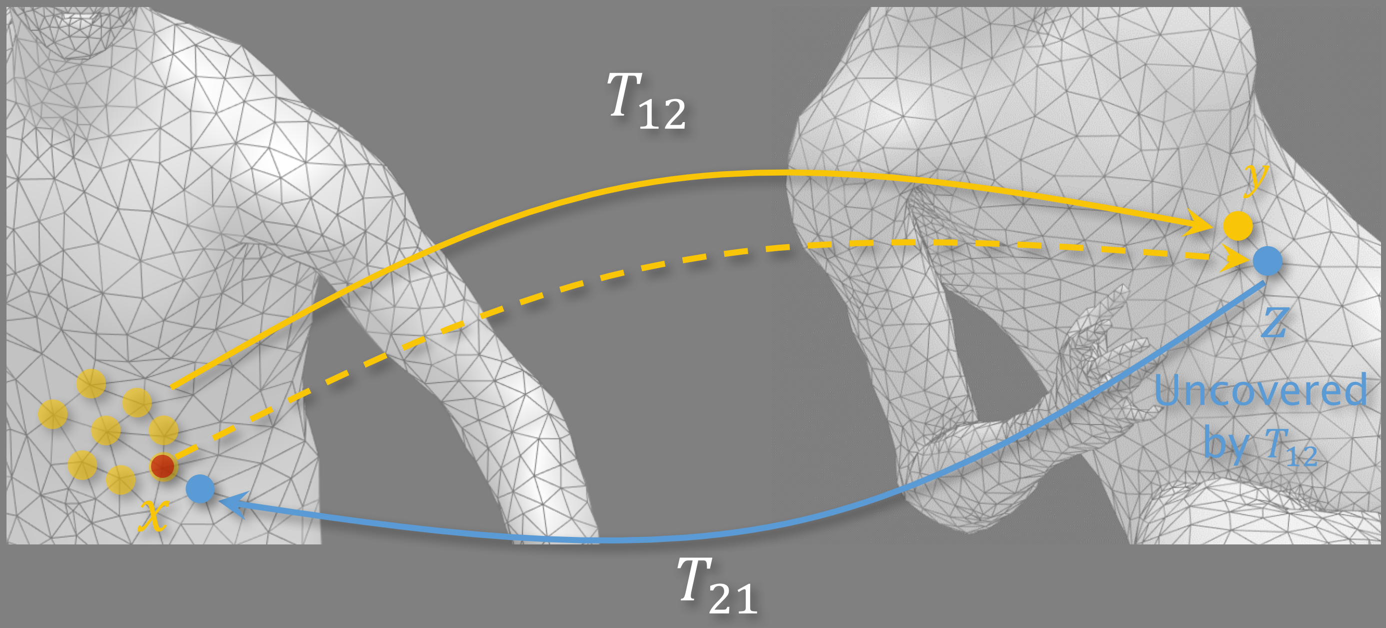
Outlier – find its NN in the "trustworthy" vertex set

BCICP refinement – improve coverage

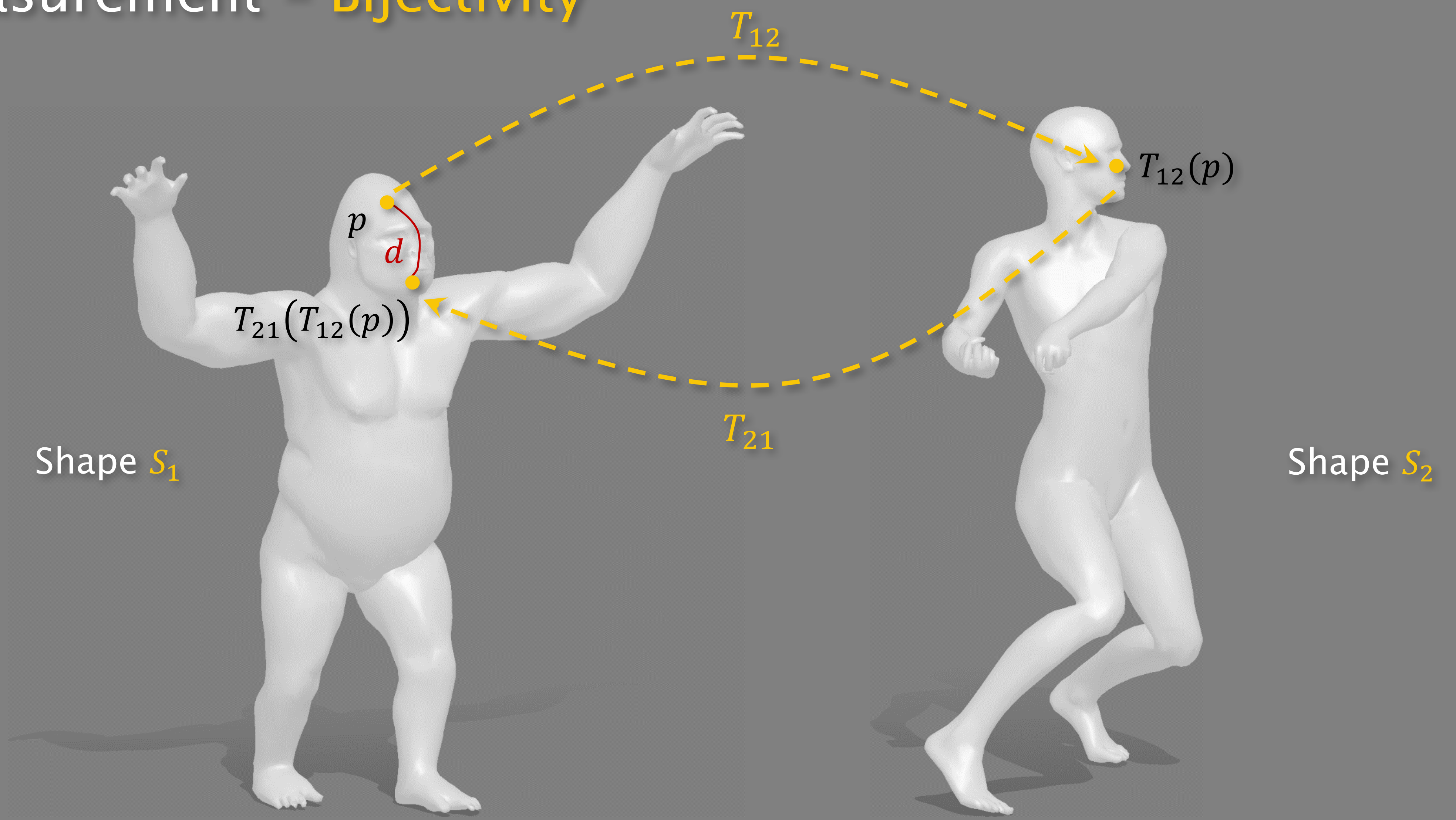
Shape S_1



Shape S_2

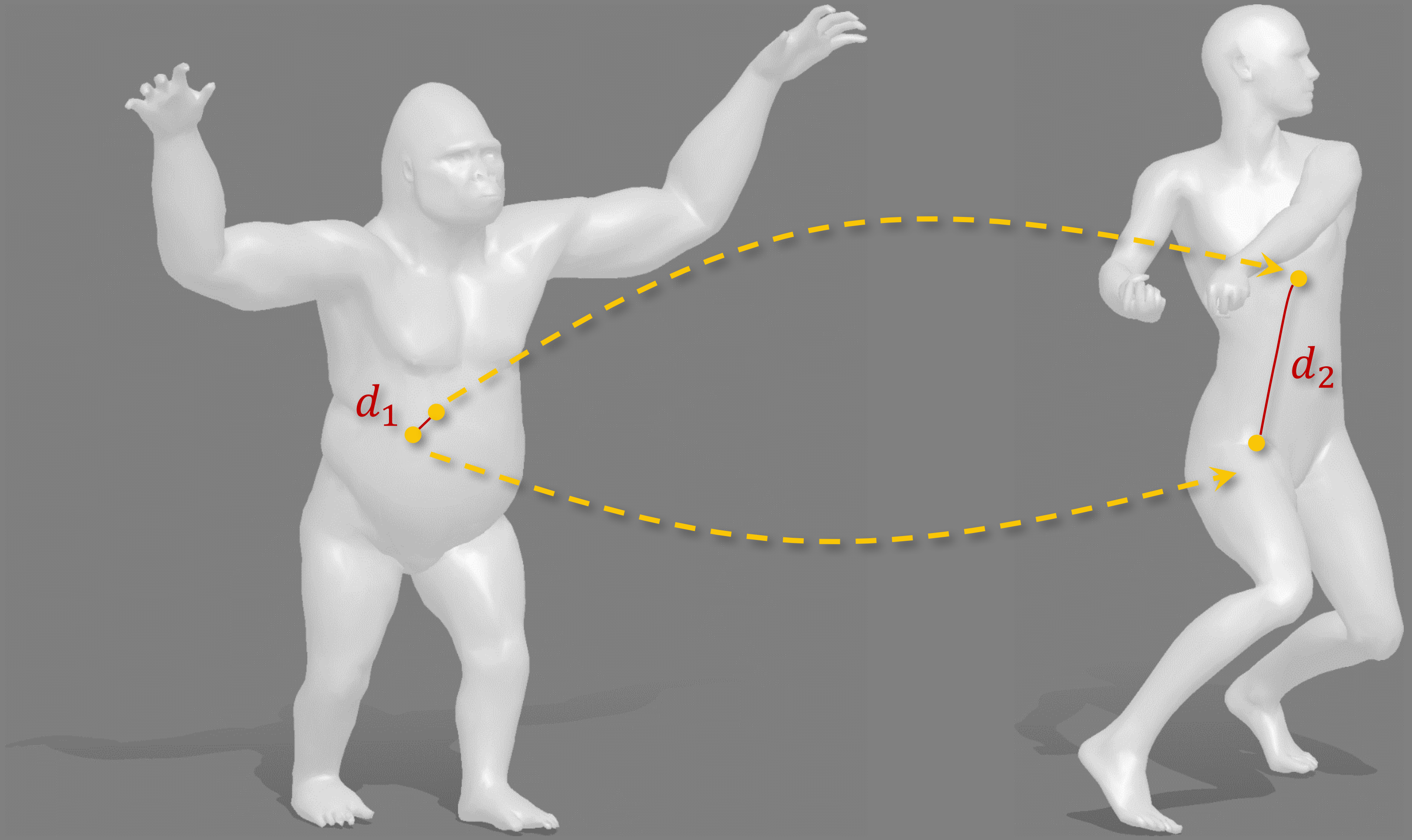


Measurement – Bijectivity



- d: geodesic distance between p and $T_{21}(T_{12}(p))$
- Measure the difference between $T_{21}(T_{12}(\cdot))$ and the identity map

Measurement – Smoothness



- $\frac{d_2}{d_1}$: edge distortion ratio to measure the smoothness

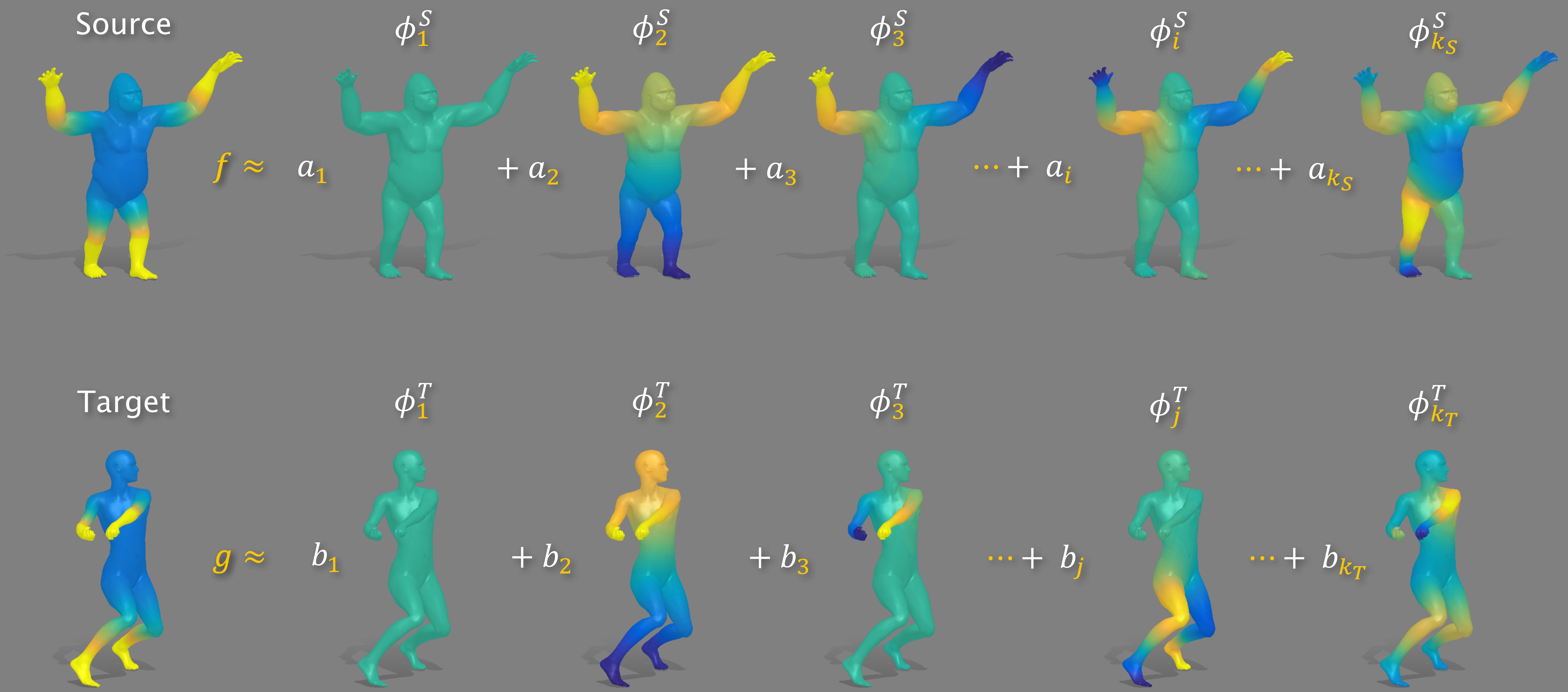
Refinement

Heuristic measurement
- High bijectivity
- High smoothness
  - No outliers
  - High coverage

Our solution – bijective and continuous ICP (BCICP)
- Refine the map simultaneously in the spectral and the spatial domain
- Improve the bijectivity, continuity, and coverage

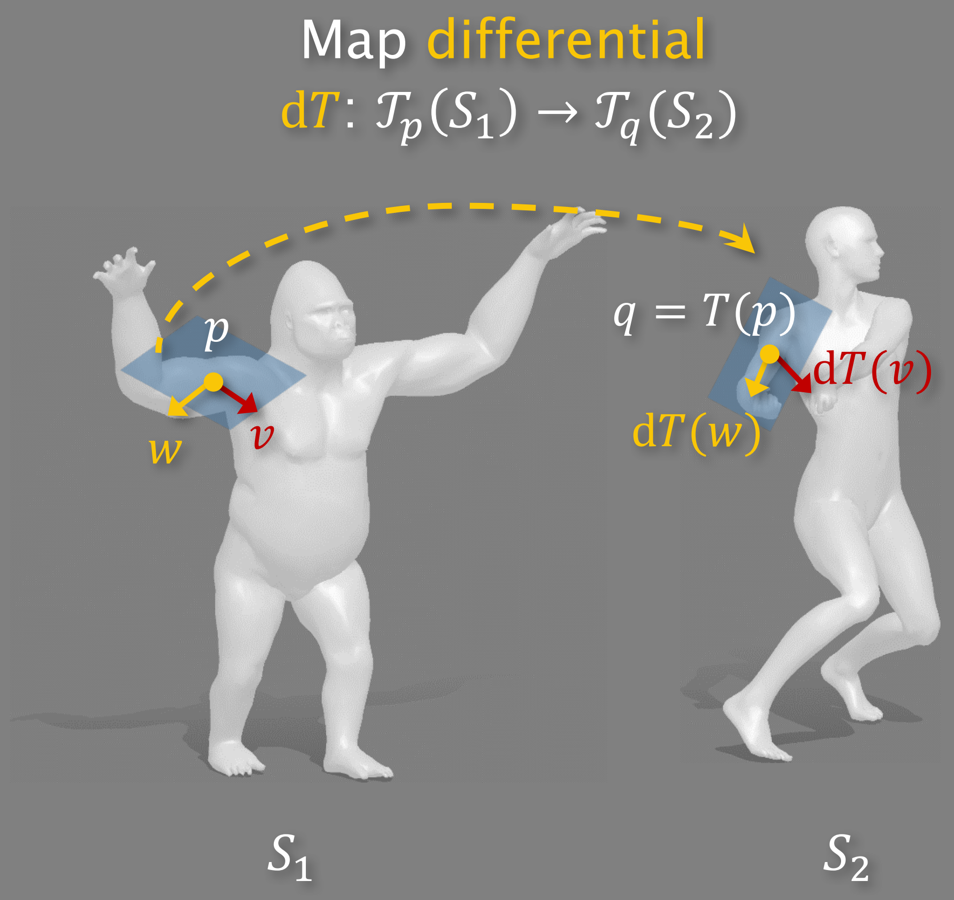
Q1: Functional map pipeline

Descriptors f/g

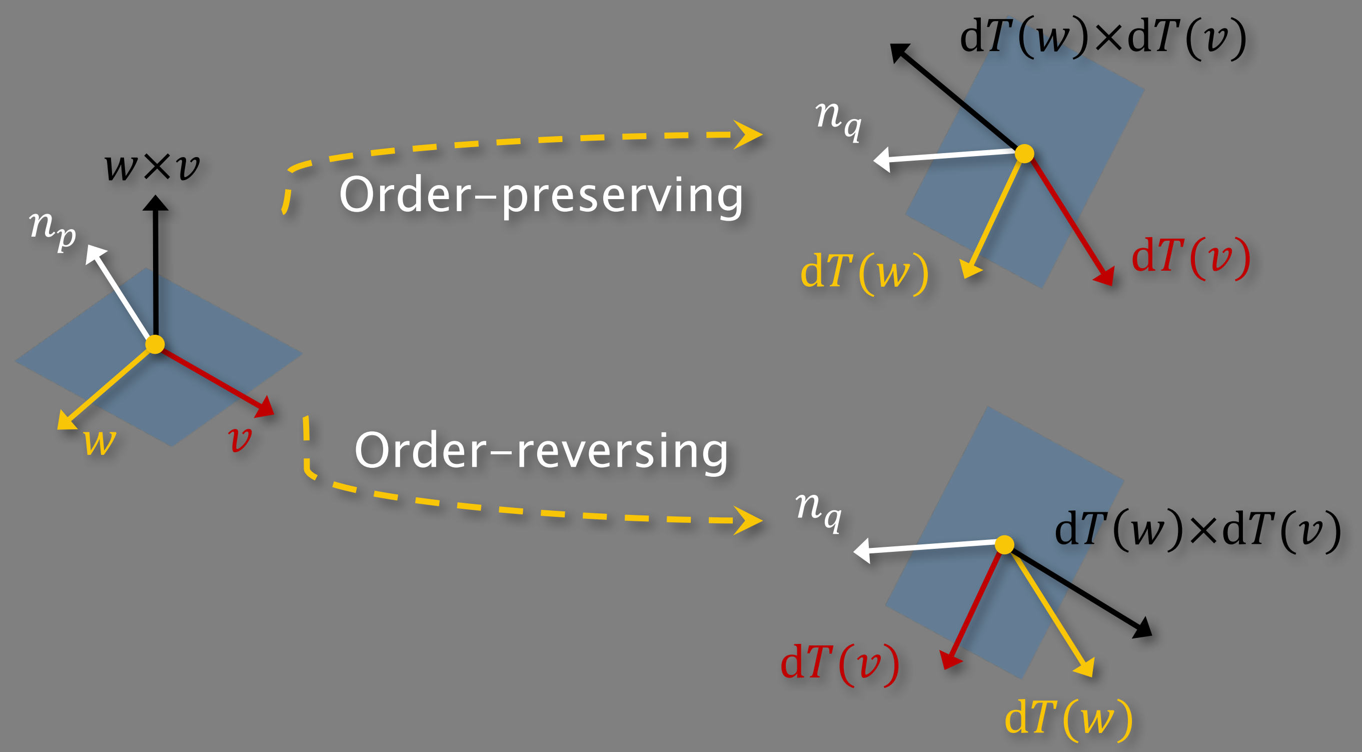


Q1: Functional map pipeline — Laplacian–Beltrami basis

Q2: Orientation-preserving operators



Frame at p: (w, v, n_p)
Frame at $q = T(p)$: $\left(dT(w), dT(v), n_{T(p)}\right)$



$(w \times v)^T n_p$ should have the same sign as $\left(dT(w) \times dT(v)\right)^T n_{T(p)}$

Q1: Functional map pipeline

- First introduced by Ovsjanikov et al in 2012 : "*Functional Maps: A Flexible Representation of Maps between Shapes*"
- Map a function defined on the source shape to another function on the target
- The functions defined on the source/target shape are represented in a compressed form using the Laplacian-Beltrami basis