Label space

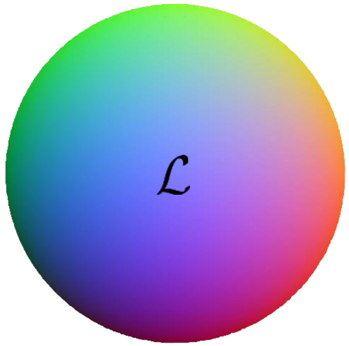
Bijections between shapes

$\downarrow$

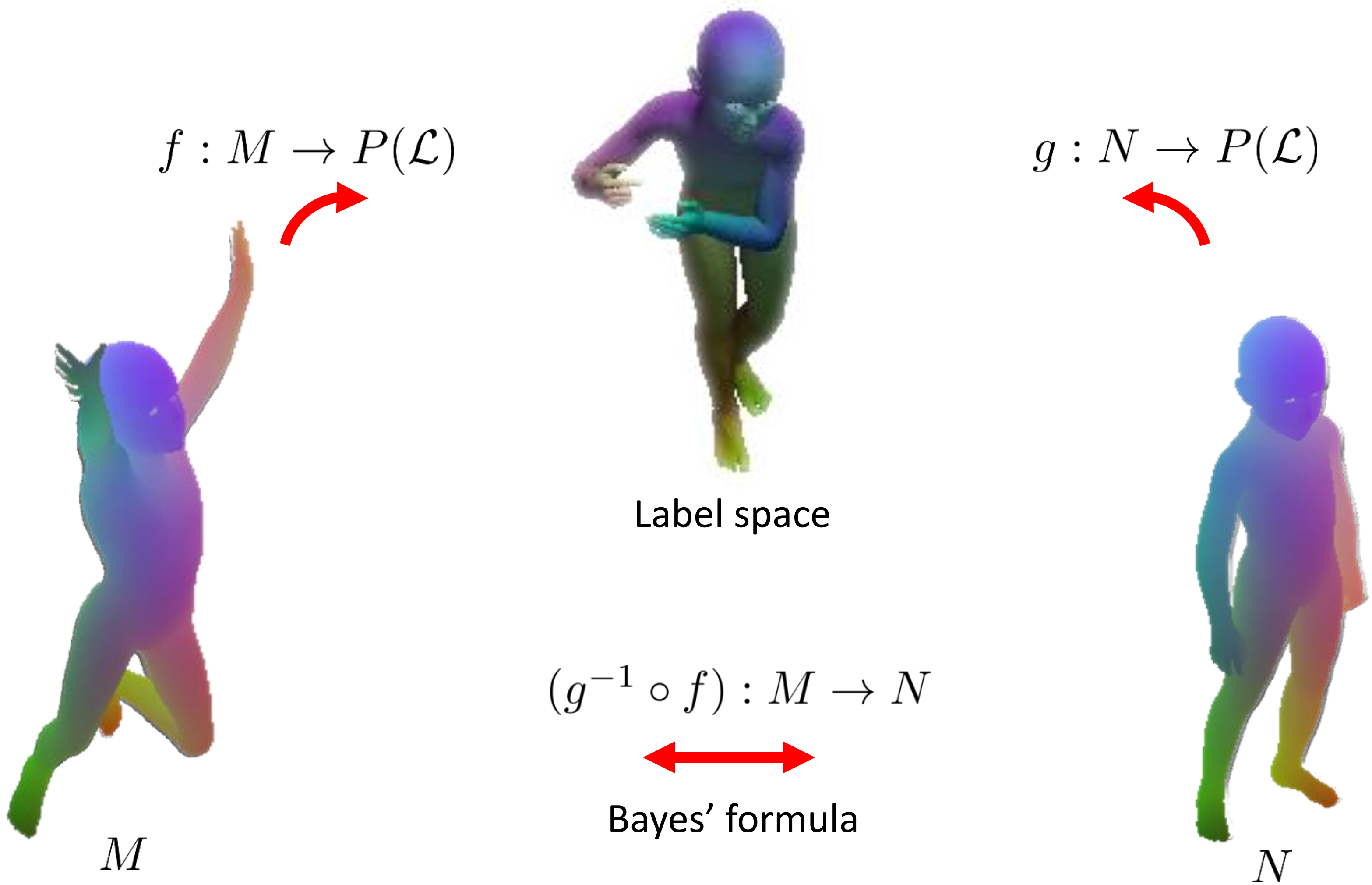
Linear mappings between a label space and probability distributions on shapes

Ways to visualize the label space:

$$\mathcal{L} = \{1, 2, \ldots, L\}$$



Composing the maps

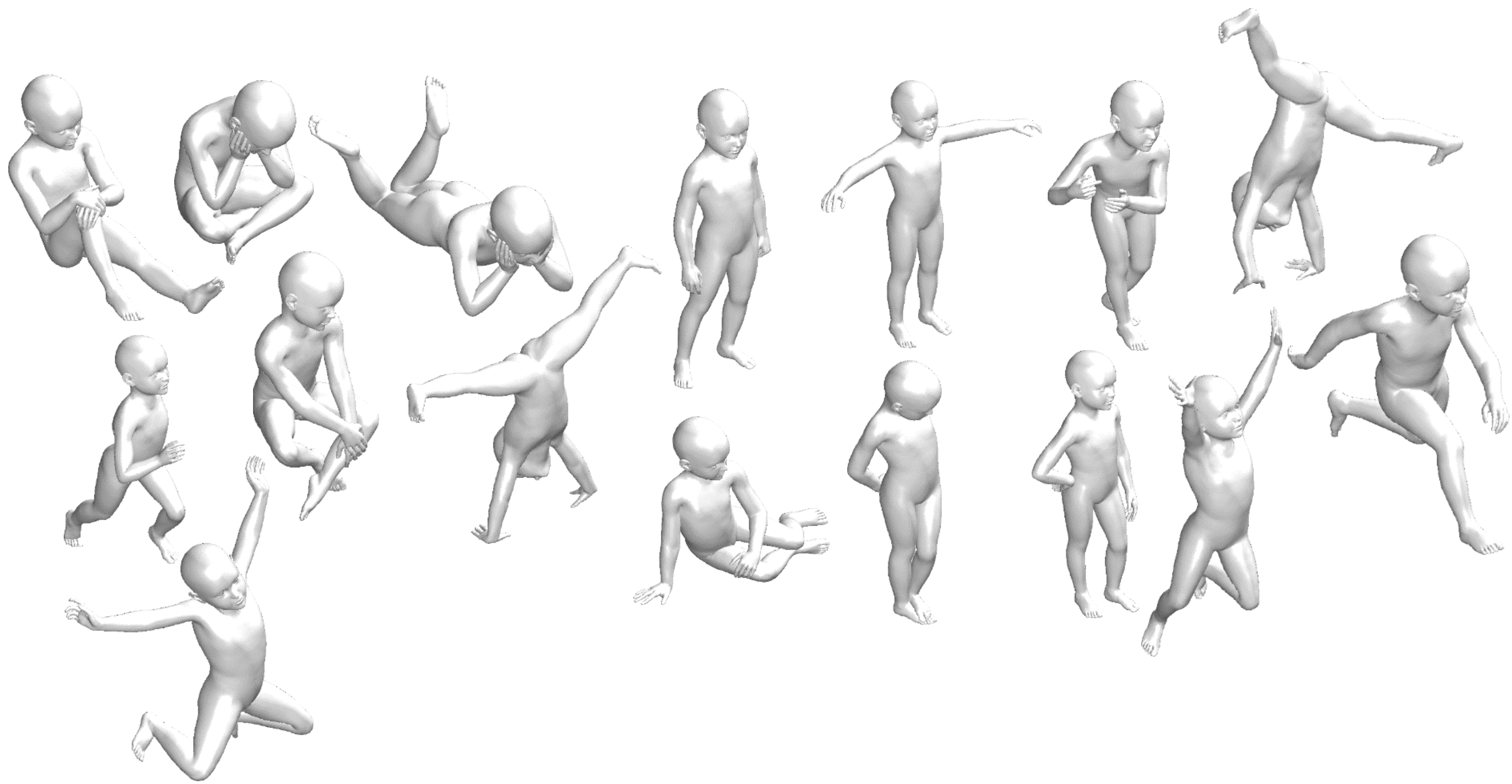


Learning the mapping

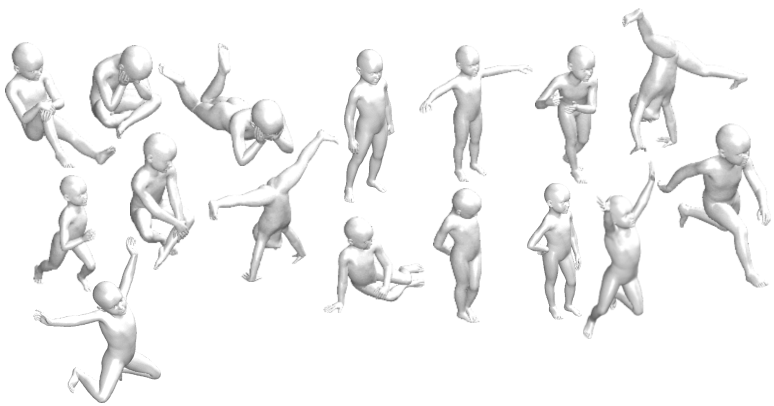
Idea: Consider a learning-by-example approach.

Input: A collection of shapes and ground-truth correspondences between them. Corresponding points have the same label.

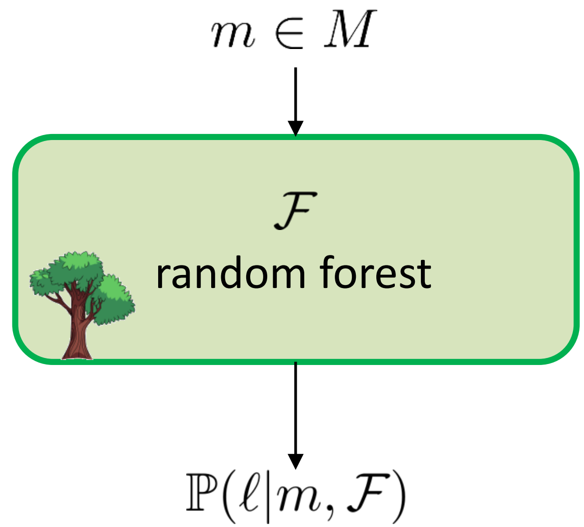
Output: For each test point, a probability distribution over the set of labels.



Random forests: overview



$\rightarrow$

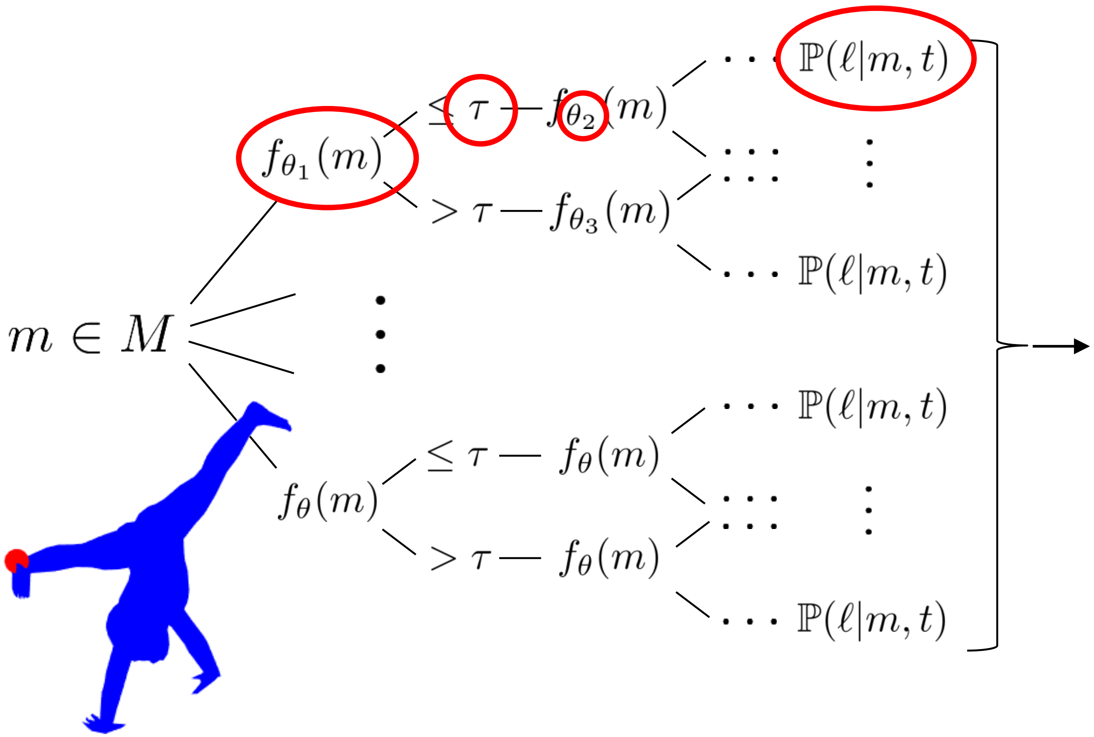


Each point is routed through the forest, and hence matched independently from the others (accounts for partiality).

Breiman 2001

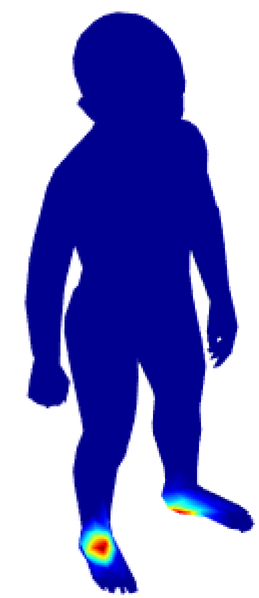
Random forests: inference

Assuming a forest has already been learnt, matching (inference) works as follows:



$$\mathbb{P}(\ell|m,\mathcal{F}) = \frac{1}{|\mathcal{F}|}\sum_{t\in\mathcal{F}}\mathbb{P}(\ell|m,t)$$

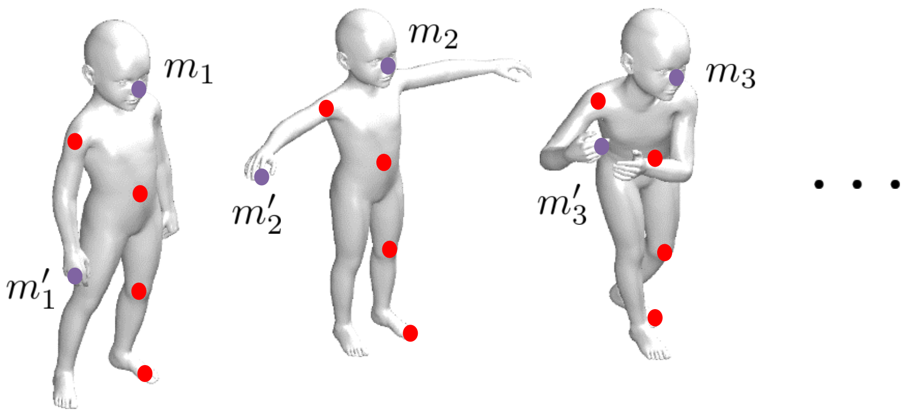
Forest prediction for the given point



Breiman 2001

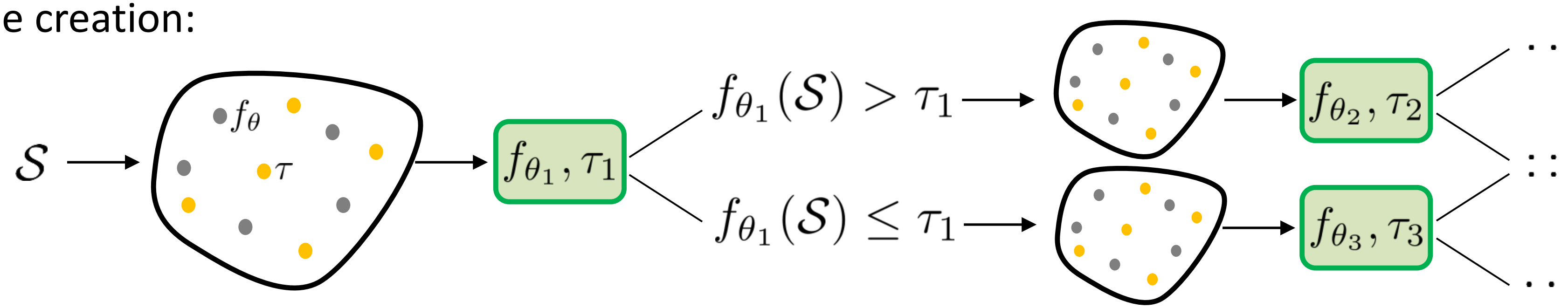
Random forests: learning

The tree structure is determined by defining test (split) functions and randomly generated real-valued thresholds.



$$\mathcal{S} = (m_1, \ell), (m_2, \ell), (m_3, \ell), \ldots \\ \cup \\ (m'_1, \ell'), (m'_2, \ell'), (m'_3, \ell'), \ldots \\ \cup \\ \ldots$$

Node creation:



Pool of randomly generated
split functions and thresholds

Random forests: split functions

Split functions that work well for this problem are classical point descriptors.

For instance, consider the Wave Kernel Signature (WKS):

$$f_\theta(m) = \sum_{k=1}^{\bar{k}} \exp\left[-\frac{(\log e - \log \lambda_k)^2}{2\sigma^2}\right] \phi_k^2(m) \,, \quad \text{where} \quad \theta = \{e, \bar{k}\}$$

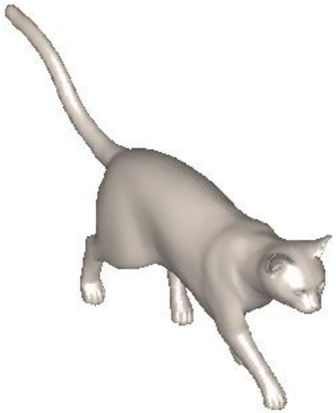
After generating the pool, we keep the split function and the threshold that maximize the expected information gain:

$$\mathrm{IG}(f) = \underbrace{\mathrm{H}(\mathbb{P}(\cdot|\mathcal{S}))}_{\text{before split}} - \underbrace{\mathrm{H}(\mathbb{P}(\cdot|\mathcal{S})|f)}_{\text{after split}}$$

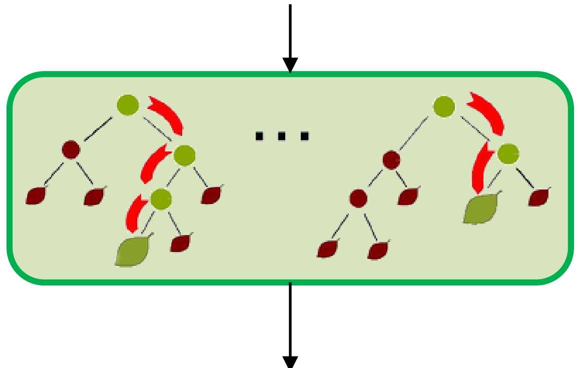
$$\mathbb{P}(\ell|\mathcal{S}) = \frac{|\{(m,\ell)\in\mathcal{S}\}|}{|\mathcal{S}|}$$

Aubry et al. 2011; Rodolà et al. 2014

Forest prediction



$m \in M$



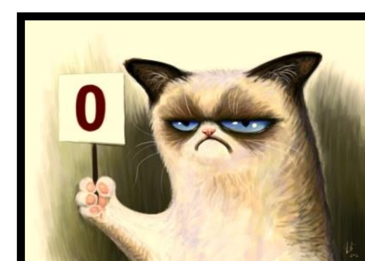
$\mathbb{P}(\ell|m, \mathcal{F})$

Implicit representation

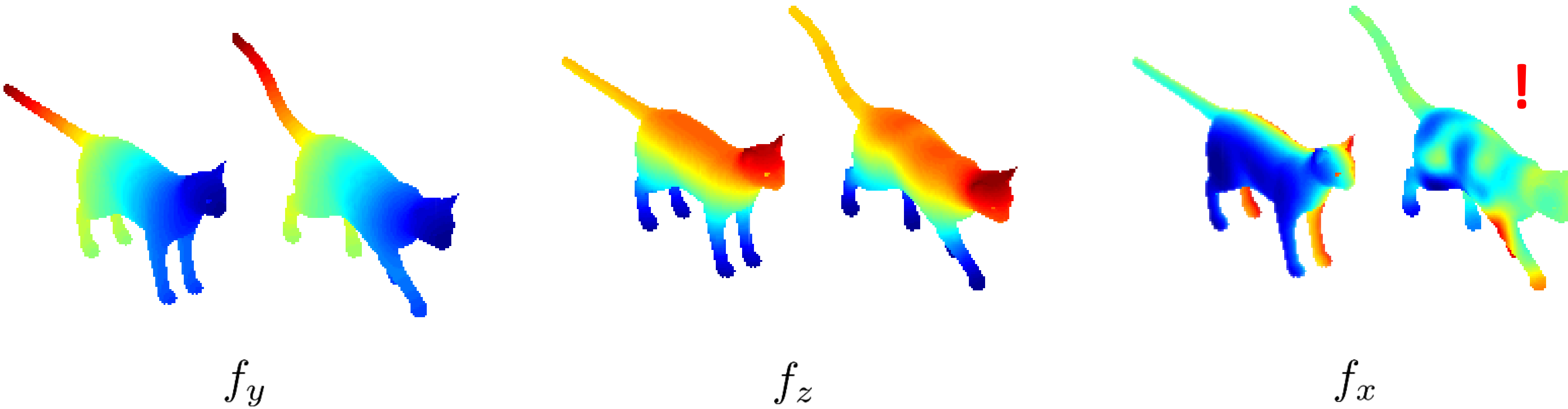
We can represent the forest prediction by the left-stochastic matrix $(X)_{\ell m} = \mathbb{P}(\ell|m, \mathcal{F})$ and take as final correspondence the maximum-likelihood (ML) estimate:

Label space

ML correspondence



Regularization



Ambiguities are generated by the global intrinsic symmetries of the object, which lead to equally good solutions.

Recall that the prediction process does not make full use of the metric structure of the manifold. This can be introduced in the form of a regularizer.

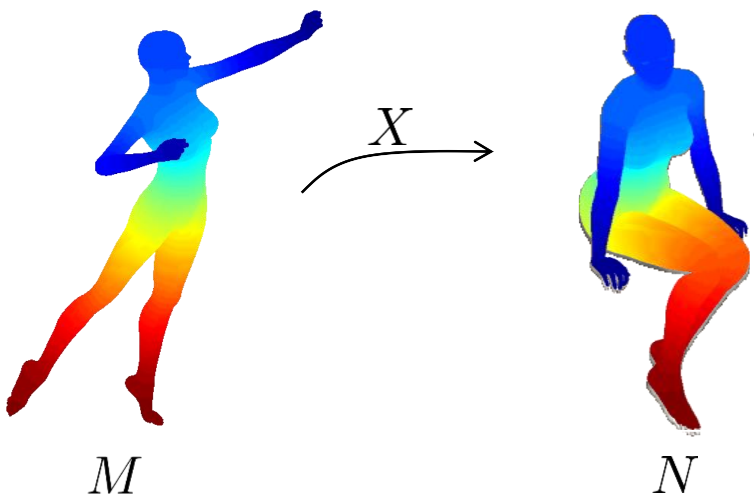
$$E(X) = d(X, X_{\mathcal{F}}) + R(X)$$

Ensures closeness to forest prediction $X_{\mathcal{F}}$

Gives preference to geometrically consistent solutions

Functional maps

We formulate this regularization problem using the language of functional maps.



$$g = f \circ X^{-1} = T_X(f)$$

$T_X : L^2(M) \to L^2(N)$ is linear.

Choice of a basis:

- Indicator (delta) functions on *M* and *N*
- Harmonic bases Φ_M, Φ_N

  Functions are well approximated when truncating the basis.

Ovsjanikov et al. 2012

Functional representation of forest prediction

The random forest gives us a left-stochastic fuzzy correspondence $X_{\mathcal{F}}$, expressed in the standard basis. The associated functional map is obtained by the change of basis:

$$\underbrace{C_{\mathcal{F}}}_{k\times k} = \Phi_N^T \underbrace{X_{\mathcal{F}}}_{n\times n} \Phi_M \qquad\qquad k \ll n$$

The regularization problem becomes:

$$E(C) = d(C, C_{\mathcal{F}}) + R(C)$$

$$\downarrow$$

$$E(C) = \|C - C_{\mathcal{F}}\|_F^2 + R(C)$$

Rodolà et al. 2014

Functional representation of forest prediction

The random forest gives us a left-stochastic fuzzy correspondence $X_{\mathcal{F}}$, expressed in the standard basis. The associated functional map is obtained by the change of basis:

$$\underbrace{C_{\mathcal{F}}}_{k\times k} = \Phi_N^T \underbrace{X_{\mathcal{F}}}_{n\times n} \Phi_M \qquad k \ll n$$

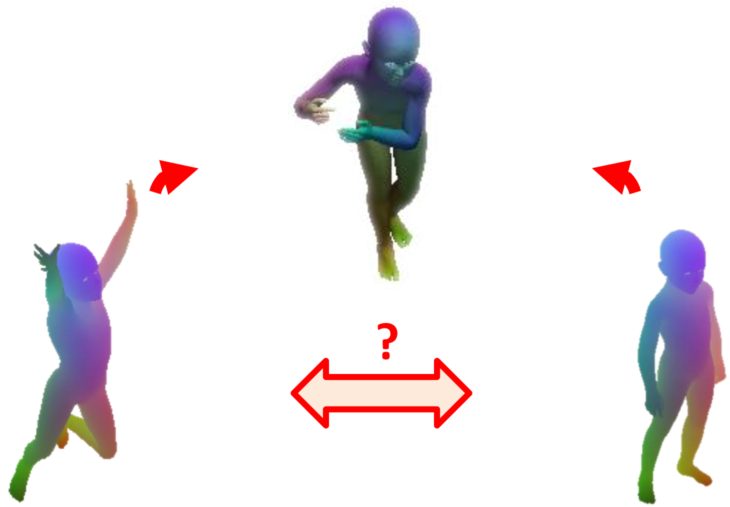
Note: The (truncated) change of basis already has a regularizing effect!

In particular, the projection followed by reconstruction can be seen as a low-pass filtering of the predicted correspondence:

$$R(X_{\mathcal{F}}) = \Phi_N(\Phi_N^T X_{\mathcal{F}} \Phi_M)\Phi_M^T$$

Rodolà et al. 2014

Composing predictions



The matching process gives us two forest predictions defined by:

$$(X_M)_{\ell m} = \mathbb{P}(\ell|m) \quad \text{sparse}$$

$$(X_N)_{\ell n} = \mathbb{P}(\ell|n) \quad \text{sparse}$$

Using the law of total probability, we can compute the fuzzy correspondence:

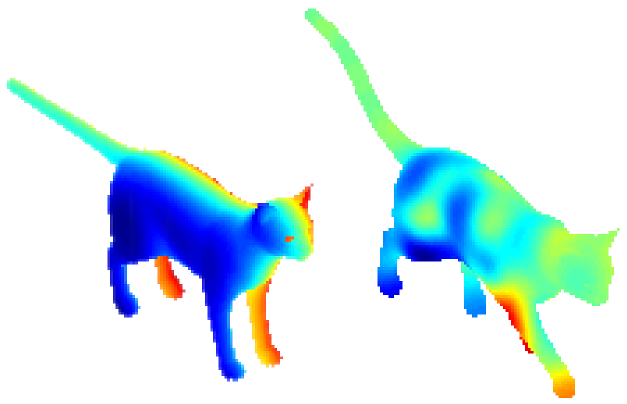
$$(X_{M,N})_{nm} = \mathbb{P}(n|m) = \sum_\ell \mathbb{P}(n|\ell)\mathbb{P}(\ell|m) = (\tilde{X}_N^T X_M)_{nm} \quad \text{big and dense!}$$

We shift again to a functional map representation:

$$X_{M,N} \approx \Phi_N \underbrace{(\Phi_N^T \tilde{X}_N^T)(X_M \Phi_M)}_{C_{M,N}} \Phi_M^T$$

Parentheses are crucial as we avoid computing $\tilde{X}_N^T X_M$

Regularization: commutativity



If the intrinsic symmetry is known, we can impose preservation of the symmetry operator:

$$S_M C = C S_N$$

S_M associates with every function $f : M \to \mathbb{R}$ another function $f \circ S^{-1}$, where $S : M \to M$ is some symmetry on *M*.

In general we cannot assume the symmetry to be known.

In the near-isometric case, however, we can require preservation of the Laplacian:

$$\Delta_M C = C \Delta_N$$

$$\downarrow$$

$$R(C) = \|\Delta_M C - C \Delta_N\|_F^2$$

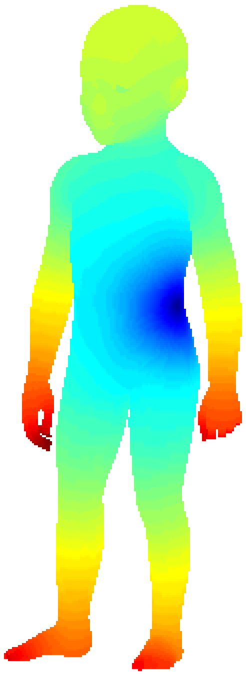
Regularization: sparse matches

Suppose we are given a sparse collection of matches $O \subset M \times N$.

Then for each $(p, q) \in O$ we can define two distance maps:

$d_p(x) = d_M(p, x)$

$d_q(y) = d_N(q, y)$



And thus we can penalize the metric distortion by the regularity term:

$$\|C\hat{d}_p - \hat{d}_q\|^2$$

where $\hat{d}_p = \Phi_M^T d_p$ and $\hat{d}_q = \Phi_N^T d_q$

$$\downarrow$$

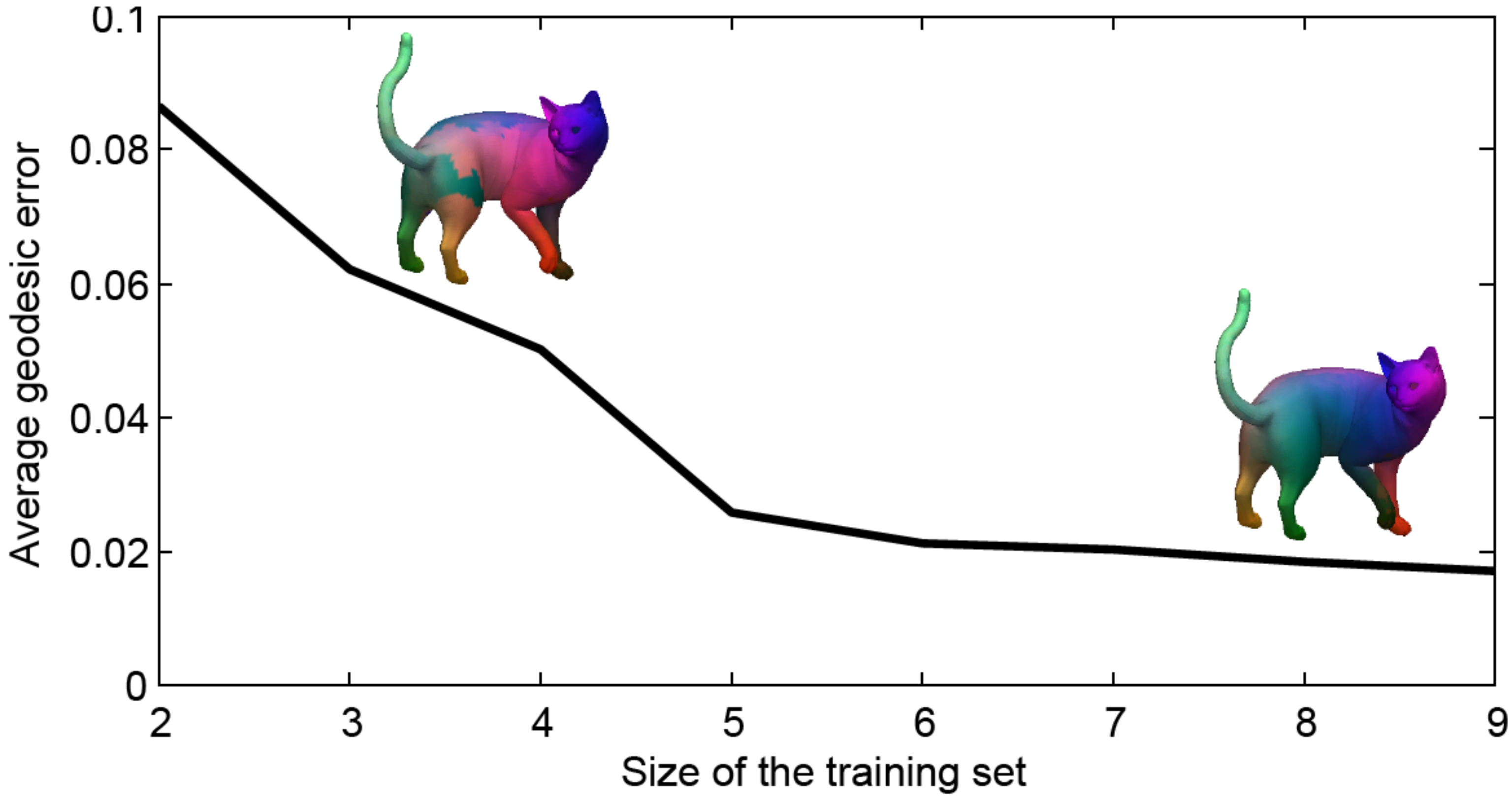
$$R(C) = \sum_{(p,q)\in O} w_{pq}\|C\hat{d}_p - \hat{d}_q\|^2$$

A regular cat

We arrive at the simple least-squares problem:

$$\min_C \underbrace{\|C - C_{\mathcal{F}}\|_F^2}_{\text{closeness to forest prediction}} + \alpha \underbrace{\|\Delta_M C - C\Delta_N\|_F^2}_{\text{preservation of LB operator}} + \beta \underbrace{\sum_{(p,q)\in O} w_{pq}\|C\hat{d}_p - \hat{d}_q\|^2}_{\text{metric distortion}}$$

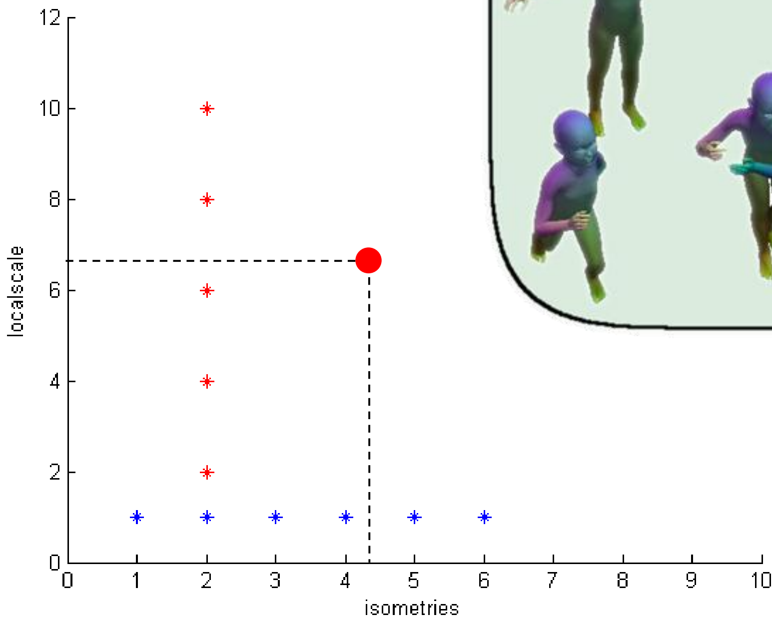
Size of the training set

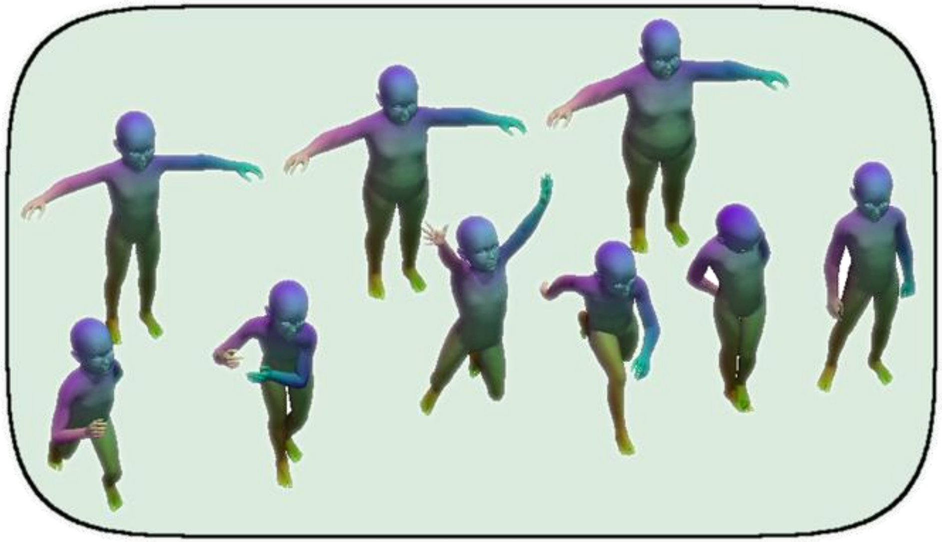


#labels: 10-50K
#shapes: 5-10

We need just few examples (small training sets!). This is because each shape has thousands of vertices with known correspondence.

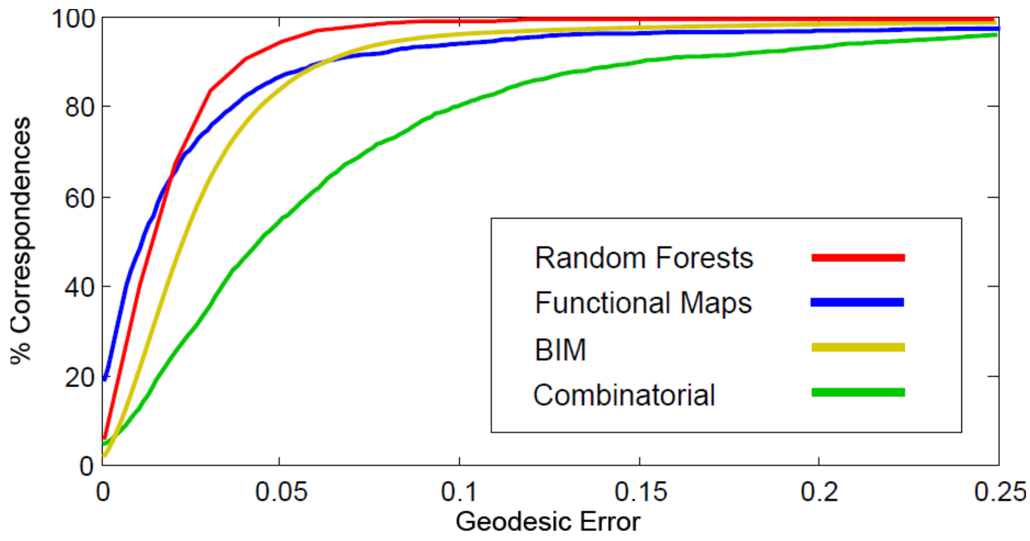
Learning general transformations





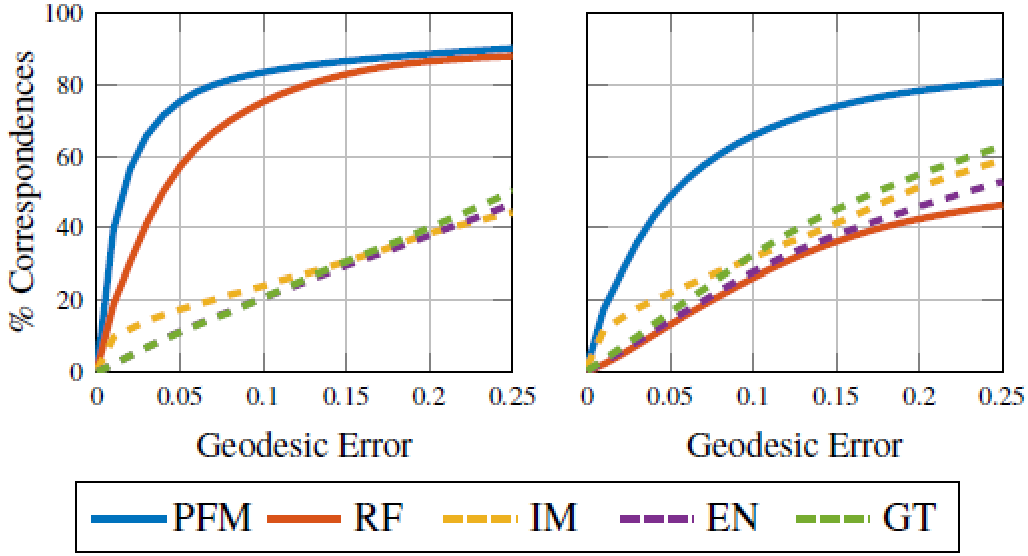
Rodolà et al. 2014

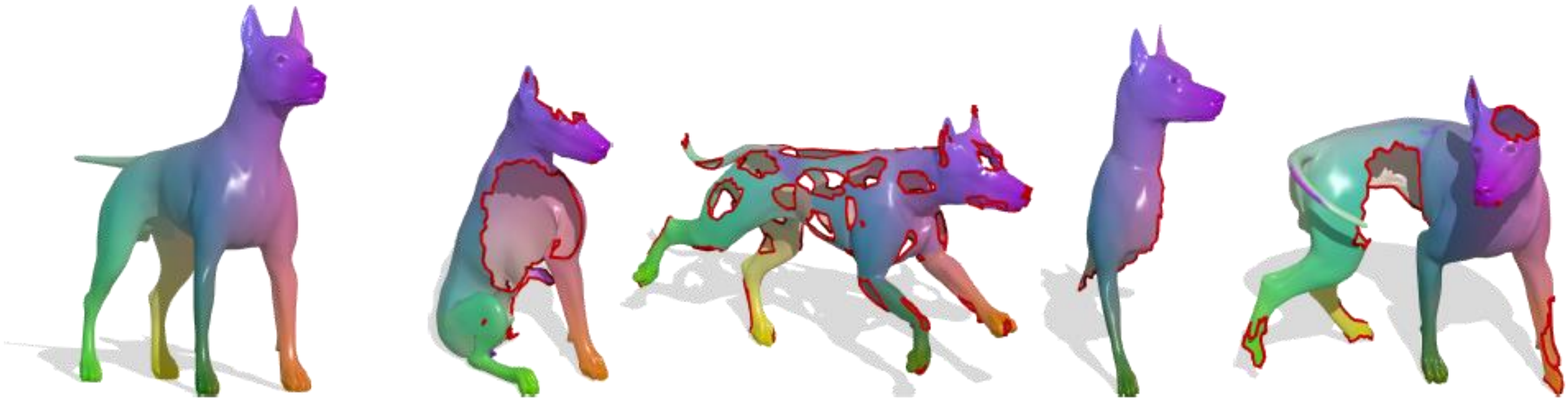
Performance: near-isometric shapes



Rodolà et al. 2014

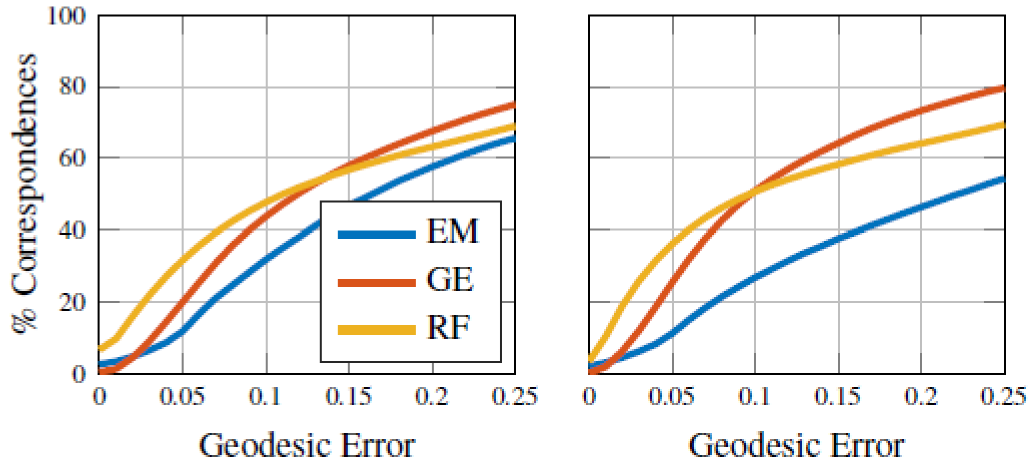
Performance: missing parts (SHREC'16)

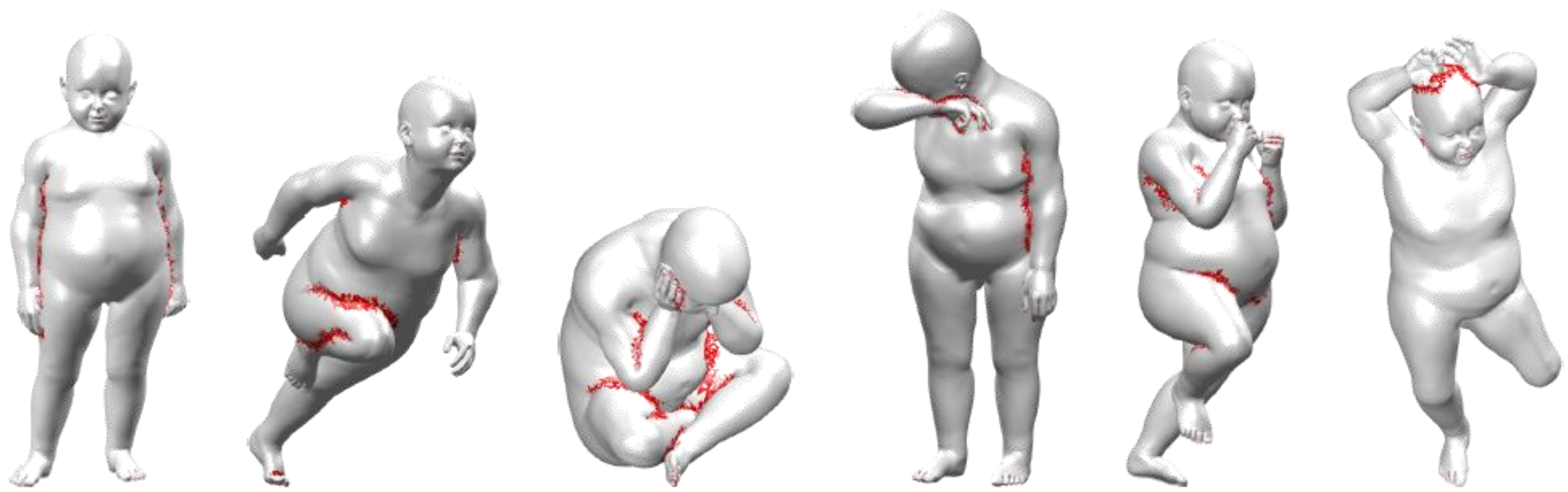




Cosmo, Rodolà, Bronstein, Torsello et al. 2016

Performance: topological noise (SHREC’16)





Lähner, Rodolà, Bronstein, Cremers et al. 2016

Summary

Random forests do a great job at classifying points, and hence work well in correspondence problems. A few extensions one could play with:

- Replace WKS by other descriptors or even mixtures to better capture the variability of deformations
- Introduce structural information to reduce ambiguities (e.g., learn by patches rather than points)
- Learn pairwise rather than pointwise invariants

Some big challenges:

- Ground-truth matches are needed. Difficult to obtain for non-isometric shapes!
- Learn properties of the map, e.g. continuity, orientation, injectivity