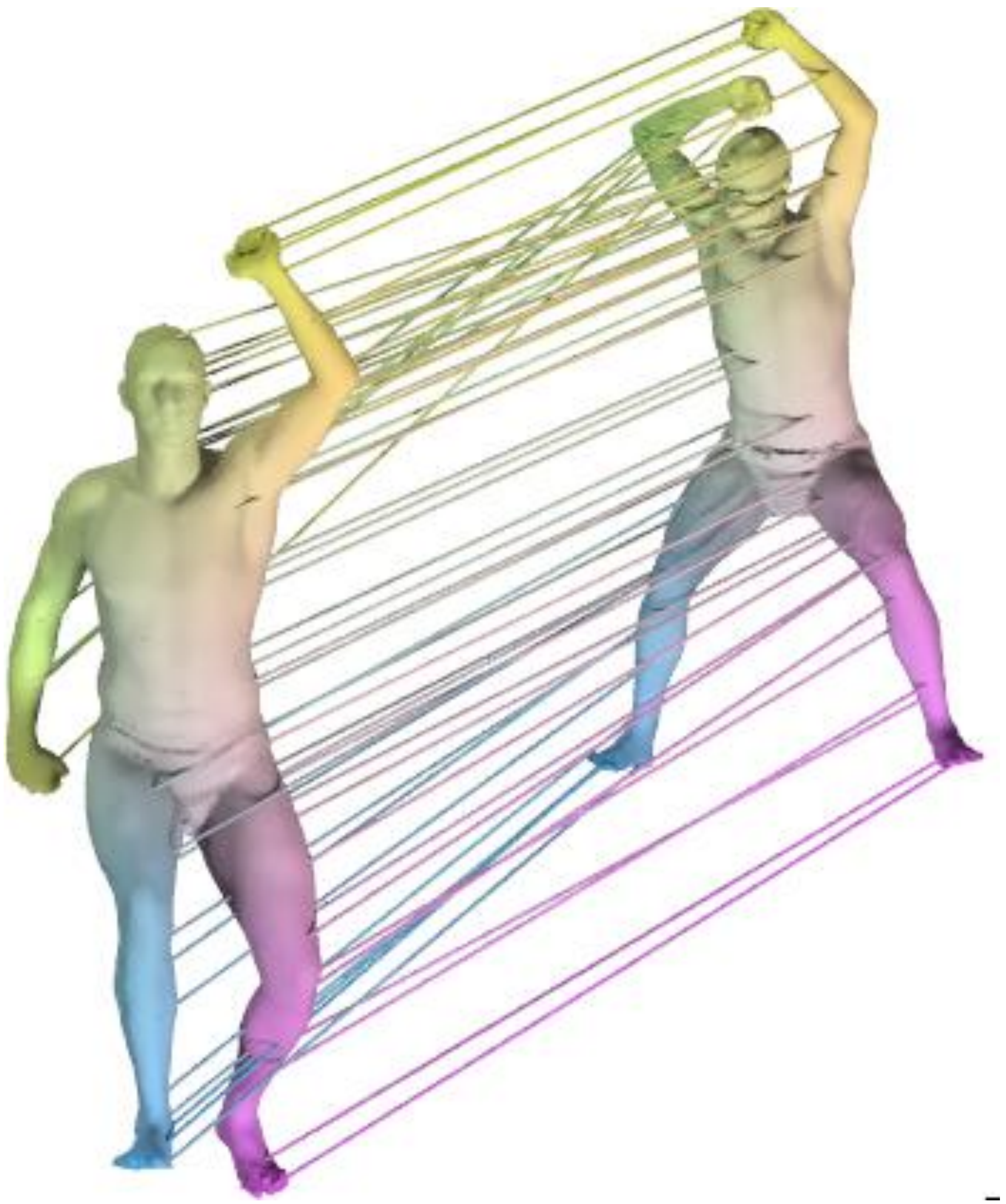
Research Project Proposal: Compact and Adaptive Basis for Shape Correspondence via Functional Maps

Michele Colombo
Michele11.colombo@mail.polimi.it
CSE track

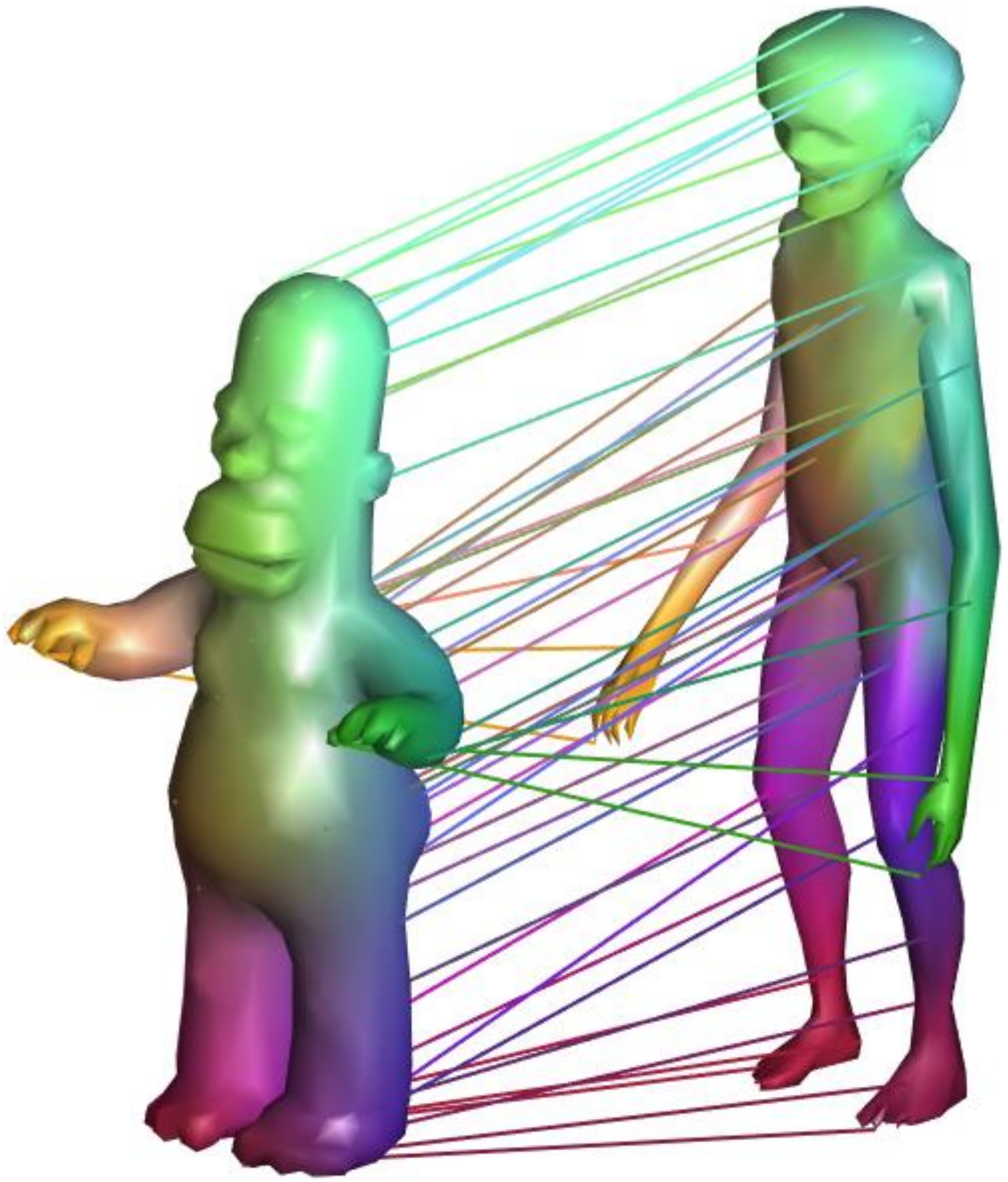
POLITECNICO MILANO 1863

HONOURS PROGRAMME HP-SR in Information Technology

Matching 3D shapes



Source: http://www.lix.polytechnique.fr/~maks/fmaps_course/

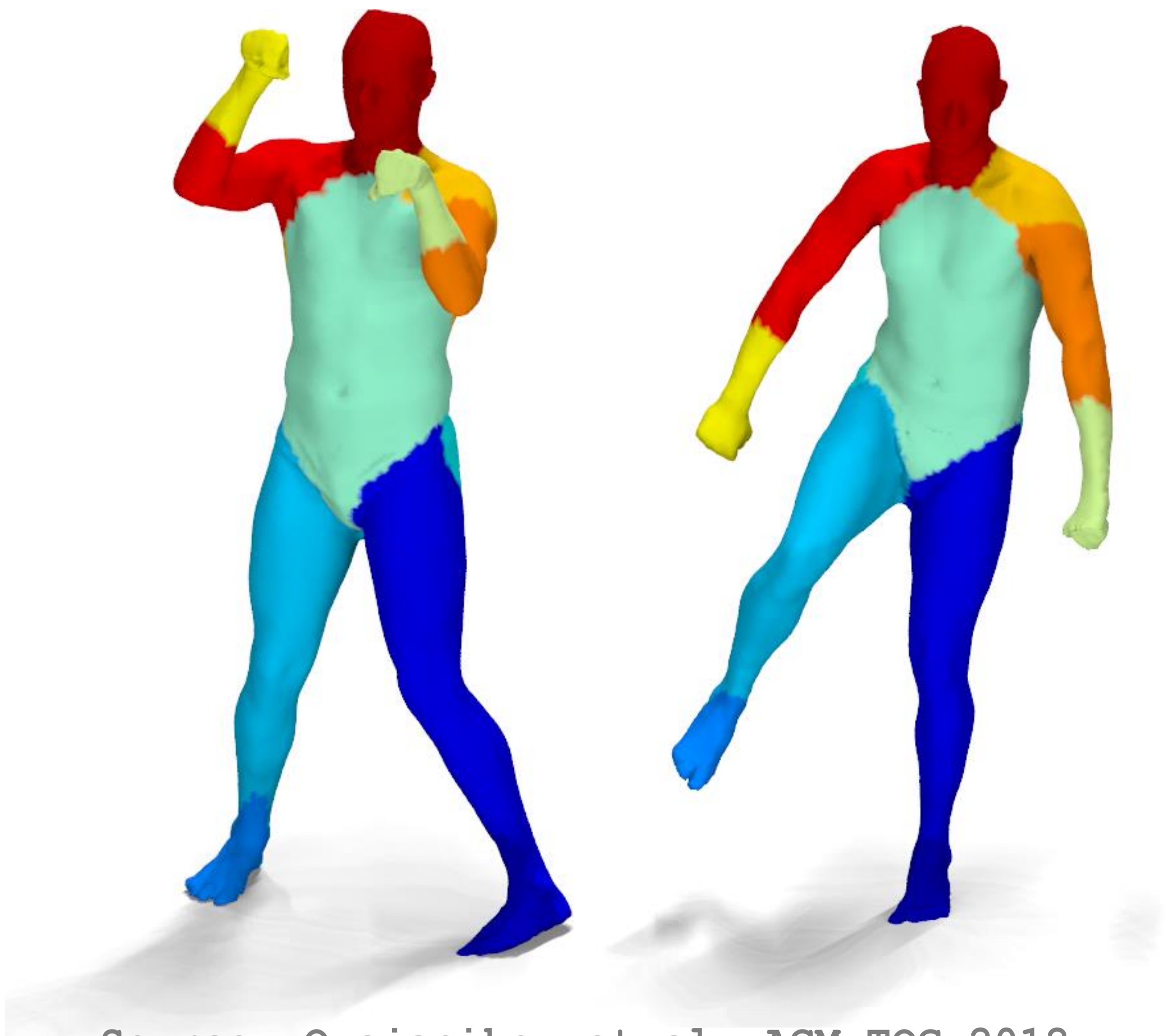


http://www.lix.polytechnique.fr/~maks/publications.html

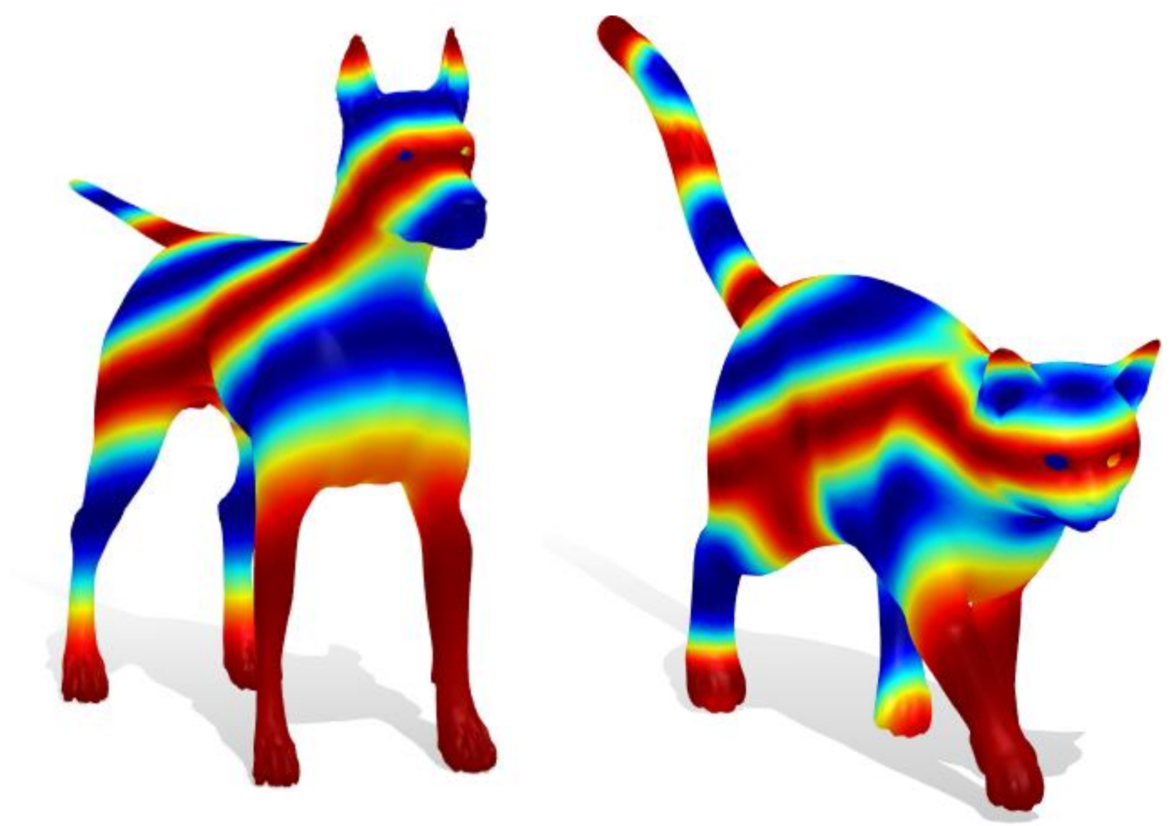
Motivation

- Common step in geometry processing
- Allows for information transfer between shapes:
    - Textures
    - Segmentation
    - Functions

Source: Melzi et al, ACM TOG 2019

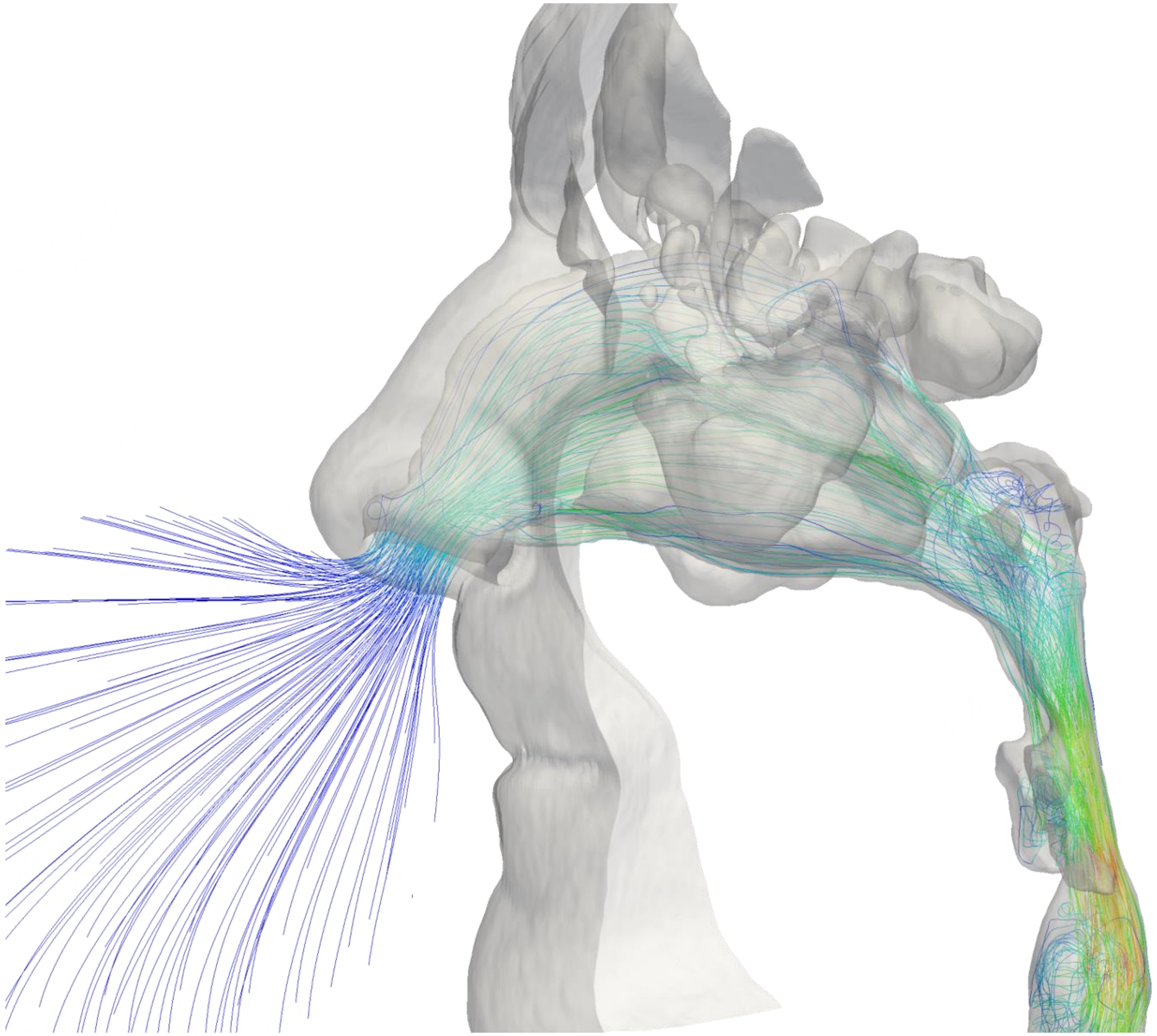


Source: Ovsjanikov et al, ACM TOG 2012



Source: Melzi et al, ACM TOG 2019

Practical application: synthetic dataset creation for medical diagnosis



Source: Schillaci et al, *Inferring functional properties from CFD*

- Goal: automatically diagnose nasal pathologies from CFD simulations on nose scans
- Problem: too few nose scans to train a classifier
- Possible solution: build a synthetic dataset from a collection of healthy nose scans. How?
  - An expert defines pathological deformations on one nose
  - Deformations are transferred on scans of healthy noses
  - Train

This requires non-rigid matching between 3D shapes

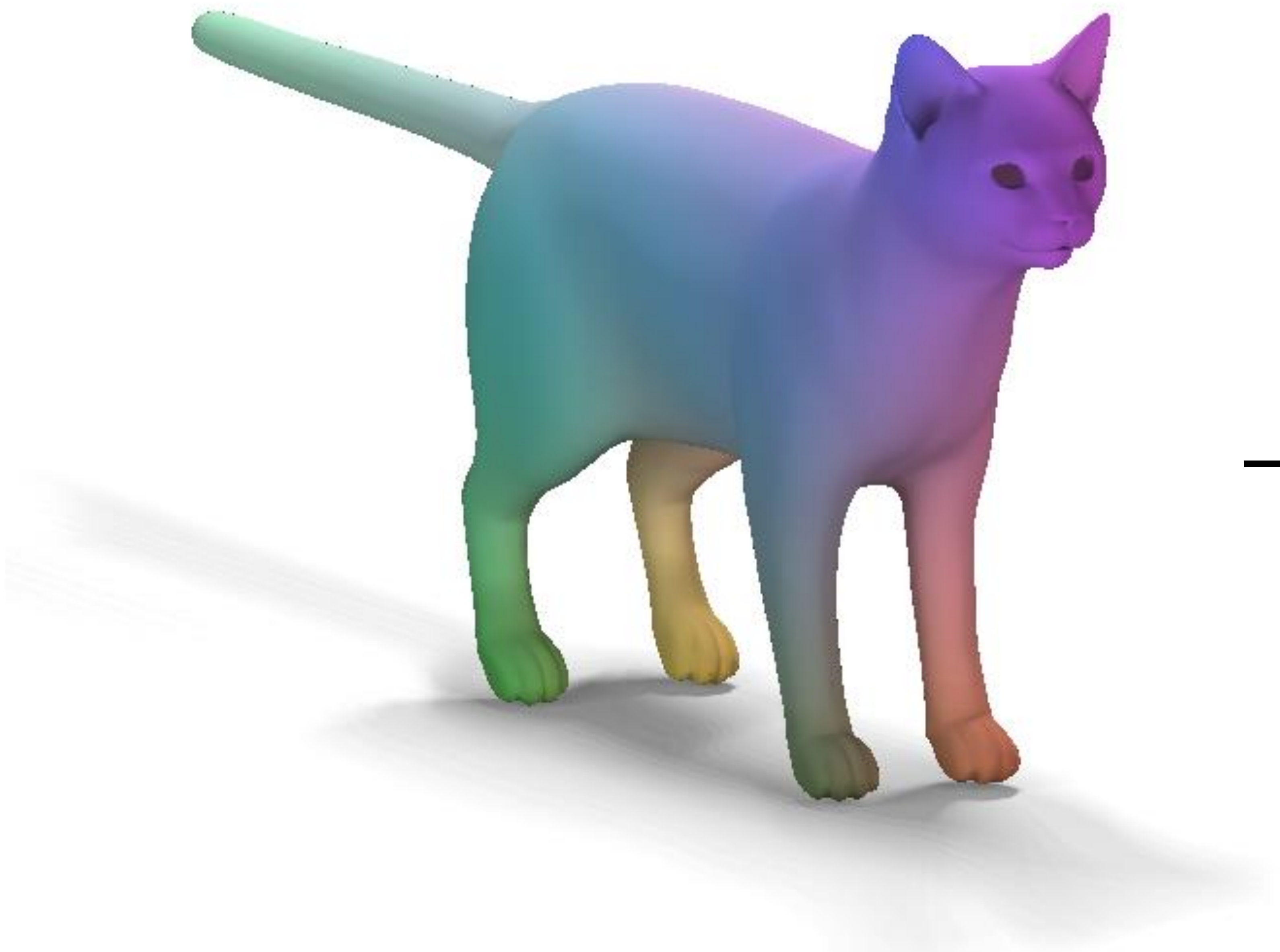
Non-rigid matching: a complex problem

- Point-to-point correspondences live in a huge space, dependent on the size of the shapes
- Rigid matching can be compactly represented as a matrix
- Lack of a compact representation for non-rigid matching

Functional Maps

Matching represented as a correspondence of functions

$\mathcal{M}$



$\mathcal{N}$

Source: Ovsjanikov et al, ACM TOG 2012

$\boldsymbol{T}: \mathcal{M} \rightarrow \mathcal{N}$

$\boldsymbol{f}: \mathcal{M} \rightarrow \mathbb{R}$

$\boldsymbol{T_F}: \mathcal{F}(\mathcal{M}, \mathbb{R}) \rightarrow \mathcal{F}(\mathcal{N}, \mathbb{R})$

$\boldsymbol{g}: \mathcal{N} \rightarrow \mathbb{R}$

$\boldsymbol{g} = \boldsymbol{f} \circ \boldsymbol{T}^{-1}$

$\boldsymbol{g} = \boldsymbol{T_F}(\boldsymbol{f})$

Functional Maps

- T can be recovered from T_F
- Not all functional maps have a correspondent point-to-point map

Functional maps are strictly more expressive than point-to-point maps

Functional Maps

- Given a basis $\{\phi_i^{\mathcal{M}}\}$ of real-valued functions defined on $\mathcal{M}$:

$$f = \sum_i a_i \phi_i^{\mathcal{M}}$$

- T_F is linear

- The transformation becomes:

$$T_F(f) = T_F\left(\sum_i a_i \phi_i^{\mathcal{M}}\right) = \sum_i a_i T_F(\phi_i^{\mathcal{M}})$$

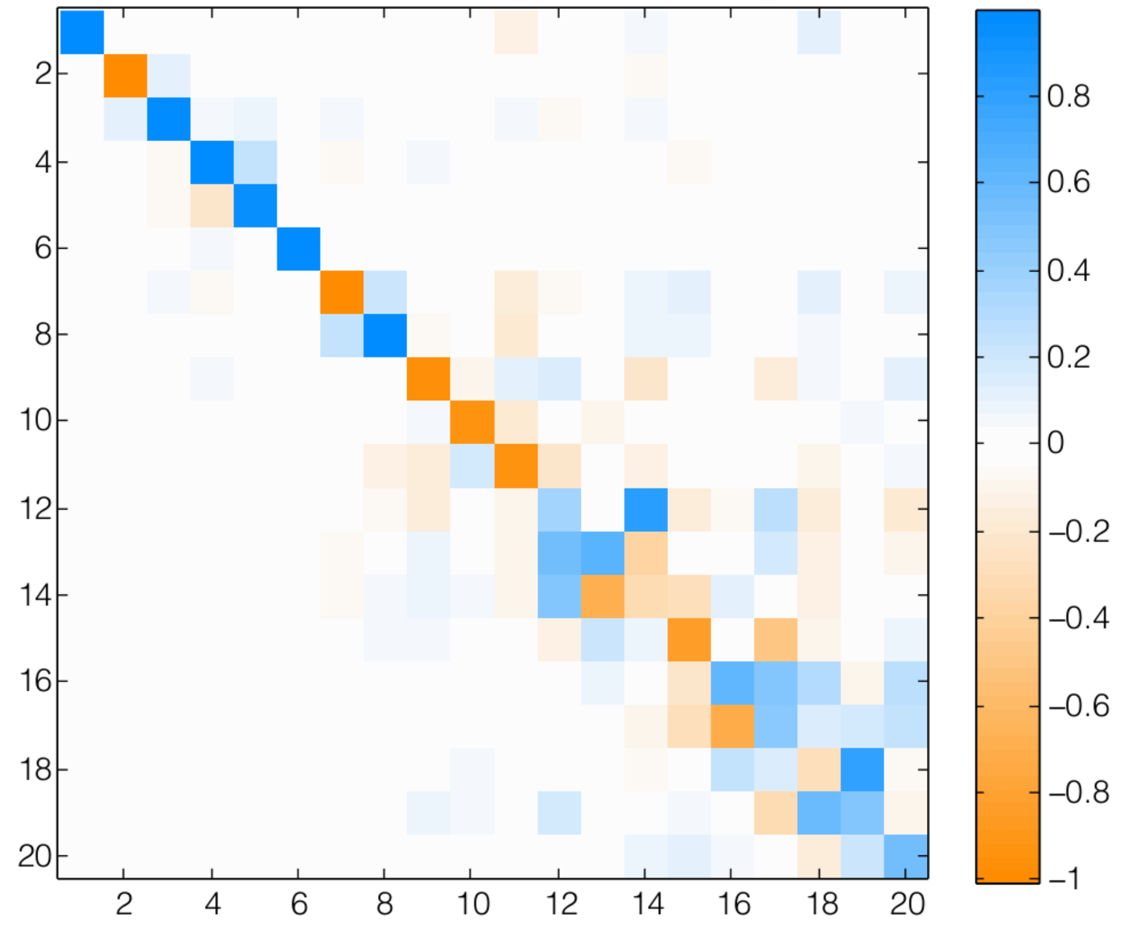
Functional Maps

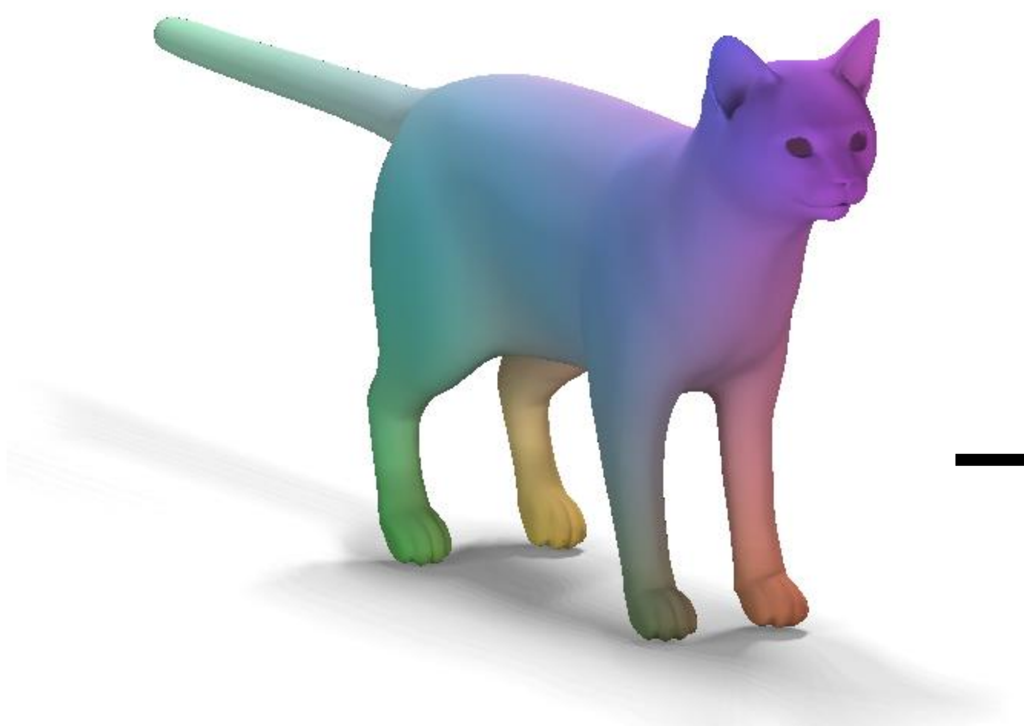
- Let $\{\phi_j^{\mathcal{N}}\}$ be a basis of real-valued functions defined on $\mathcal{N}$
- $T_F(\phi_i^{\mathcal{M}}) = \sum_j c_{ji}\phi_j^{\mathcal{N}}$ for some c_{ji}
- $T_F(f) = \sum_i a_i \sum_j c_{ji}\phi_j^{\mathcal{N}} = \sum_j \sum_i a_i c_{ji}\phi_j^{\mathcal{N}}$

Matrix representation

- c_{ji} are independent of f, depend only on the basis
- c_{ji} completely define the mapping between $\phi_i^{\mathcal{M}}$ and $\phi_j^{\mathcal{N}}$

Given the two bases, the matrix C fully represents the functional mapping





$$\boldsymbol{f} = \sum_i \boldsymbol{a}_i \boldsymbol{\phi}_i^{\mathcal{M}}$$

$$\boldsymbol{b} = \boldsymbol{T_F}(\boldsymbol{a}) = \boldsymbol{C} \cdot \boldsymbol{a}$$

$$\mathbf{g} = \sum_j \boldsymbol{b}_j \boldsymbol{\phi}_j^{\mathcal{N}}$$

Finding C

- Solve and optimization problem to compute C
- The objective function is given by the minimization of the error on function preservation:

$$\min \left\| C\,\hat{f} - \hat{g} \right\|^2$$

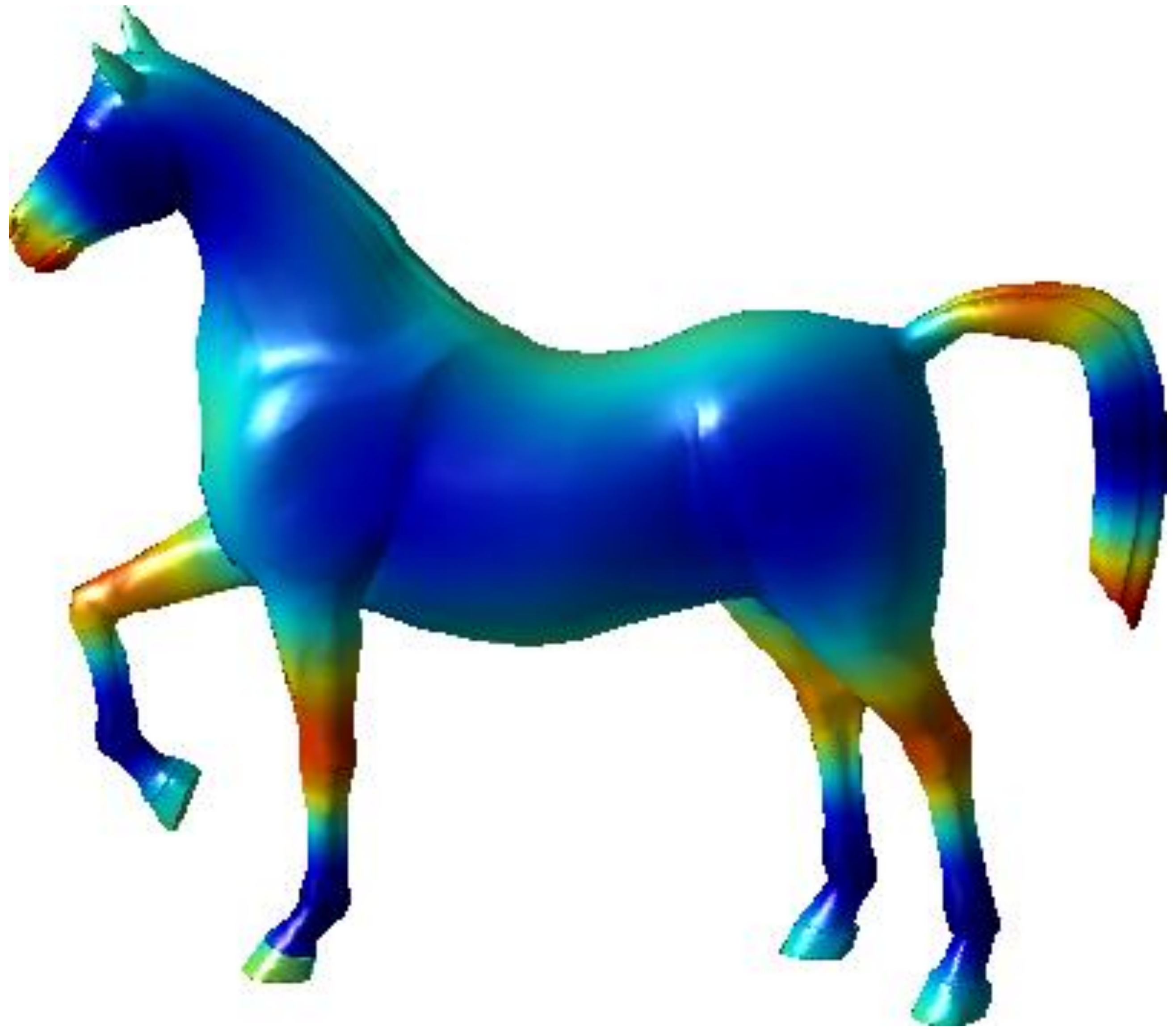
- Different kinds of functions can be used:
  - Descriptors
  - Landmarks
  - Segments

This leads to a linear problem in C

Descriptors

- Probe functions that should characterize any point of the shape as precisely as possible
- Intrinsic descriptors are independent of isometric transformations
- Examples:
  - HKS
  - WKS
  - Learn-based

Source: Sun et al, ESGP 2009

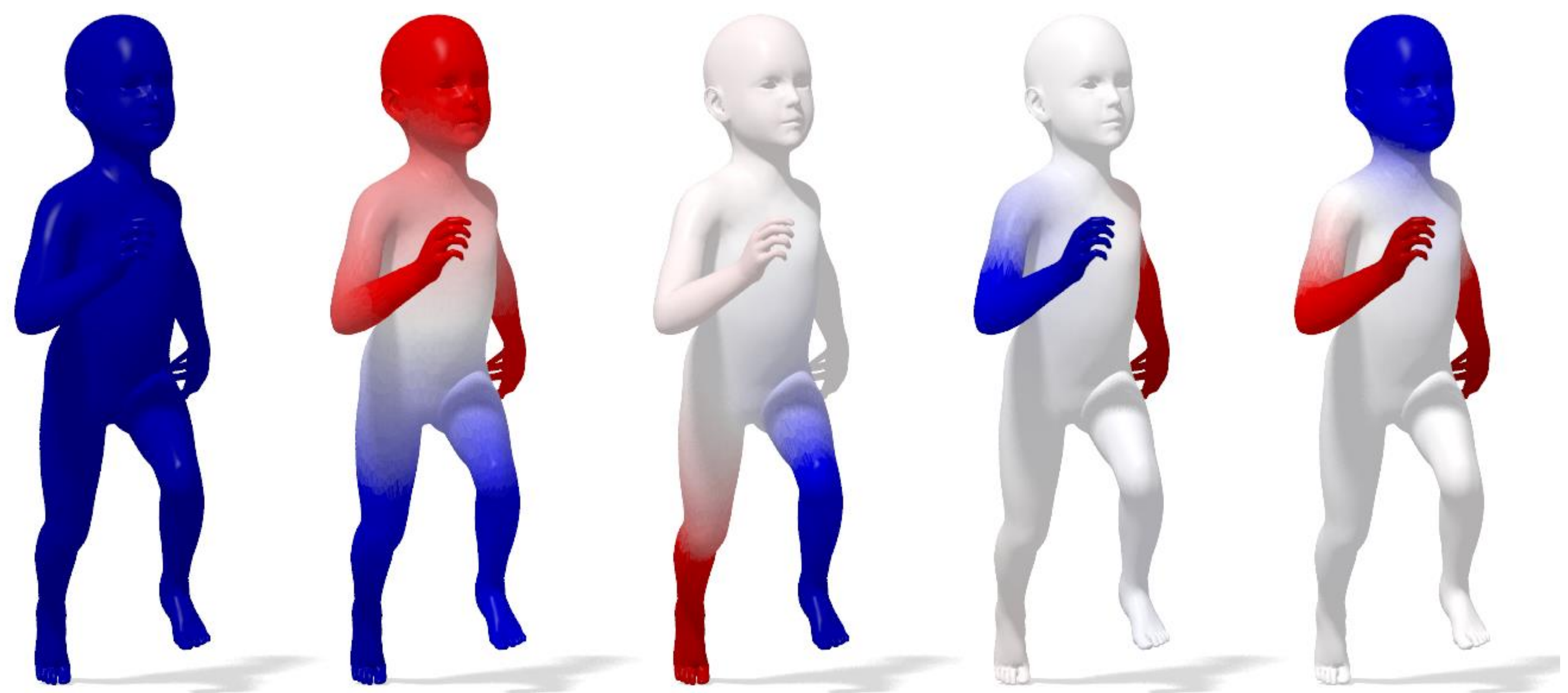


Source: Aubry et al, ICCV 2011

Choice of the Basis

- Crucial aspect of functional maps
- Desired properties:
  - **Compactness**: most natural functions should be well approximated with a small number of basis elements
  - **Stability**: the space of functions spanned should be stable under small or near-isometric shape deformations

LB eigenbasis

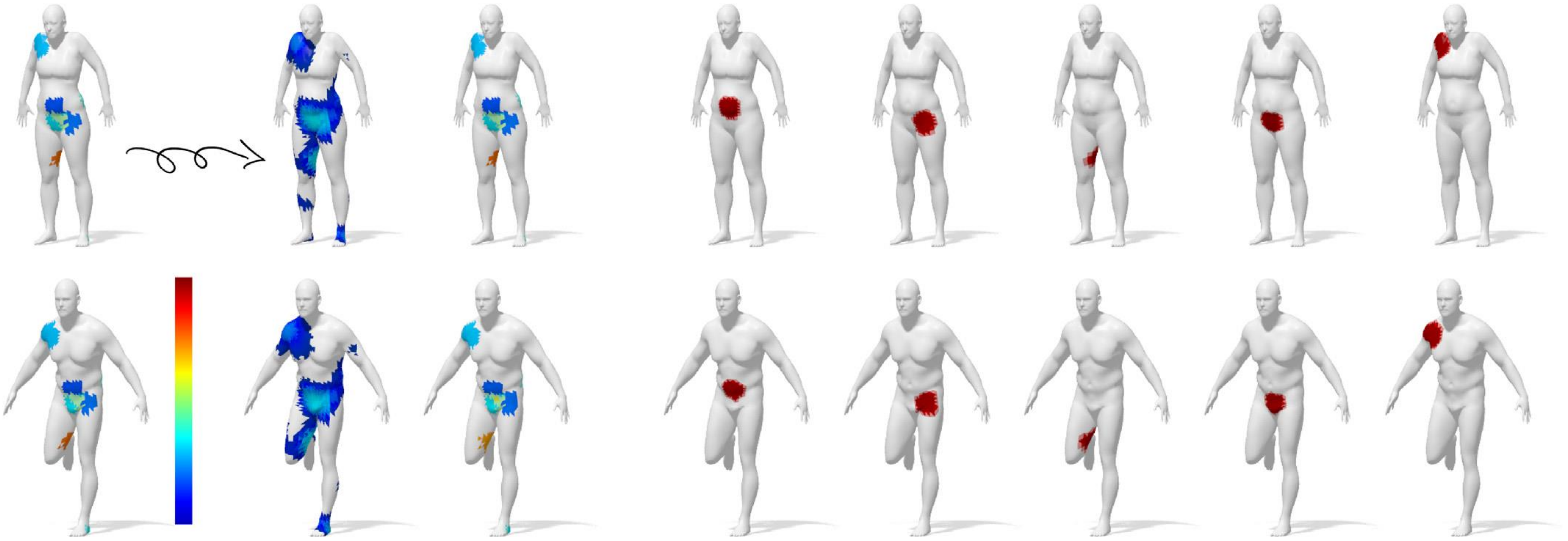


Source: Melzi et al, Localized Manifold Harmonics, 2017

- Eigenfunctions of the Laplace-Beltrami operator
- Manifold equivalent of Fourier basis
- **Ordered** in increasing frequency
- Selecting the first k elements correspond to a **low pass filter** approximation
- Proved to be **optimal** for smooth (bounded variation) functions [Aflalo et al., 2015]
- Problems:
  - Instable at higher frequencies -> not suitable for detailed functions
  - Not well behaved for non smooth functions (e.g. indicators)

Binary Sparse Frame

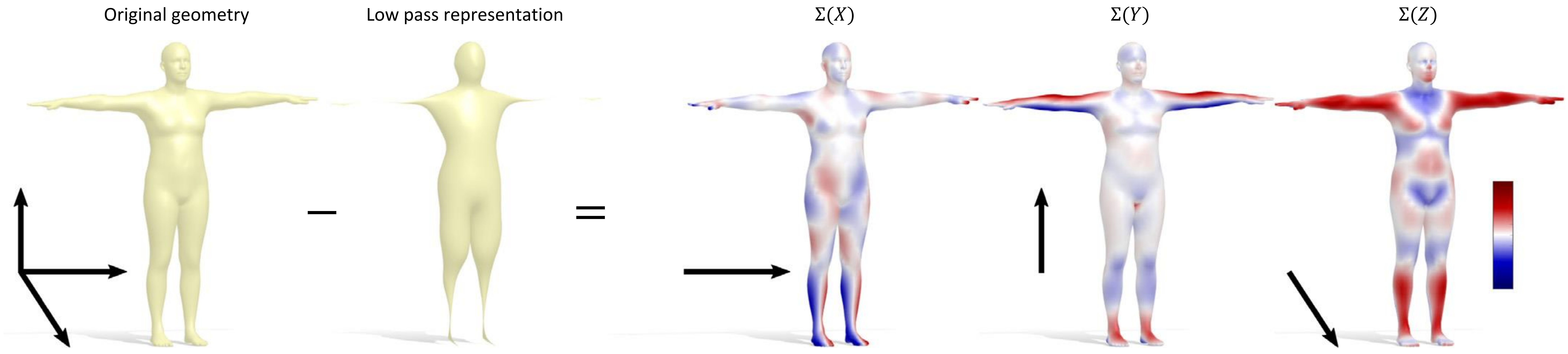
Optimized for representation and transfer of step functions



[Melzi, Computer & Graphics, 2018]

Coordinate Manifold Harmonics

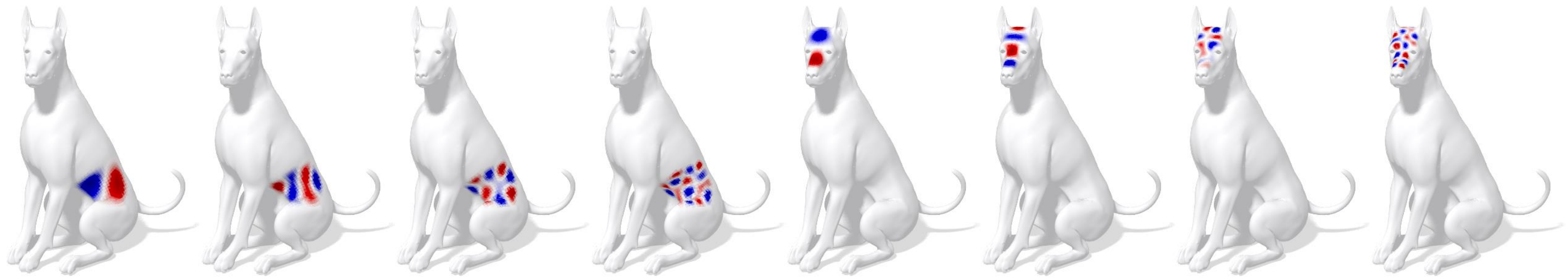
- Integrates LB eigenbasis with information about the spatial coordinates (extrinsic information)
- Designed for mesh structure transferring between shapes
- Requires similar pose



[Melzi et al., Computer & Graphics, 2019]

Localized Manifold Harmonics

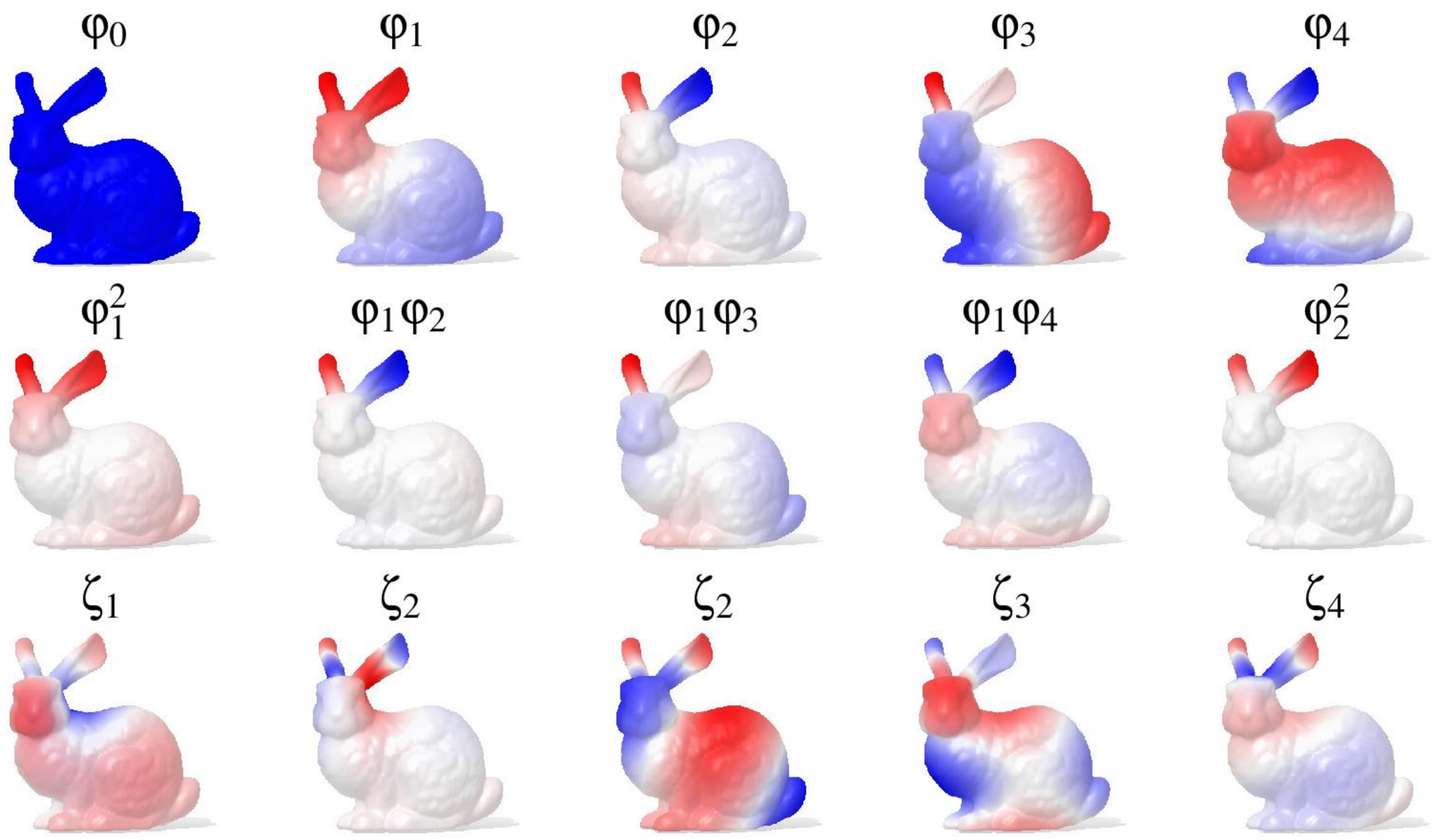
- Integrates LB eigenfunctions with localized functions
- Still ordered in frequency, naturally extends LB eigenbasis
- Requires the definition of regions



[Melzi et al., Computer & Graphics, 2017]

Pointwise products of LB eigenfunctions

- A functional map preserves pointwise product of functions if and only if it corresponds to a point-to-point map
- Products contain information in higher frequencies
- Their mapping is given by the alignment of LB eigenfunctions only, but it is sensitive to noise



[Nogneng et al., Eurographics, 2018; Maggioli et al., Eurographics, 2021]

Research directions

- **Goal:** Find a compact and adaptive basis for the representation of real functions on manifolds.
- Able to represent and transfer detailed functions
- Two step process:
  - **Expand**: create an overcomplete dictionary
  - **Select**: reduce its dimension

Expansion

- **Goal**: create a large set of generator to cover a large functional space
- Redundancy is good to leave room for a good selection after
- Freedom to use different functions defined on the shape. Some options:
  - LB eigenfunctions
  - Descriptors
  - Binary sparse frame
  - Pointwise products of these functions

Selection

- Assumption: not all the spanned space is equally interesting
- **Goal**: select the most informative part of it
- **Adaption** to specific classes of functions
- Options:
  - Principal Component Analysis
  - Double sparsity

Principal Component Analysis

- Select an orthogonal set of generators of a subspace
- Given a set of functions, the subspace is the one that best approximates them
- Simple to apply
- Fast computation of representation, thanks to orthonormality

Double sparsity

- Learn a dictionary over a base dictionary:

$$D = \Phi A$$

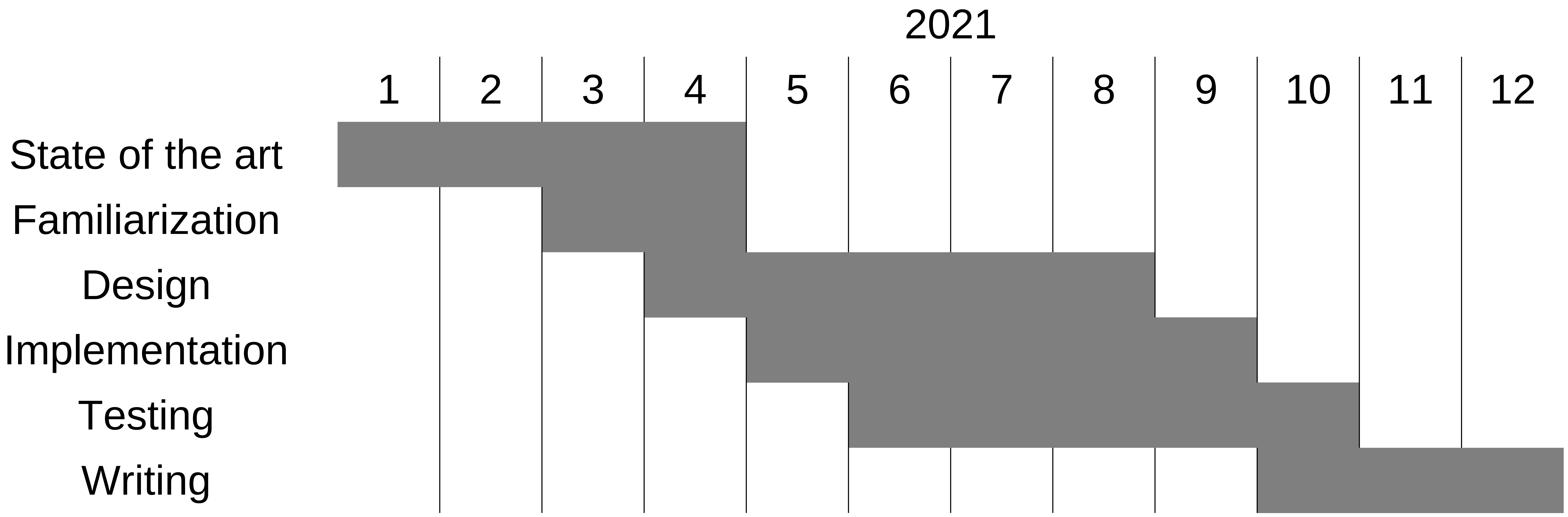
  - D is the new dictionary
  - Φ is the fixed base dictionary
  - A is sparse and it is learned from a set of samples X
- Sparsity over a predefined set of atoms acts as **regularization**
- In our case: Φ is the expanded basis, X is a set of informative functions defined on the manifold. The goal is to learn A.

[Rubinstein et al., IEEE Transactions on Signal Processing, 2010]

Evaluation

- Experimental evaluation on widely used shape datasets: FAUST [Bogo et al., CVPR 2014], TOSCA [Bronstein et al., Springer 2008] and SCAPE [Anguelov et al., NIPS 2005]
- Comparable to current state-of-the-art methods
- Two metrics:
  - Normalized approximation error for function representation and transfer
  - Geodesic error for shape matching

Planning


2021
1
2
3
4
5
6
7
8
9
10
11
12
State of the art
Familiarization
Design
Implementation
Testing
Writing