Advanced Deep Learning Models and Methods for 3D Spatial Data

Giacomo Boracchi, Luca Magri, Simone Melzi, Matteo Matteucci

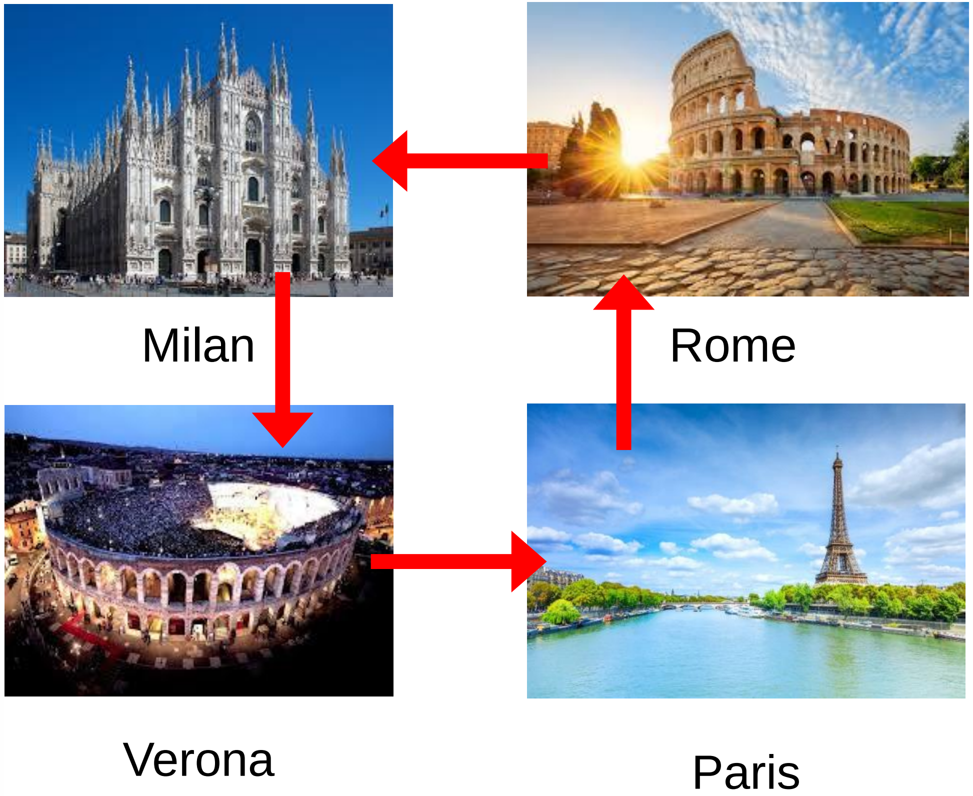
DEIB, Politecnico di Milano January 31, 2024

simone.melzi@unimib.it

https://sites.google.com/site/melzismn

Simone Melzi

https://sites.google.com/sitea/melzismn



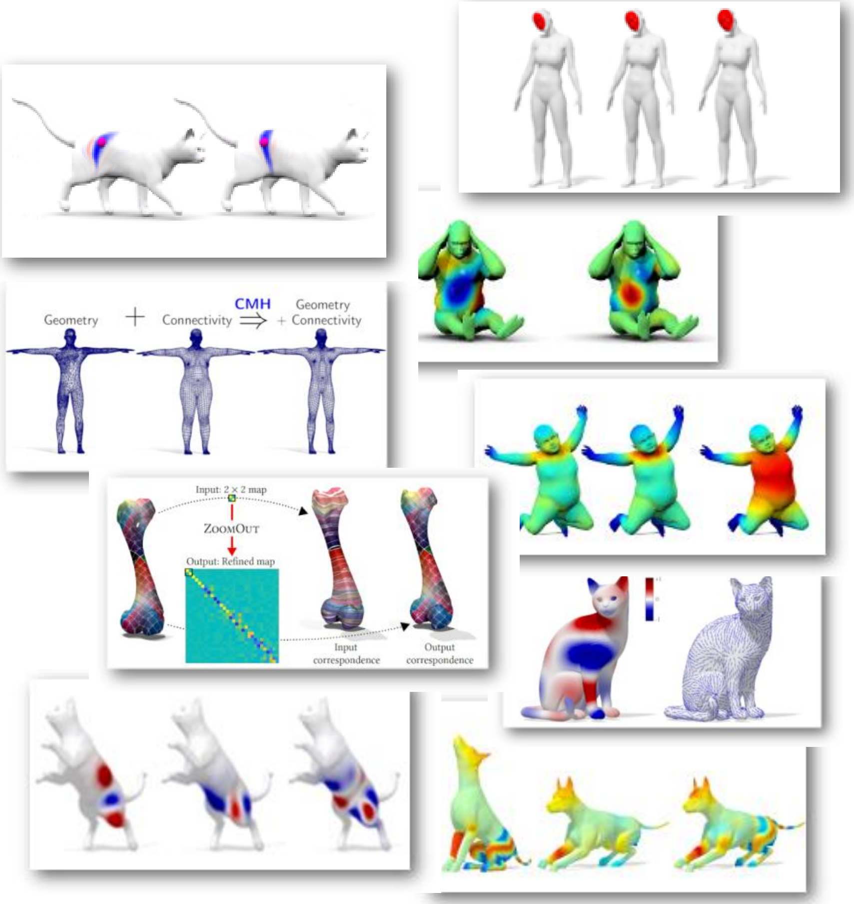
UNIVERSITA' DEGLI STUDI DI MILANO BICOCCA
DIPARTIMENTO DI INFORMATICA, SISTEMISTICA E COMUNICAZIONE
Viale Sarca, 336,20126 Milano

My research

- Geometry processing
- Spectral shape analysis
- Machine learning
- Geometric deep learning

Math and AI for Geometric data

https://sites.google.com/site/melzismn



Deep Learning in 3D Non-rigid Shape Registration

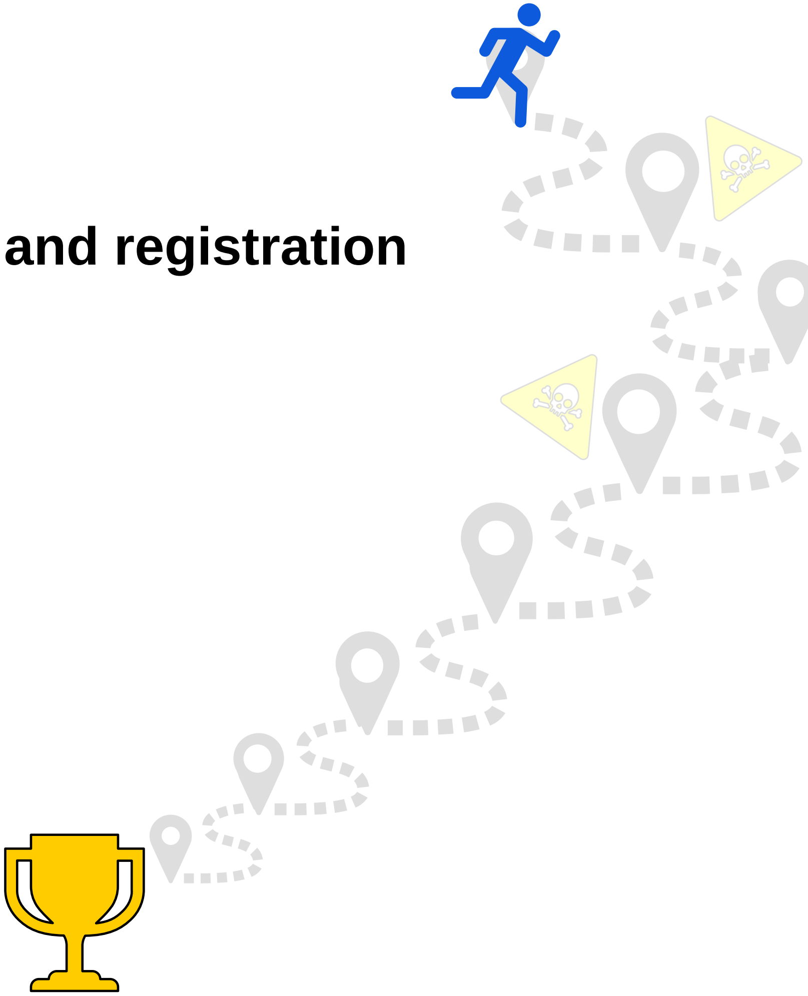
Slides credits to Maks Ovsjanikov, Emanuele Rodolà, Riccardo Marin, Jing Ren, Giovanni Trappolini, Michael Bronstein and Thibault Groueix

Today's route:

1. Introduction: 3D Non-Rigid shapes and registration
2. Spectral representation
3. Axiomatic approaches
4. Functional maps
5. Learning on geometric data
6. Learning–based Functional maps
7. Other learning-based approaches
8. Transformers

INTRODUCTION

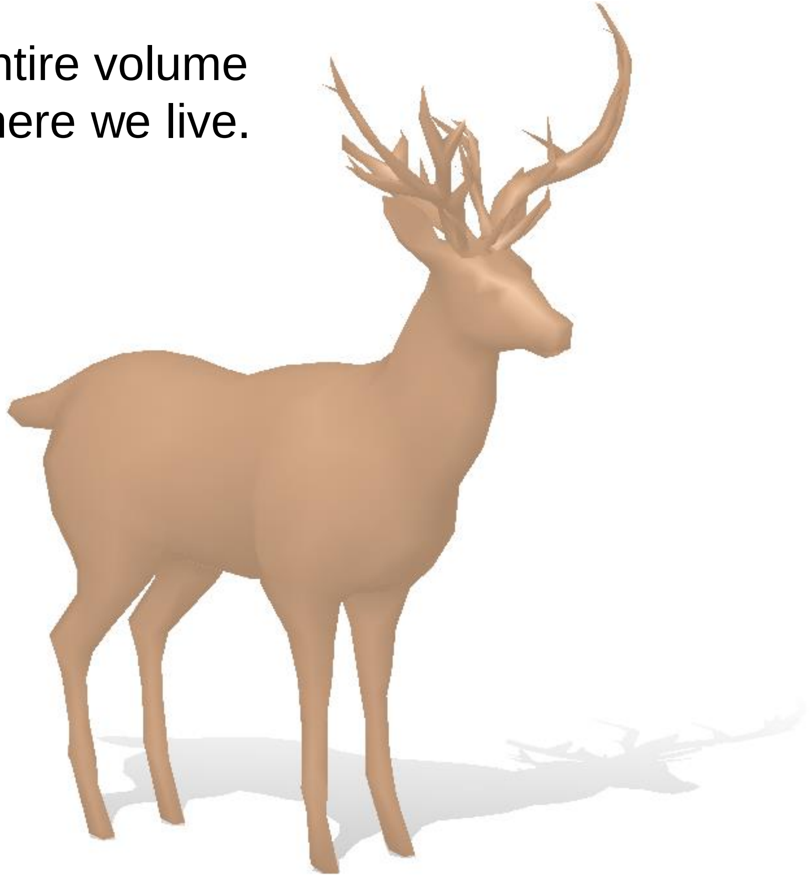
1. **Introduction: 3D Non-Rigid shapes and registration**
2. Spectral representation
3. Axiomatic approaches
4. Functional maps
5. Learning on geometric data
6. Learning–based Functional maps
7. Other learning-based approaches
8. Transformers



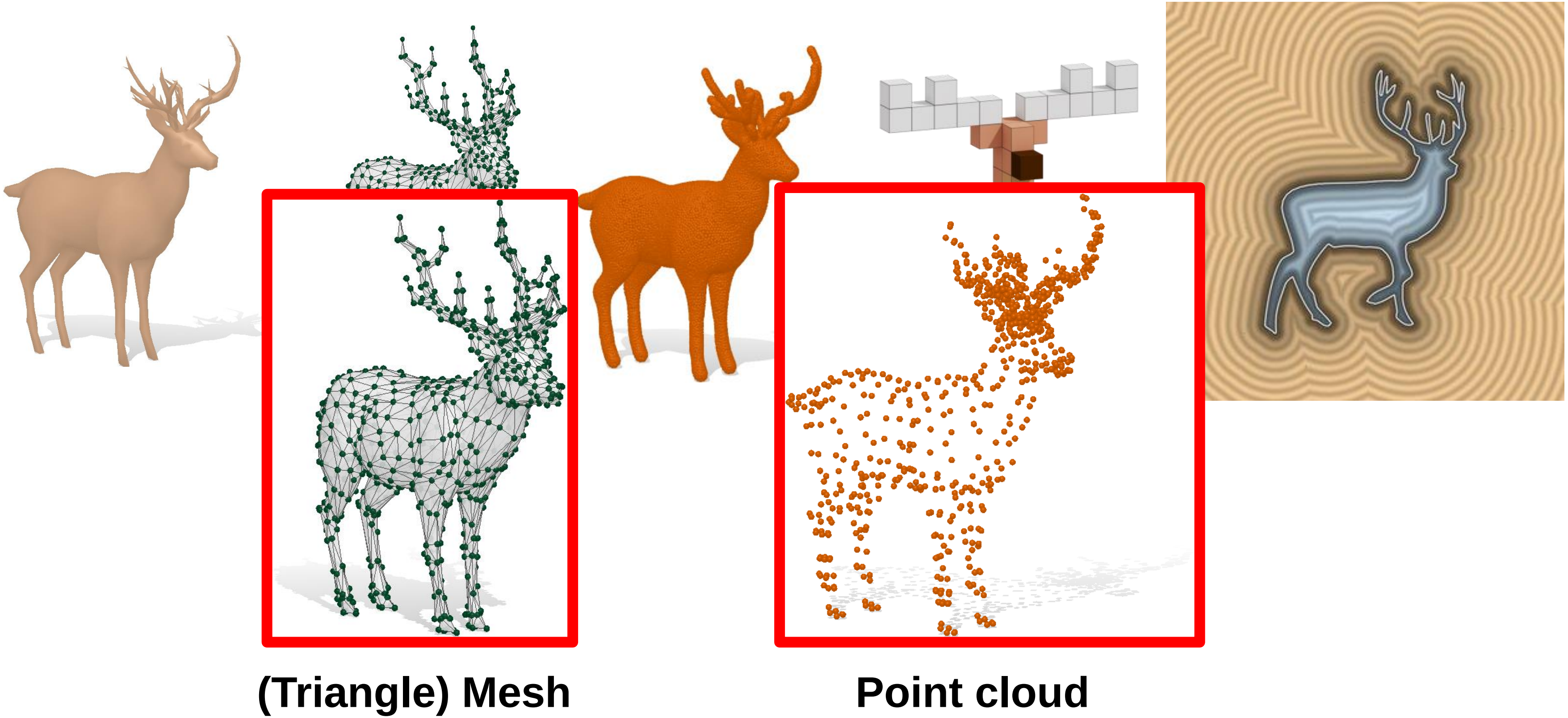
What is a shape:

Real world: the external shell or the entire volume of an object or a scene in the space where we live.

Math: 2-dimensional smooth manifold (Riemannian surface) embedded in $\mathbb{R}^3$ or a dense subset of the 3D space $\mathbb{R}^3$.

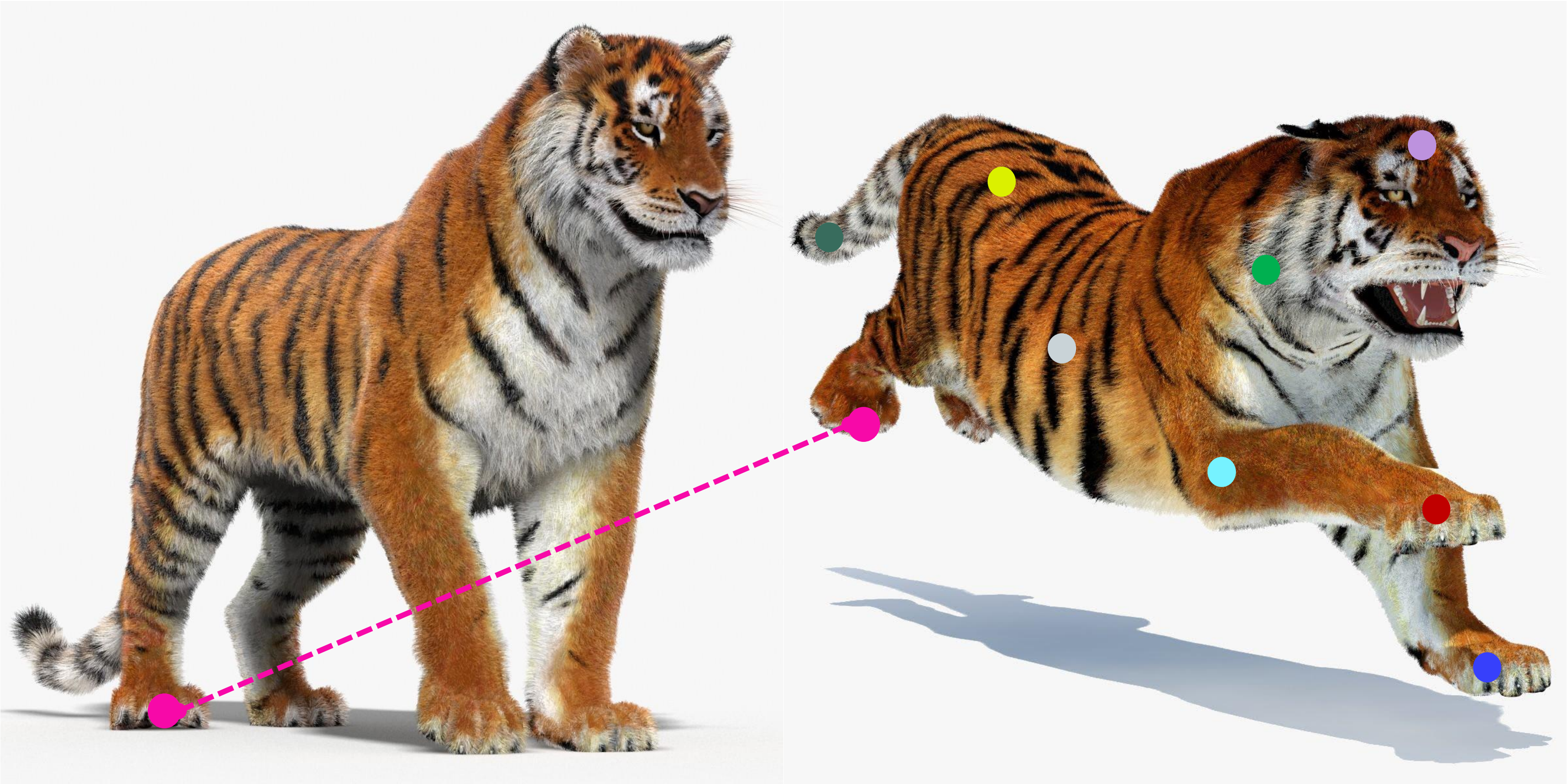


Different representations

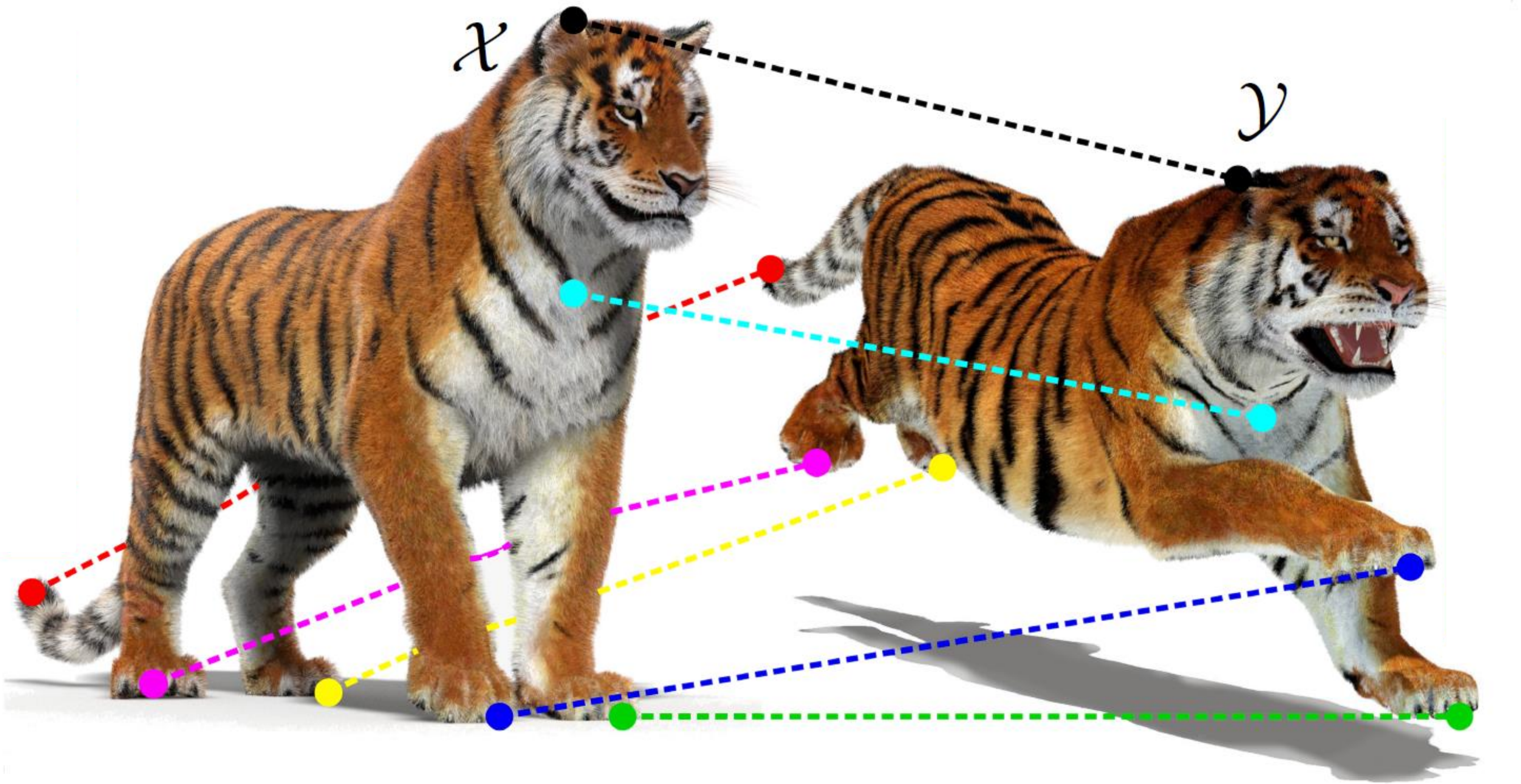


Rigid and Non-Rigid

Matching



Matching



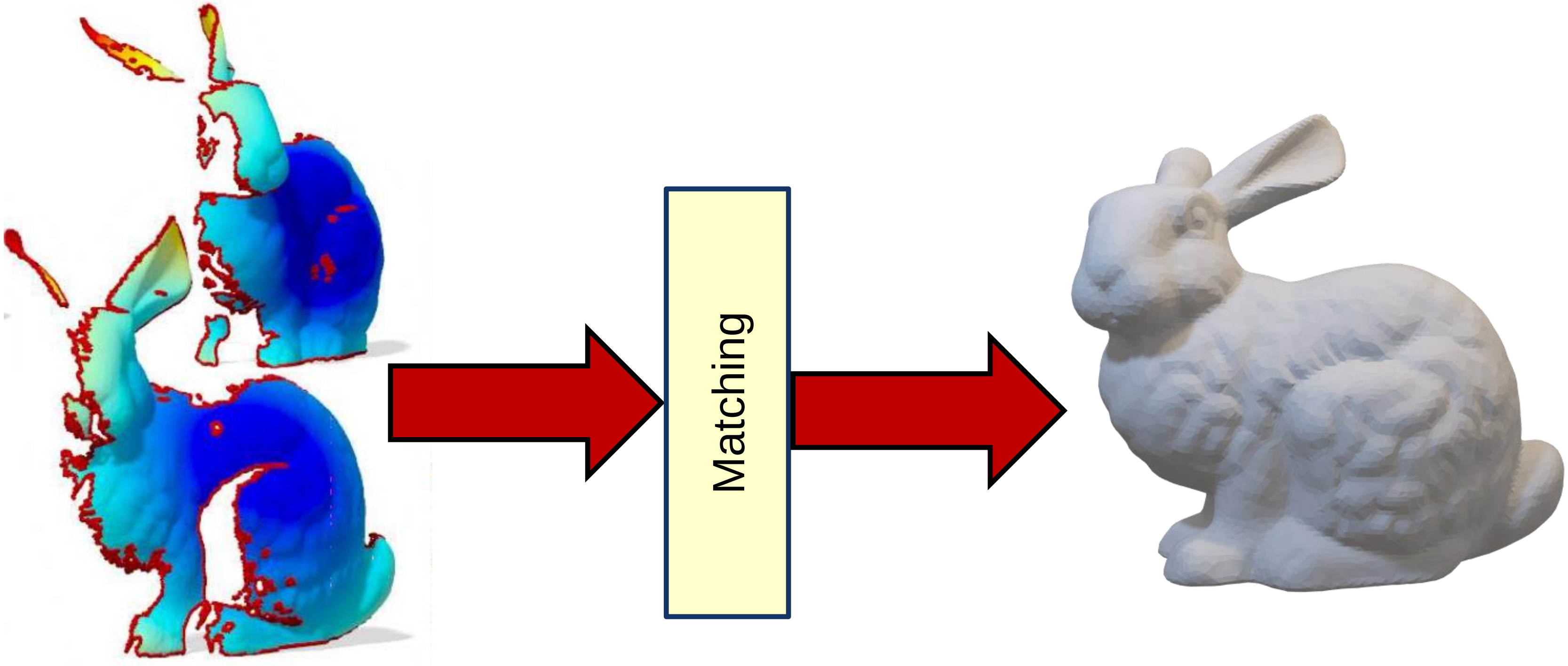
Challaenges



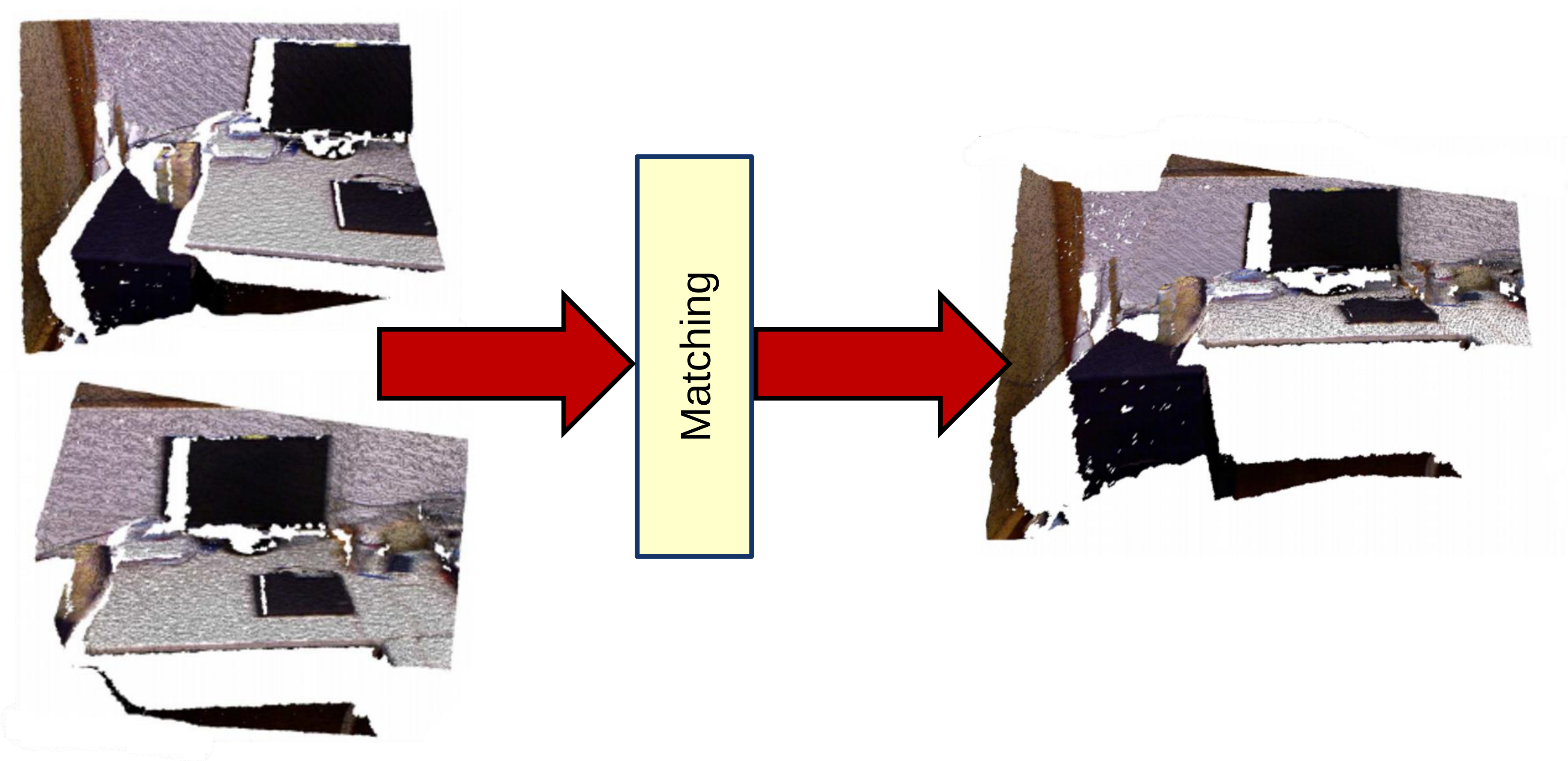




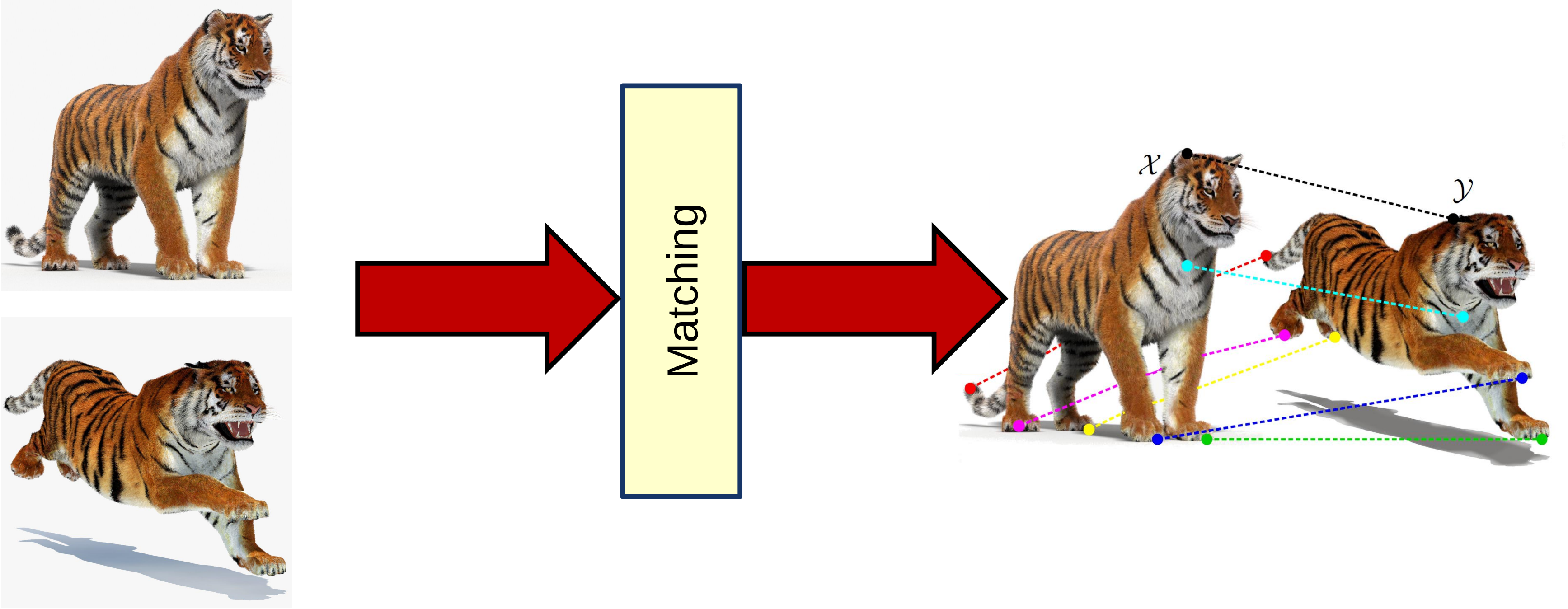
Rigid registration



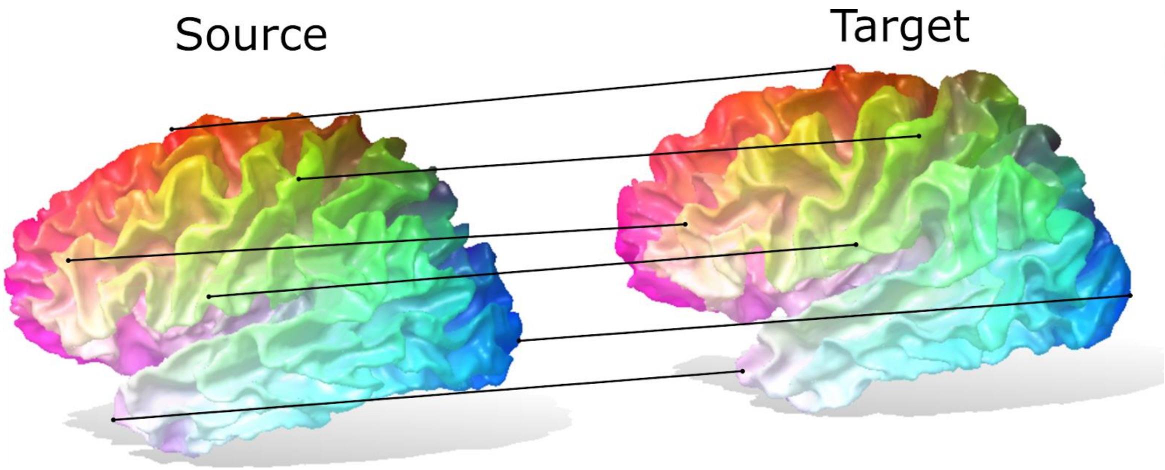
Rigid registration

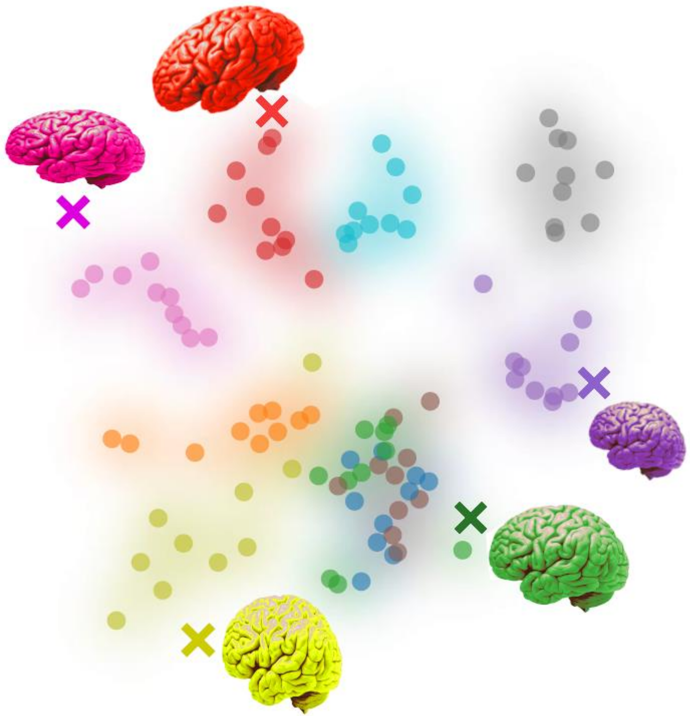


Non-rigid matching/registration

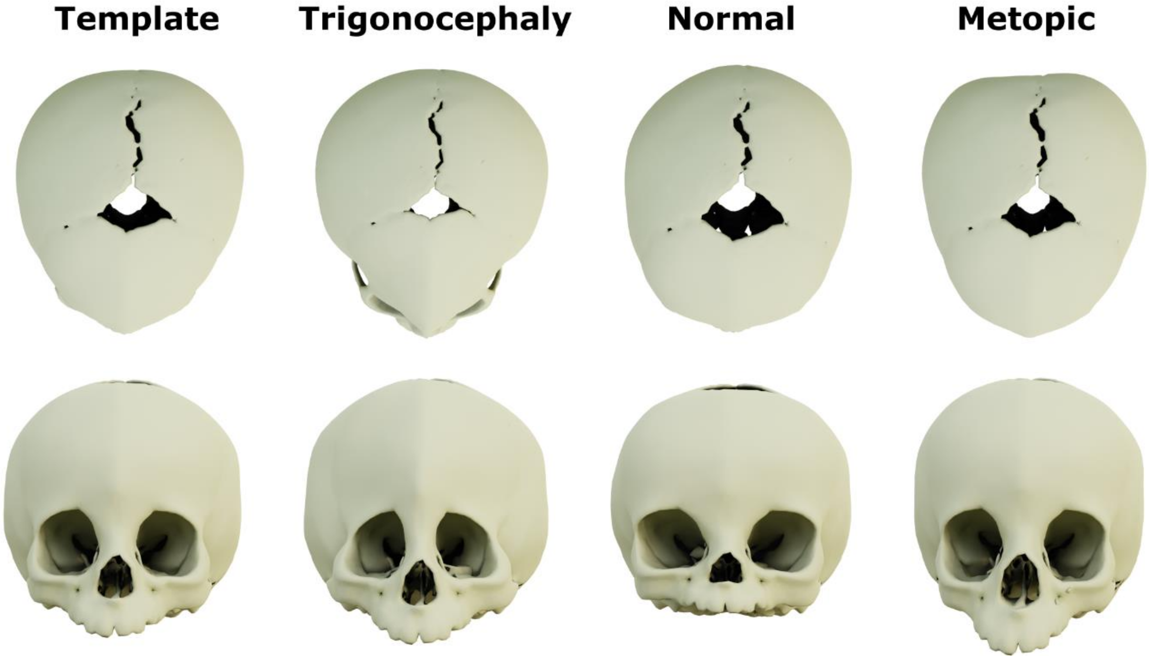


Medical applications

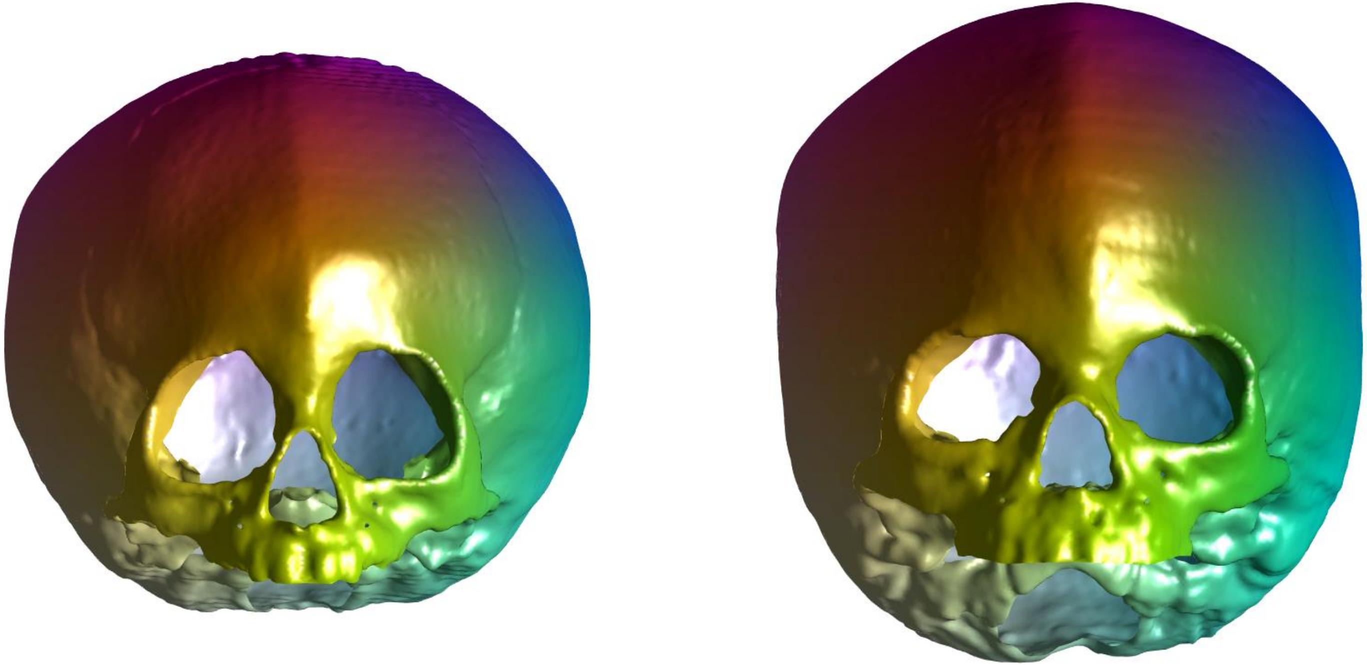




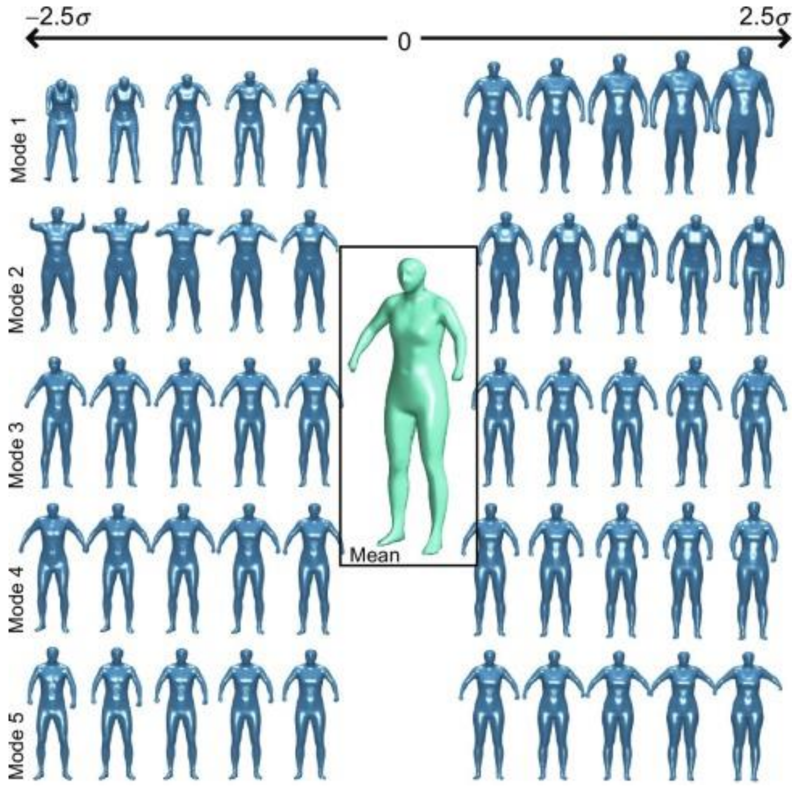
Disease detection and classification

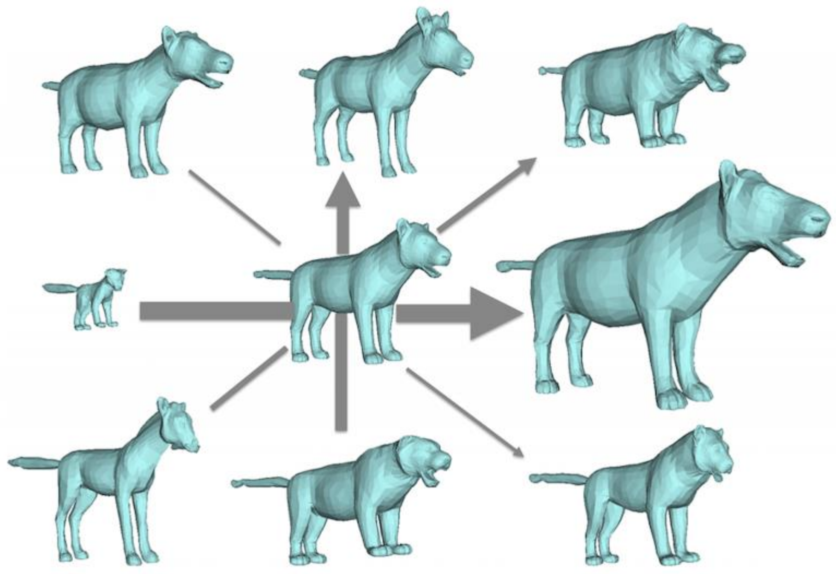


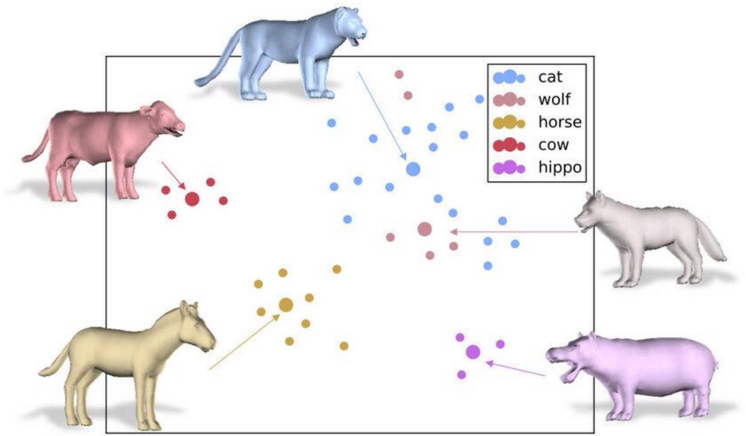
Shape interpolation



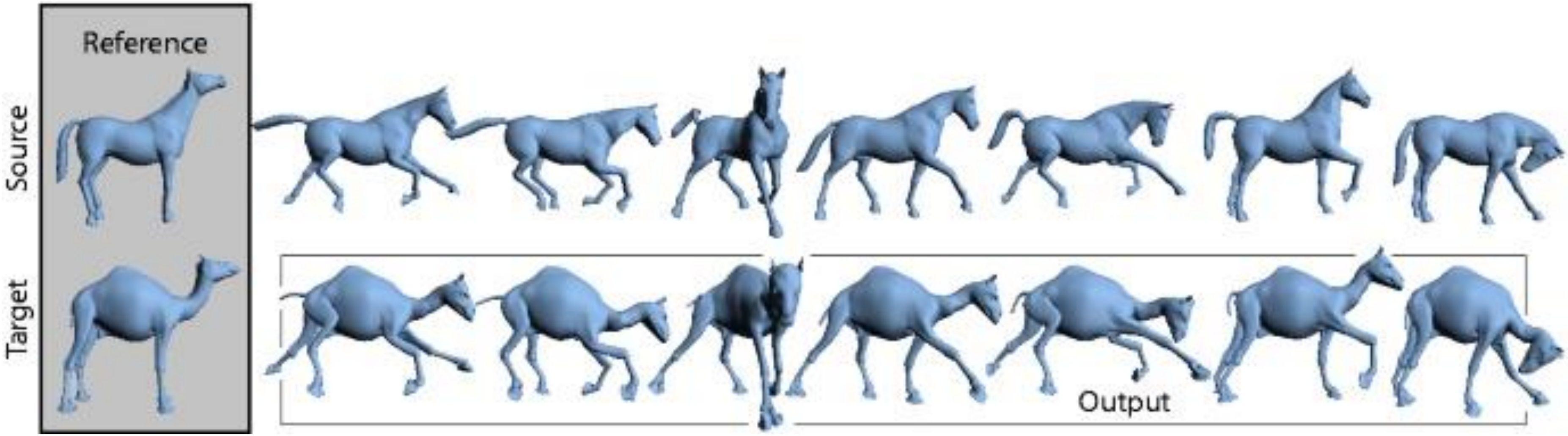
Statistical shape analysis





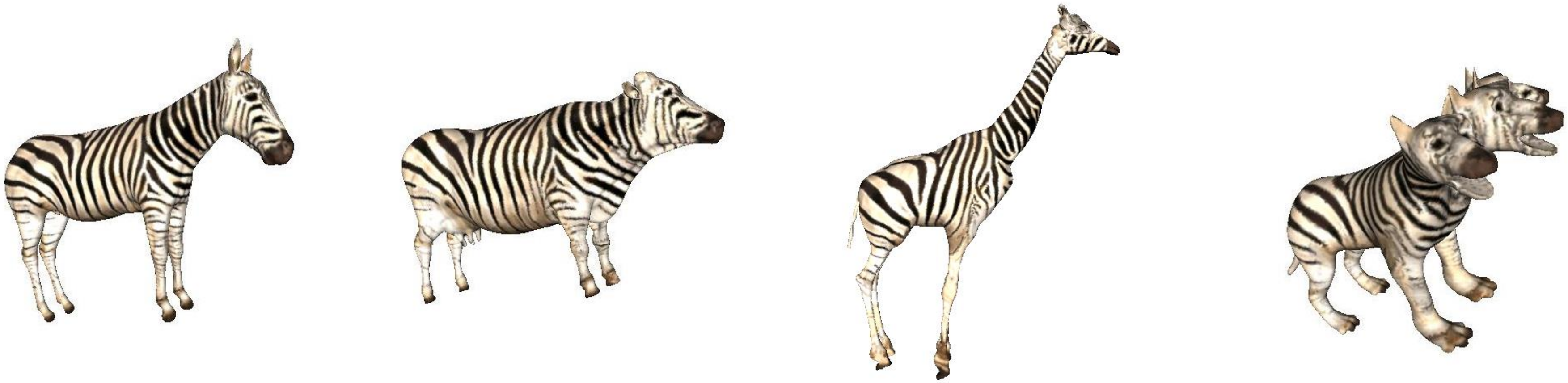


Deformation-transfer



Sumner et al. *Deformation Transfer for Triangle Meshes*, 2004

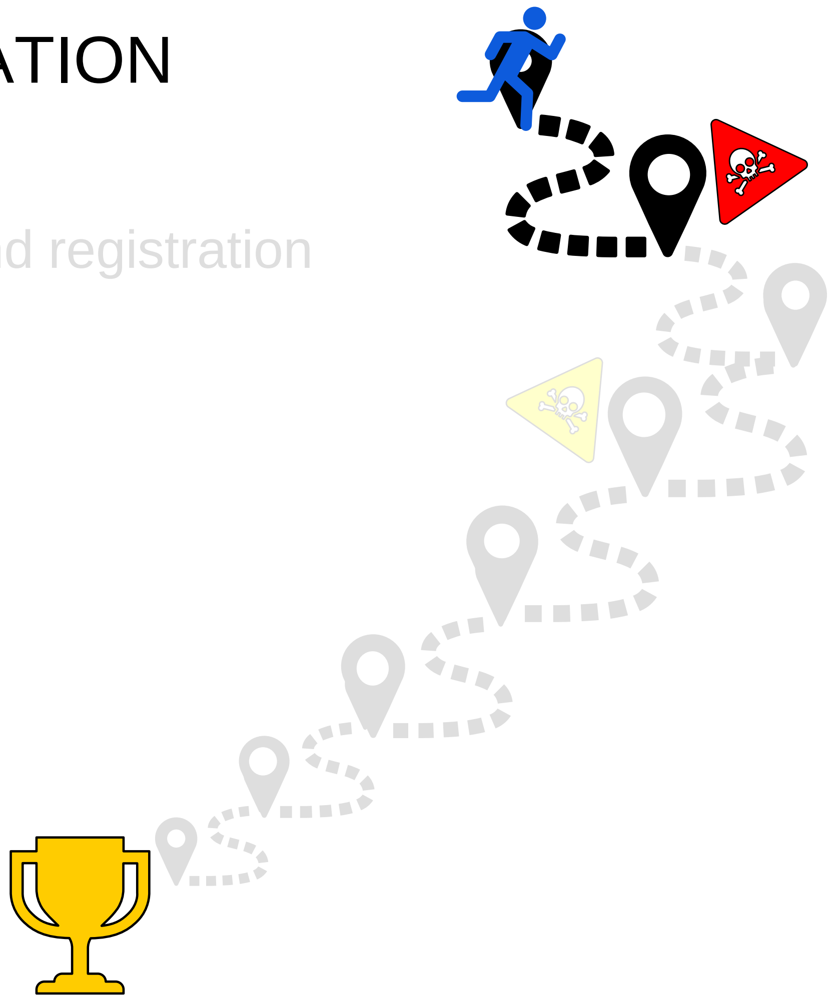
Texture-transfer



Chen et al. *Non-parametric texture transfer using MeshMatch*, 2012

SPECTRAL REPRESENTATION

1. Introduction: 3D Non-Rigid shapes and registration
2. **Spectral representation**
3. Axiomatic approaches
4. Functional maps
5. Learning on geometric data
6. Learning–based Functional maps
7. Other learning-based approaches
8. Transformers

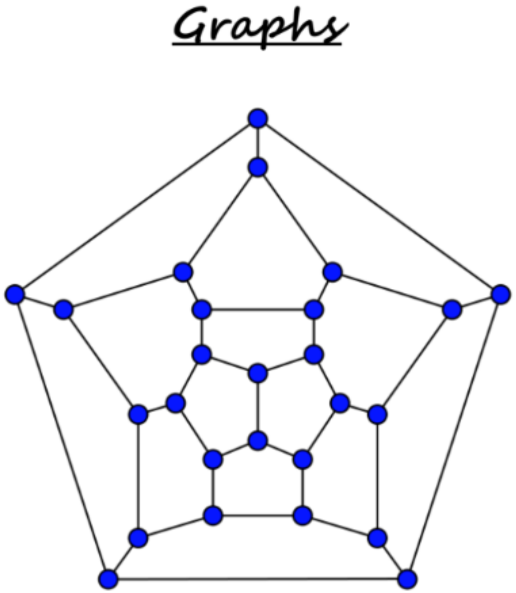


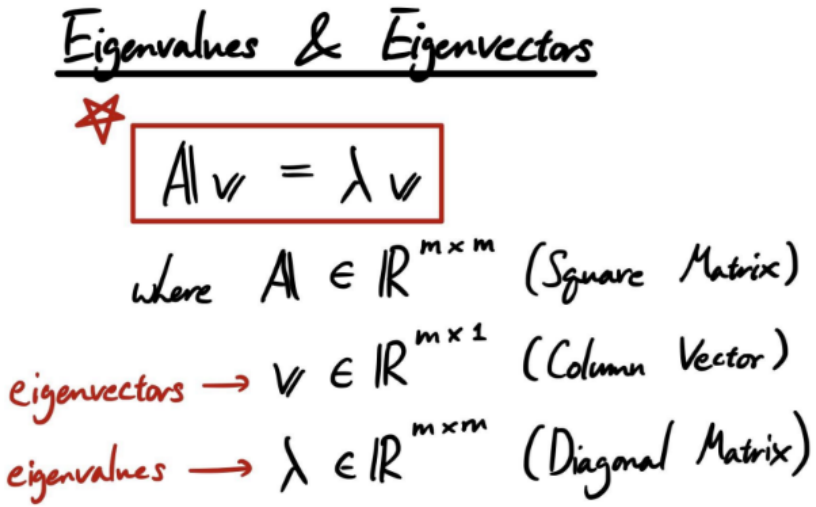
Laplace Operator

A swiss army knife to work with general representations

+ Graphs review
+ Linear Algebra review
+ Intrinsic geometry tools

Graph Spectral Theory

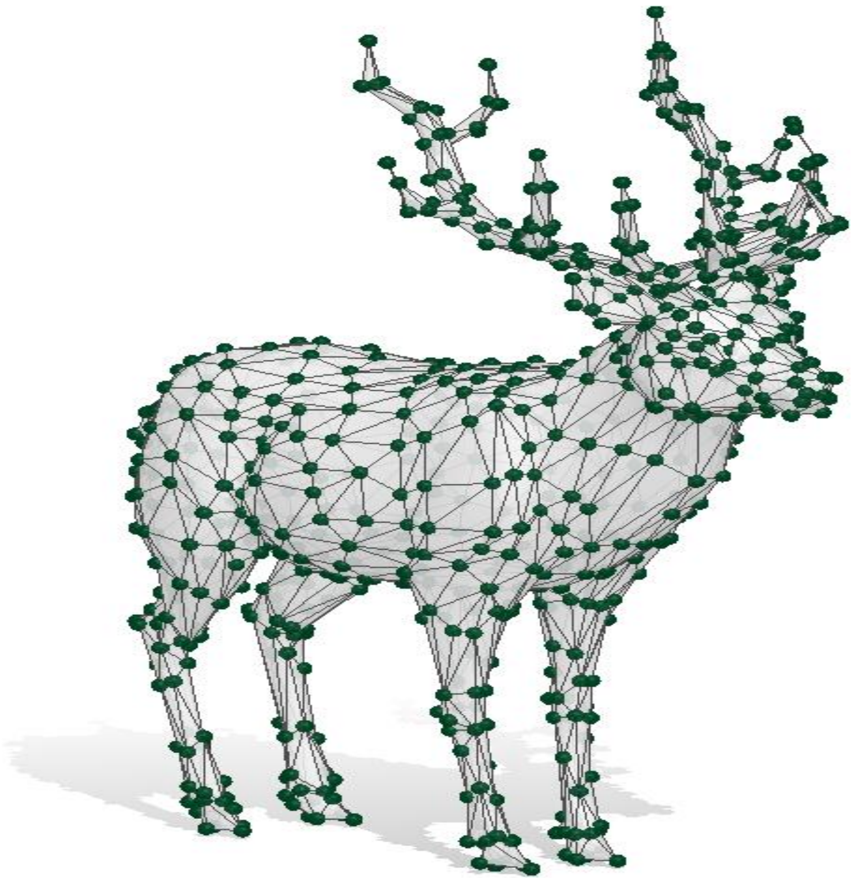




<u>Long story short</u>:

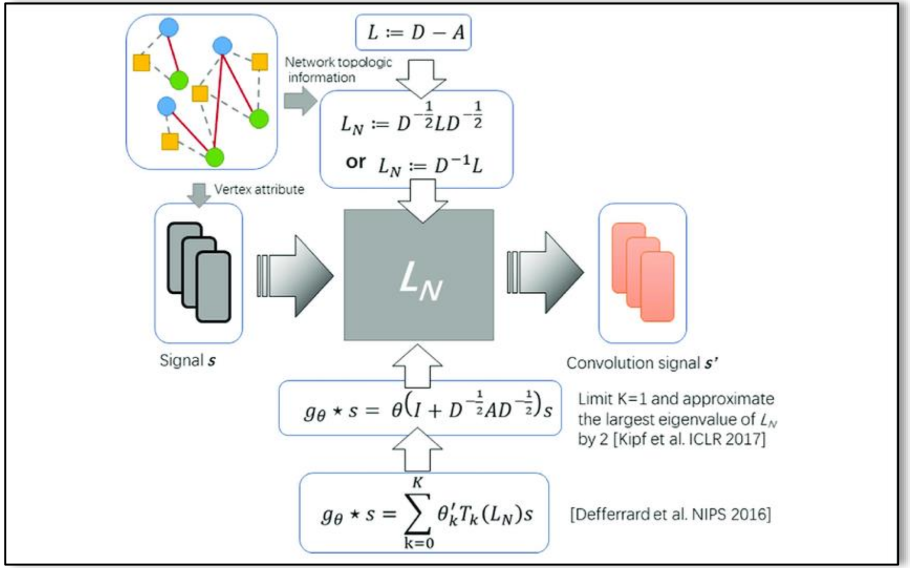
Given a graph, we compute a characteristic matrix
and we consider its eigendecomposition

Meshes are Graphs

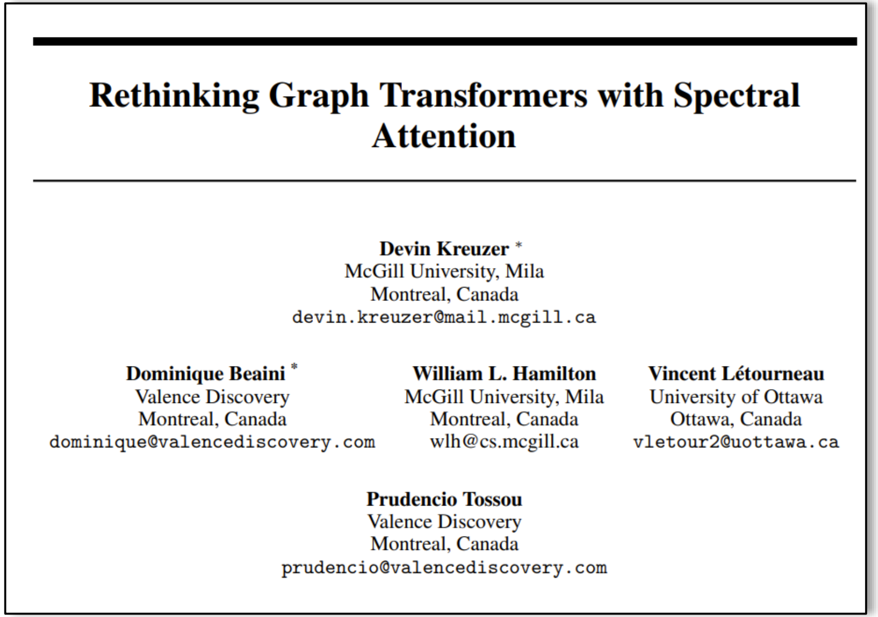


Graph Spectral Theory

Is it relevant?



Convolution



Rethinking Graph Transformers with Spectral Attention

Devin Kreuzer *
McGill University, Mila
Montreal, Canada
devin.kreuzer@mail.mcgill.ca

Dominique Beaini *
Valence Discovery
Montreal, Canada
dominique@valencediscovery.com

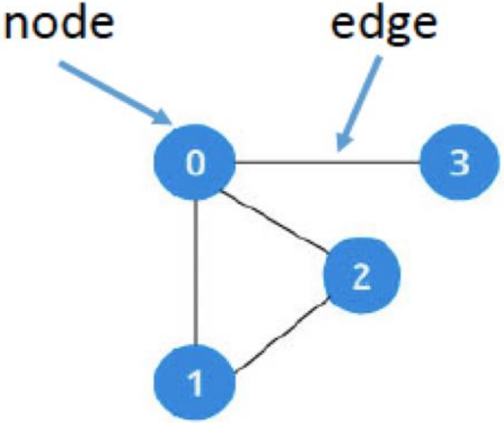
William L. Hamilton
McGill University, Mila
Montreal, Canada
wlh@cs.mcgill.ca

Vincent Létourneau
University of Ottawa
Ottawa, Canada
vletour2@uottawa.ca

Prudencio Tossou
Valence Discovery
Montreal, Canada
prudencio@valencediscovery.com

Positional Encoding

Graph Representation



	0	1	2	3
0	0	1	1	1
1	1	0	1	0
2	1	1	0	0
3	1	0	0	0

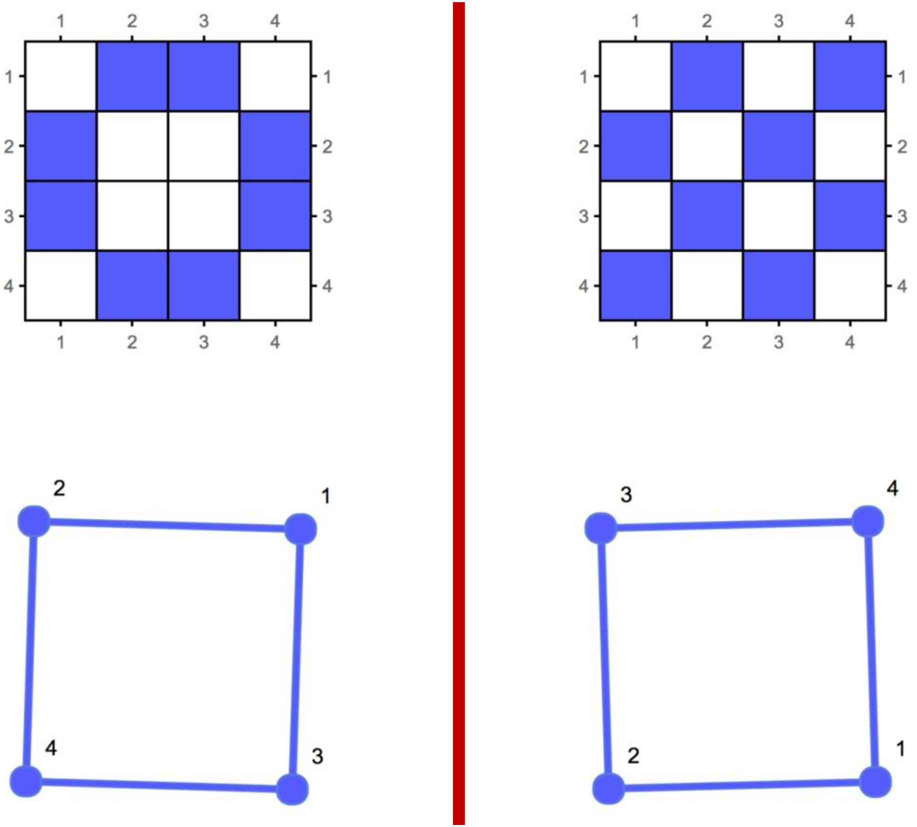
https://medium.com/basecs/a-gentle-introduction-to-graph-theory-77969829ead8

Order of the nodes

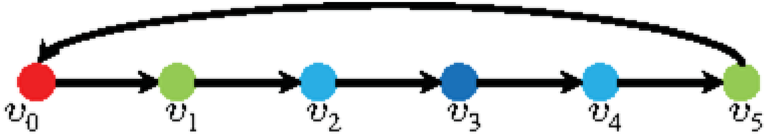
same structure but
different representations

the order of the nodes is
different

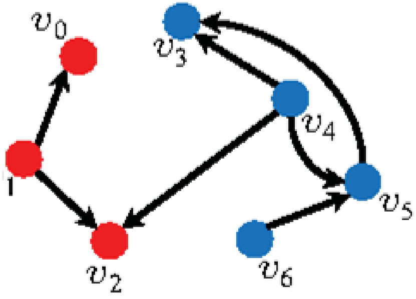
permutation invariant
representation!



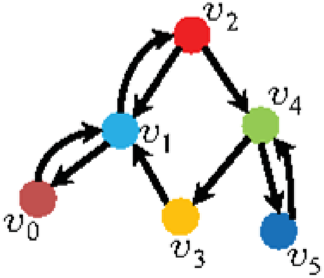
A function/signal over a graph



(a)



(c)



(d)

associates a scalar value to every vertex

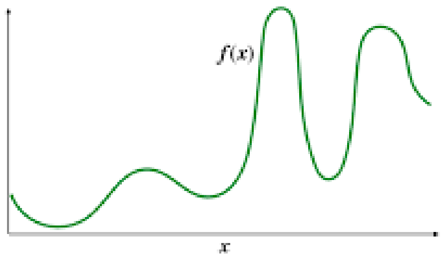
represents it as a color w.r.t. a colorbar

https://noamgit.github.io/2018-12-01-gsp/

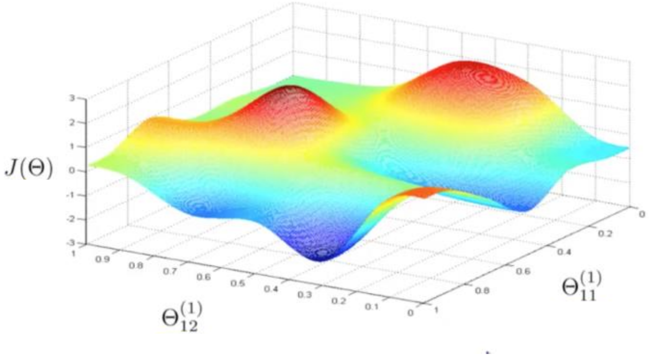
https://www.semanticscholar.org/paper/Big-Data-Analysis-with-Signal-Processing-on-Graphs%3A-Sandryhaila-Moura/65d61afd9c35b0a75d9de77c2a4a2428af0f7f7b/figure/0

Study Functions in Euclidean domains

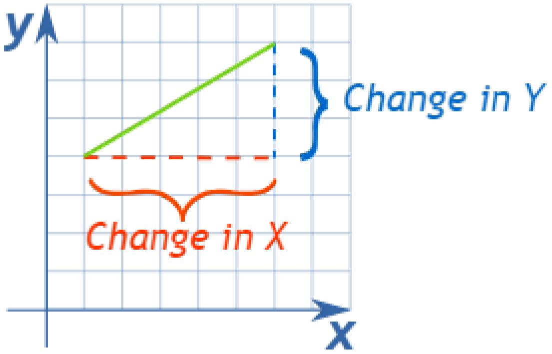
1D signal

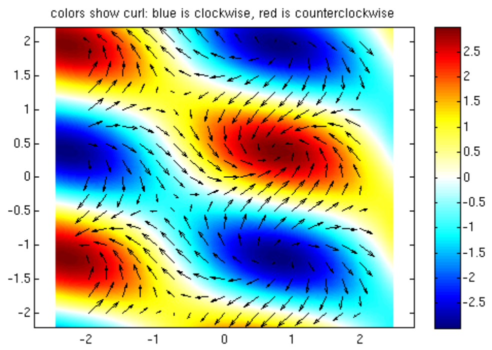


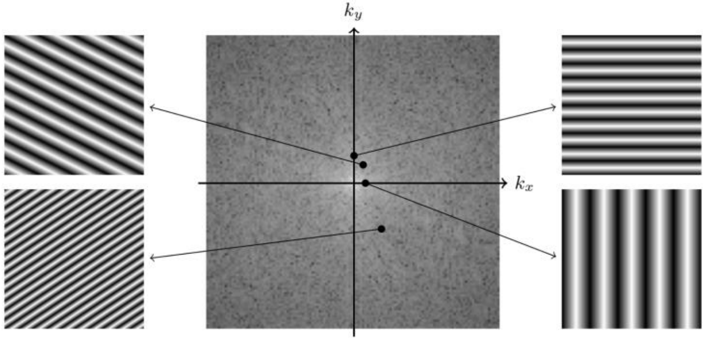
2D signal



Many tools for Function\Signal analysis

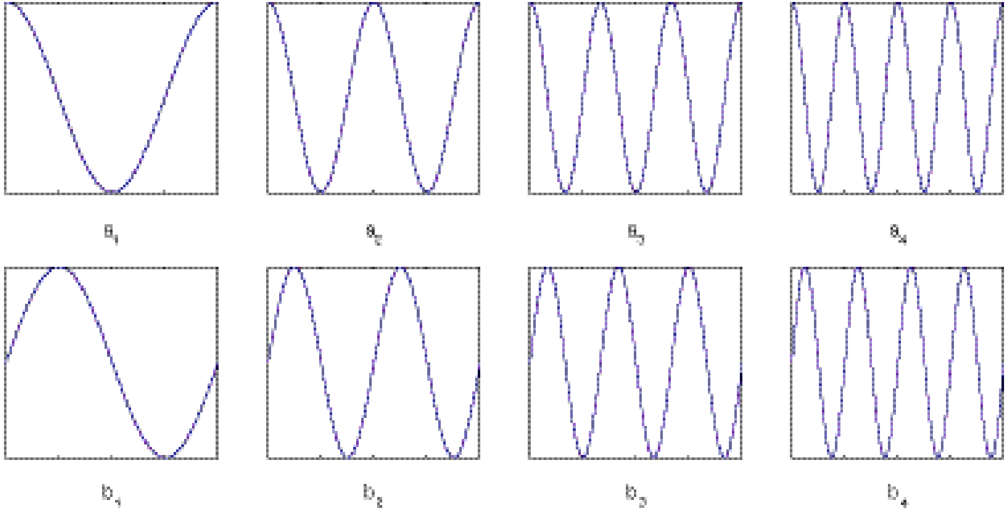






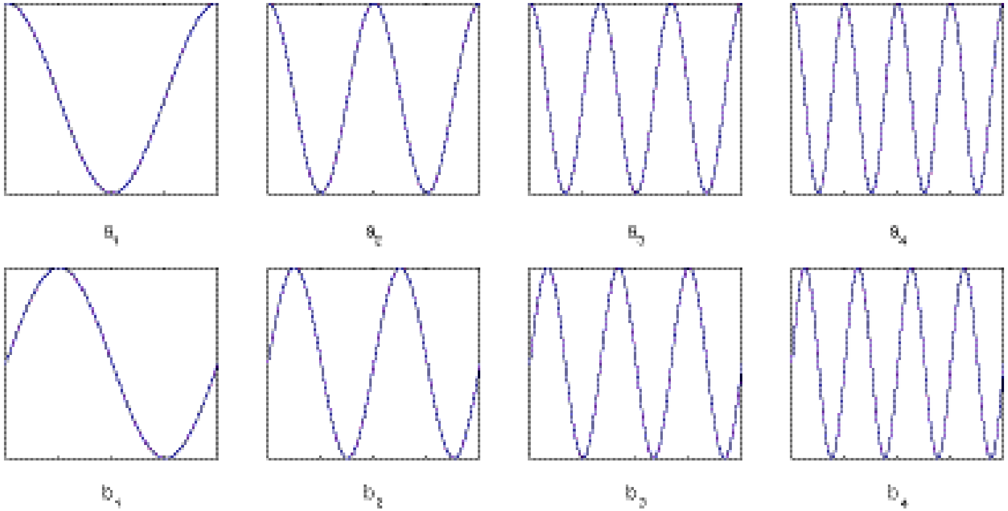
Fourier basis

Standard for signal defined on Euclidean domains

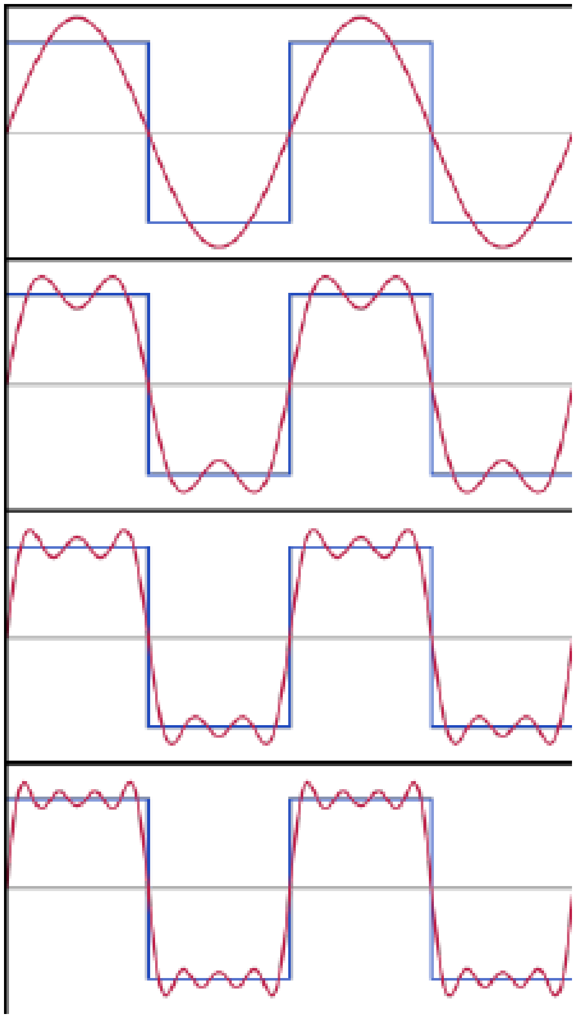


Fourier basis

Standard for signal defined on Euclidean domains



<u>Functions as a linear combination of sinusoids</u>



Linear Algebra recap

Given a vector space V, a subset B is a basis iif:

- Linearly independent
- They span all the vectors of V

$$v = \alpha_1 b_1 + \alpha_2 b_2 + \cdots + \alpha_n b$$

Useful properties:

Orthogonality

$$< b_i, b_j >= 0$$

Normality

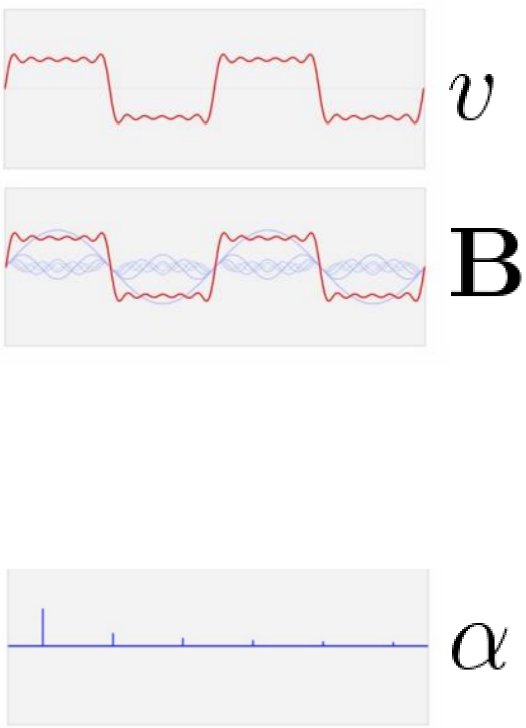
$$< b_i, b_i >= 1$$

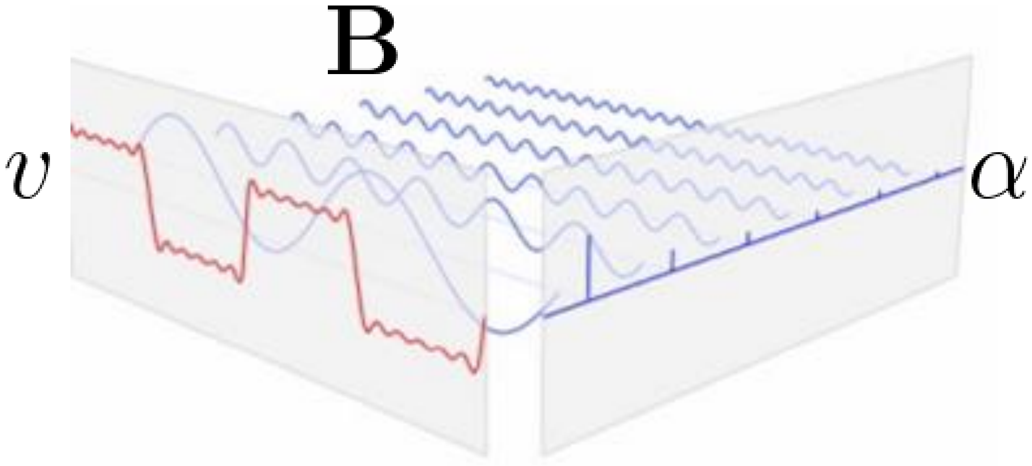
Computing coefficients is much simpler (e.g., α_2)

$$v \cdot b_2 = (\alpha_1 b_1 + \alpha_2 b_2 + \cdots + \alpha_n b_n) \cdot b_2 \ =$$

$$\alpha_2 b_2 \frac{b_2 - \alpha_1 b_1 b_2 - \alpha_3 b_3 b_2 - \cdots - \alpha_n b_n b_2}{||b_2||_2} = \alpha_2$$

Analysis



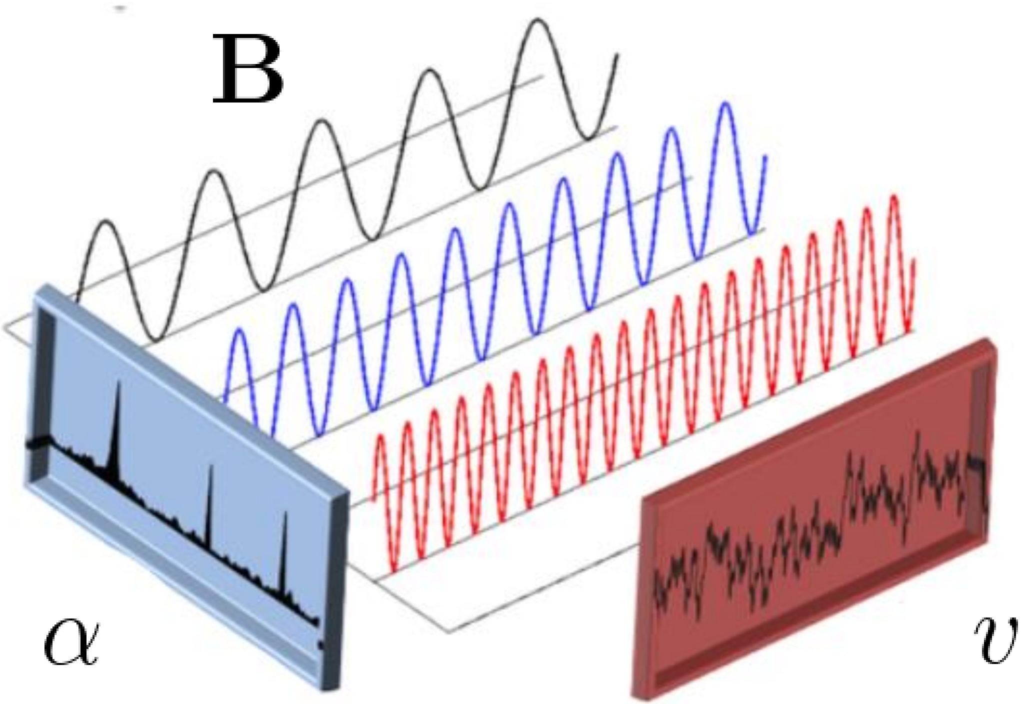


$$\alpha = < B, v > = B^T v$$

Basis projection

Synthesis

Synthesis



$$v = \mathbf{B}\alpha$$

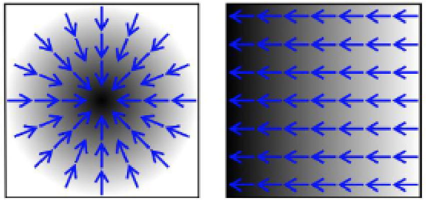
Basis recombination

Gradient, Divergence, Laplacian

Important tools from analysis:

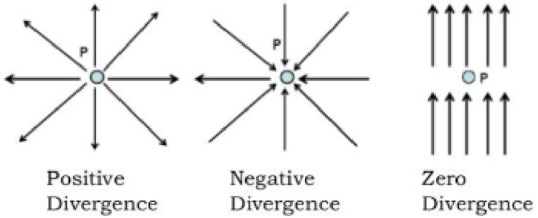
The gradient

$$\nabla : \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}, \ldots\right)$$



The divergence

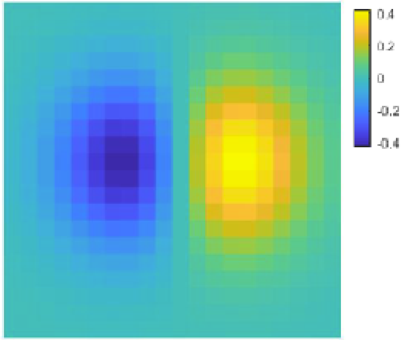
$$div : \frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z} + \ldots .$$

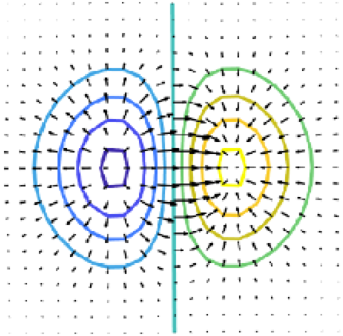


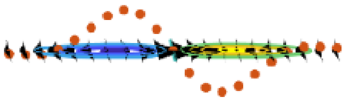
The Laplacian

$$\Delta : div\, \nabla = \frac{\partial^2}{\partial^2 x} + \frac{\partial^2}{\partial^2 y} + \frac{\partial^2}{\partial^2 z} + \ldots$$

The change of a rate of change







Note: in 1D, the Laplacian is equal to the second order derivative.

Fourier Analysis and Laplacian

A fourier basis with frequency ξ $\sin(x2\pi\xi)$

Apply the Laplacian
(second order derivative) $\Delta\sin(x2\pi\xi)$

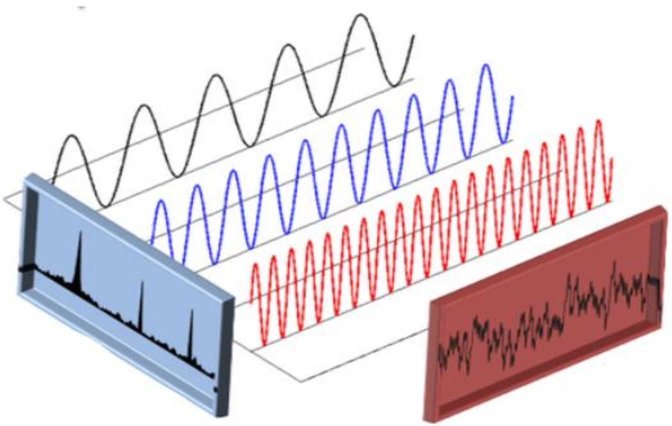
$$\Delta\sin(x2\pi\xi) = -4\pi^2\xi^2\sin(x2\pi\xi)$$

The Fourier basis functions are the eigenfunctions of the Laplacian

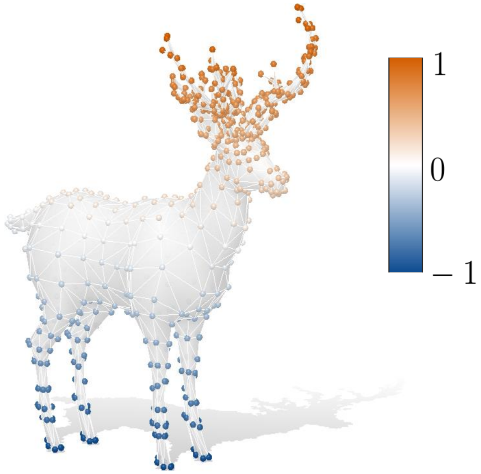
The Laplacian in 1D coincides with the sum of the second order derivatives

Our problem

Define tools like this



on



Graphs & Meshes
Non Euclidean Domain

Summary so far:

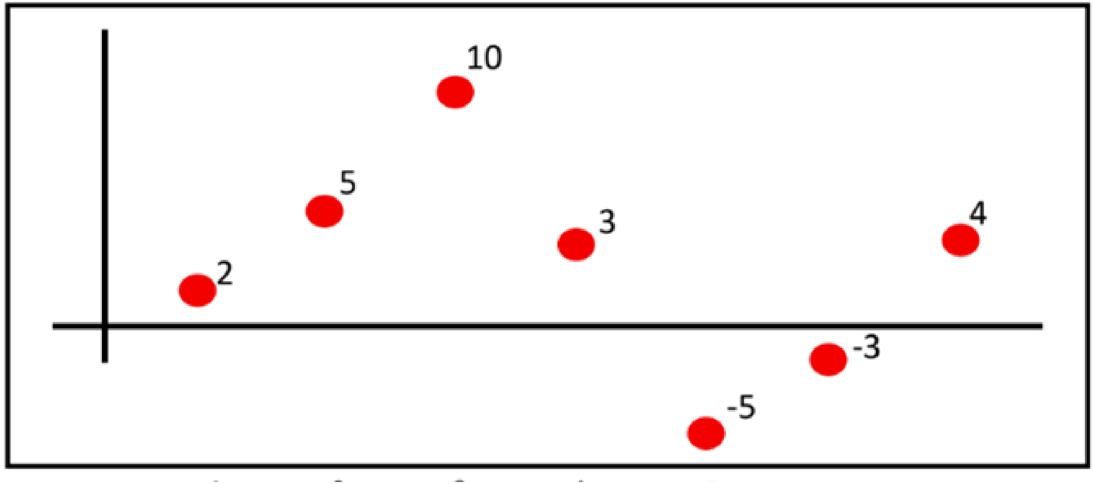
- **Graphs**: general object, with nodes and a connectivity
- **Functions on Graphs**: vectors (scalar value for each node)
- **Euclidean analysis tools**: Fourier analysis
- **Fourier basis:** Eigenfunctions of the Laplacian

Key idea:

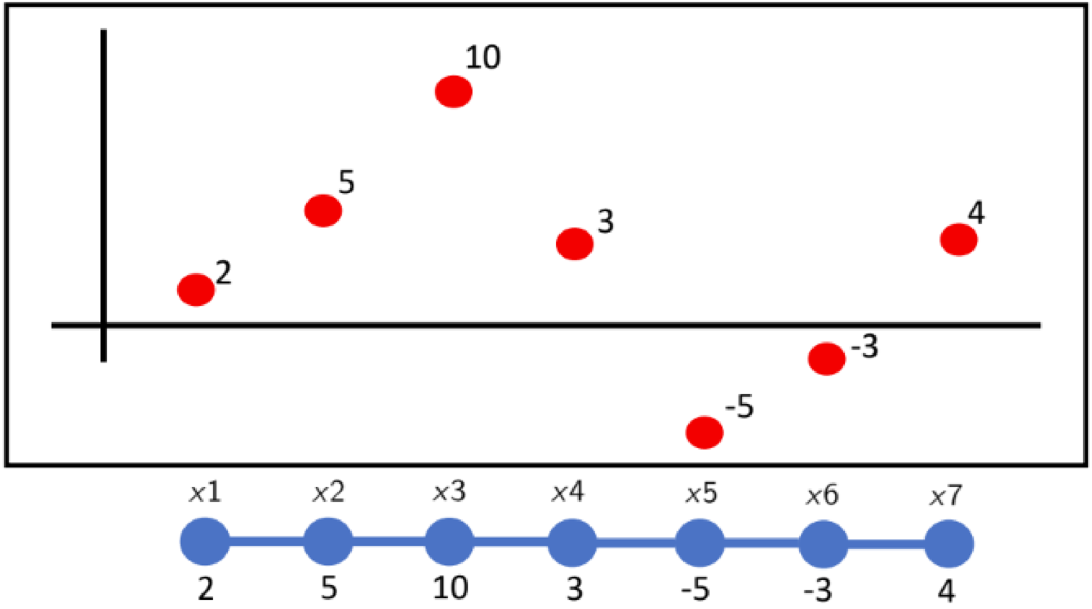
We need a Laplacian on graphs!

Trick: Adapting the definition

Discrete setting (1D)



Discrete setting (1D)

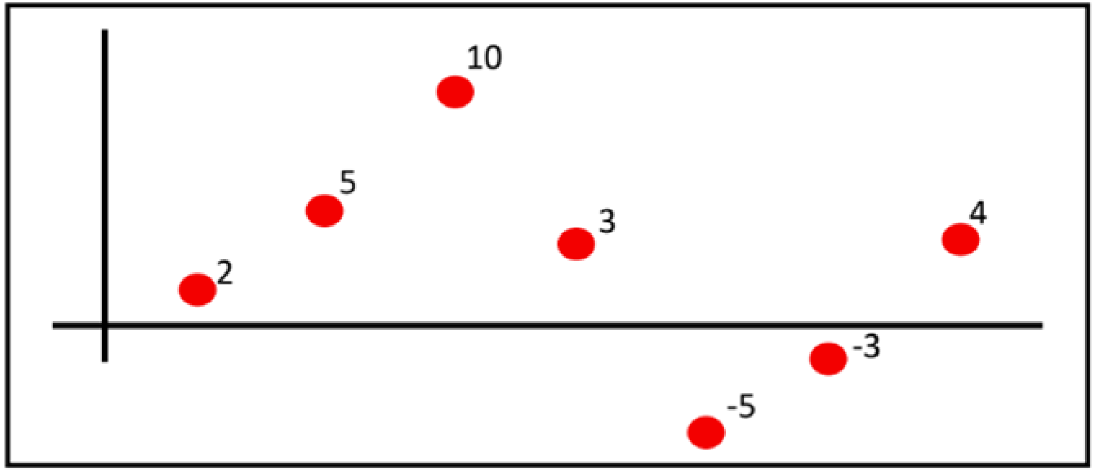


Vertices $\{x1, x2, x3, x4, x5, x6, x7\}$

Edges connect consecutive points

The y-value is a function on the graph $F : V \to \mathbb{R}$

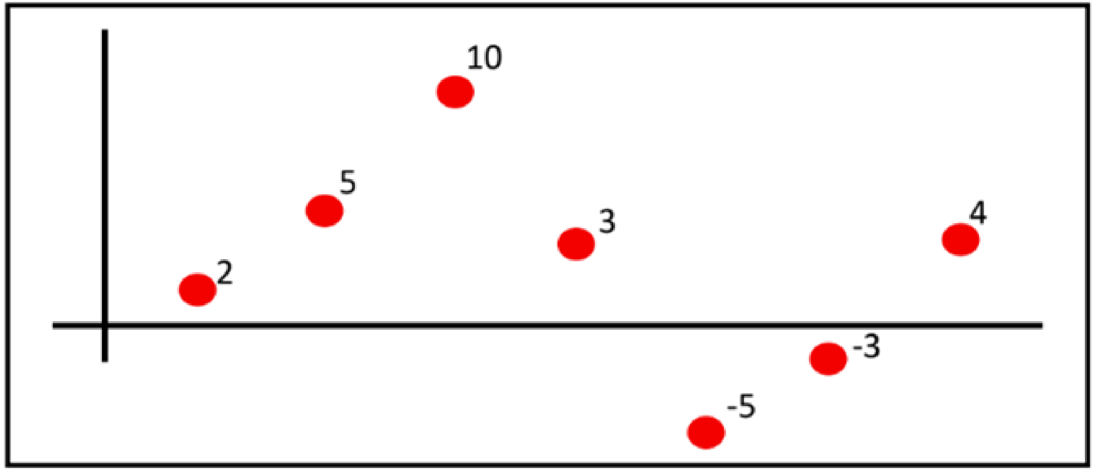
Discrete setting (1D)



$f(x) =$	2	5	10	3	-5	-3	4

<u>**Functions as vectors**</u>

Discrete setting (1D)



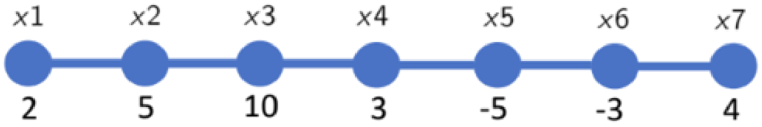
$f(x) =$	2	5	10	3	-5	-3	4

We can represent functions as a linear combination of a basis (Fourier basis)

Discrete Setting

$f(x) =$

2	5	10	3	-5	-3	4

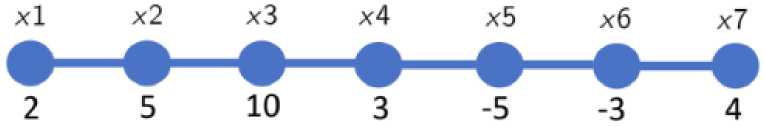


Discrete Setting

$f(x) =$	2	5	10	3	-5	-3	4

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$$

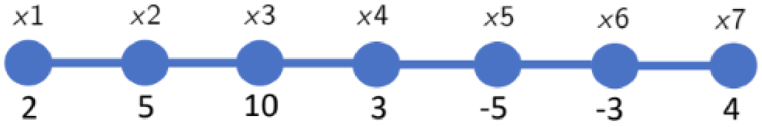
$f'(x) =$	3	5	-7	-8	2	7	...



Discrete Setting

$f(x) =$

2	5	10	3	-5	-3	4



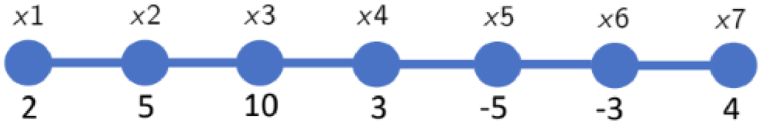
$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$$

$f'(x) =$

3	5	-7	-8	2	7	...

$$f''(x_i) = \frac{f'(x_i) - f'(x_i - 1)}{h} = \cdots = \frac{f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)}{h^2}$$

Discrete Setting



$f(x) =$

2	5	10	3	-5	-3	4

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$$

$f'(x) =$

3	5	-7	-8	2	7	...

$$f''(x_i) = \frac{f'(x_i) - f'(x_i - 1)}{h} = \cdots = \frac{f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)}{h^2}$$

assuming $h = 1$:

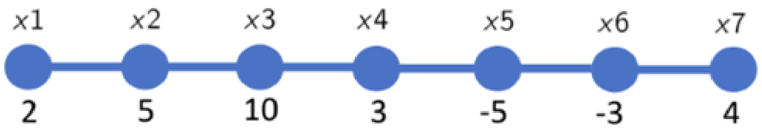
$$f''(x_i) = f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)$$

$f''(x) =$

...	2	-12	-1	10	5	...

$f(x) =$

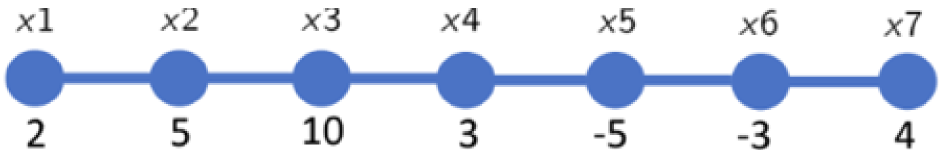
2	5	10	3	-5	-3	4



$$f''(x_i) = f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)$$

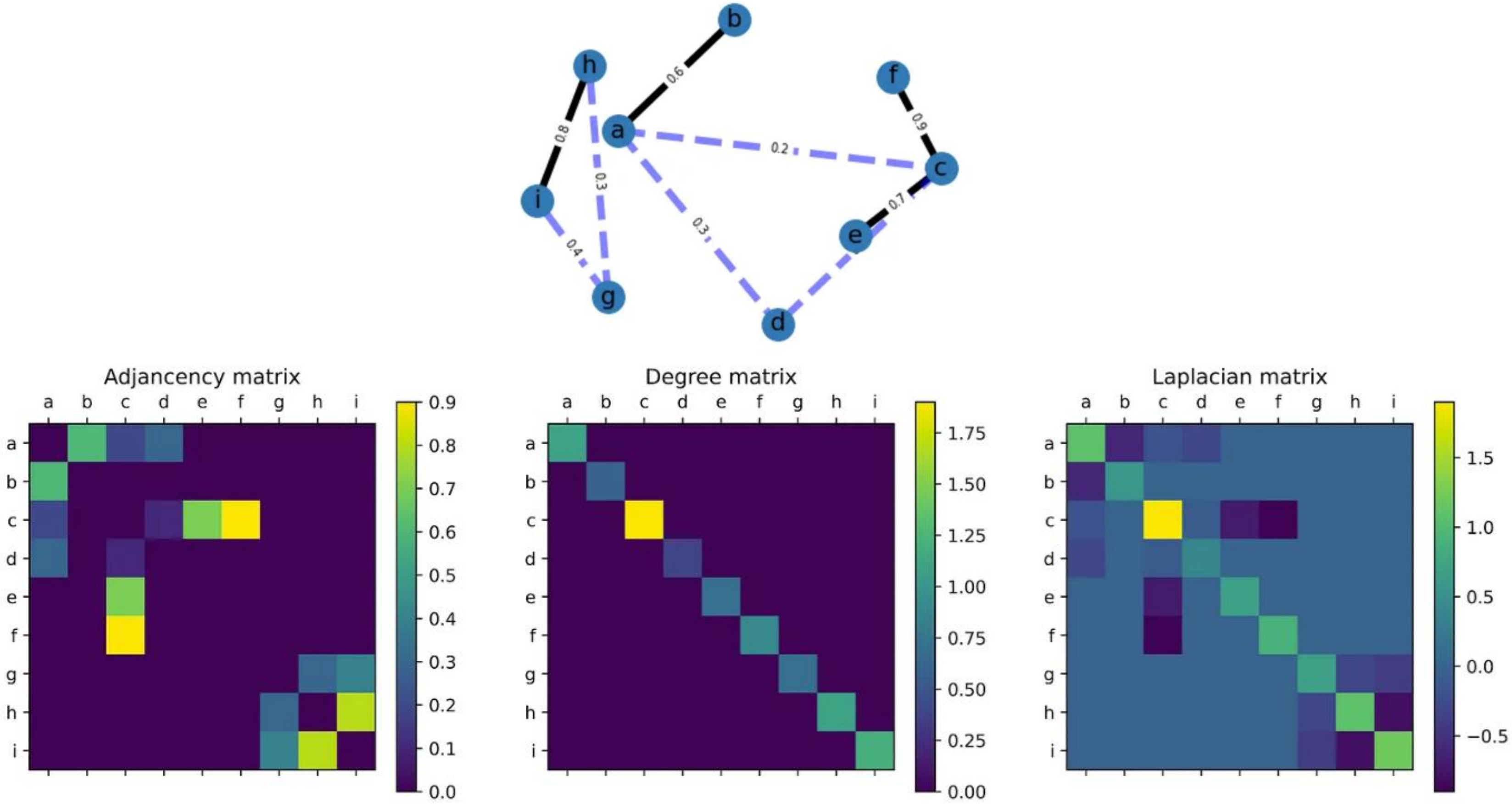
$$f''(x_i) = Lf = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \\ 10 \\ 3 \\ -5 \\ -3 \\ 4 \end{bmatrix} = \begin{bmatrix} \dots \\ 2 \\ -12 \\ -1 \\ 10 \\ 5 \\ \dots \end{bmatrix}$$

$$\begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$



$L = W - D$, where D is the diagonal matrix of the degrees, and W is the weighted adjacency matrix.

$L = W - D$, where D is the diagonal matrix of the degrees, and W is the weighted adjacency matrix.



Discrete Fourier Analysis and Laplacian

A fourier basis with frequency i-th

Discrete Setting:
it is a vector

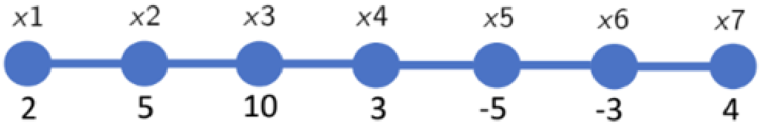
Apply the Laplacian
(second order derivative)

The Laplacian
is a matrix

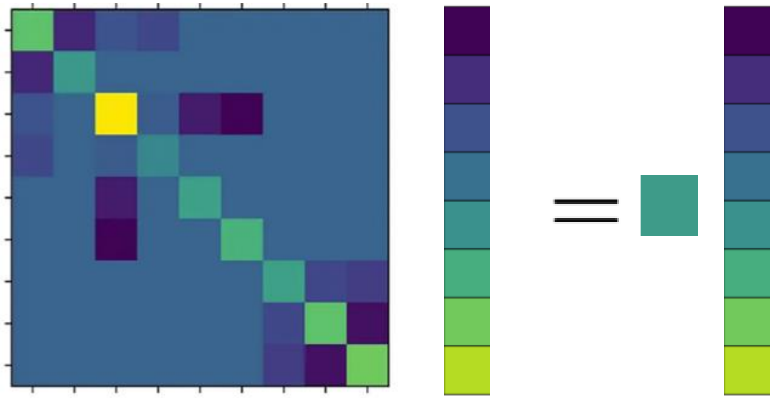
$$L\phi_i = \lambda_i \phi_i$$

The vectors of the Fourier basis are the eigenvectors of the Laplacian

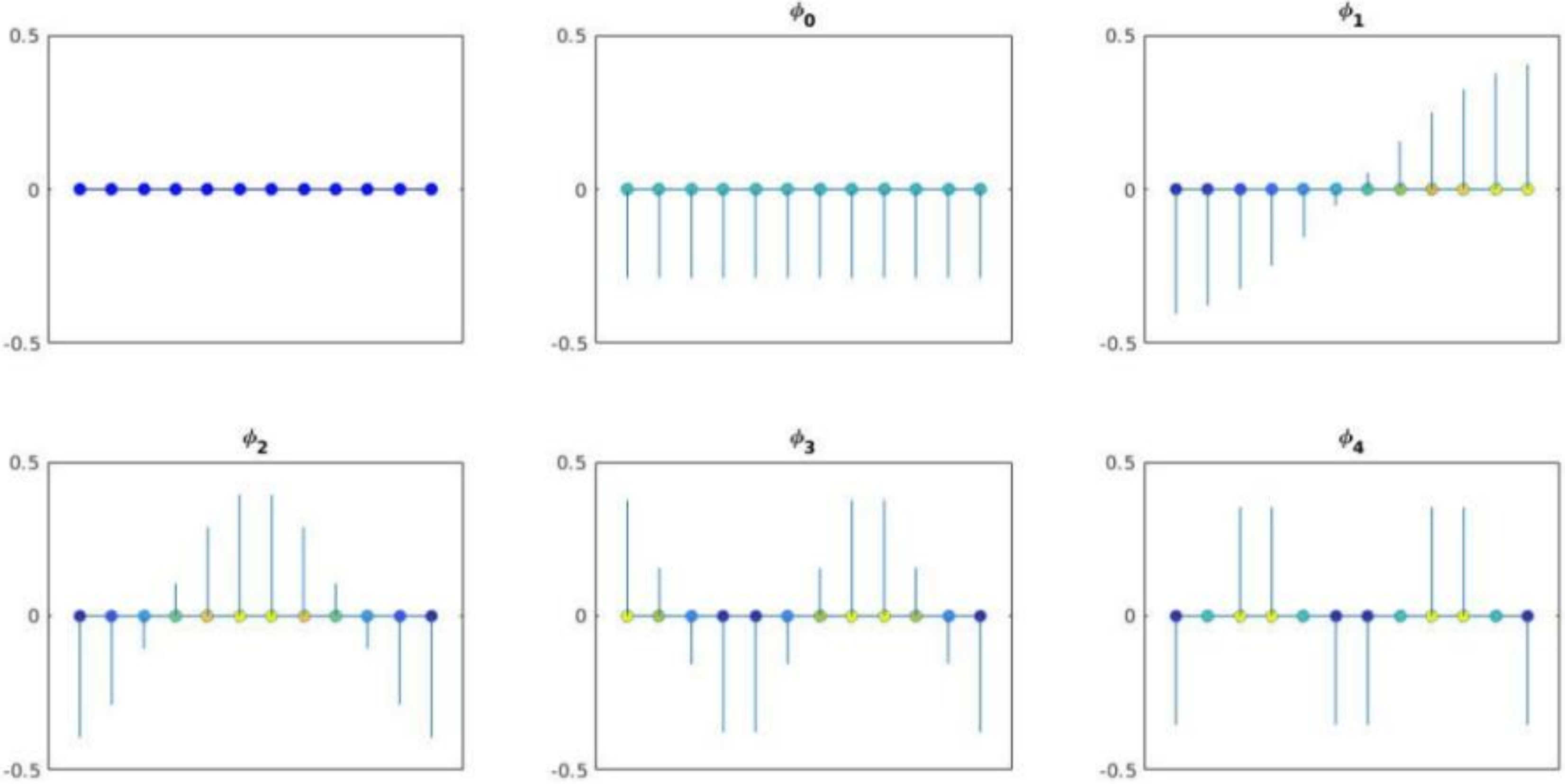
Eigendecomposition



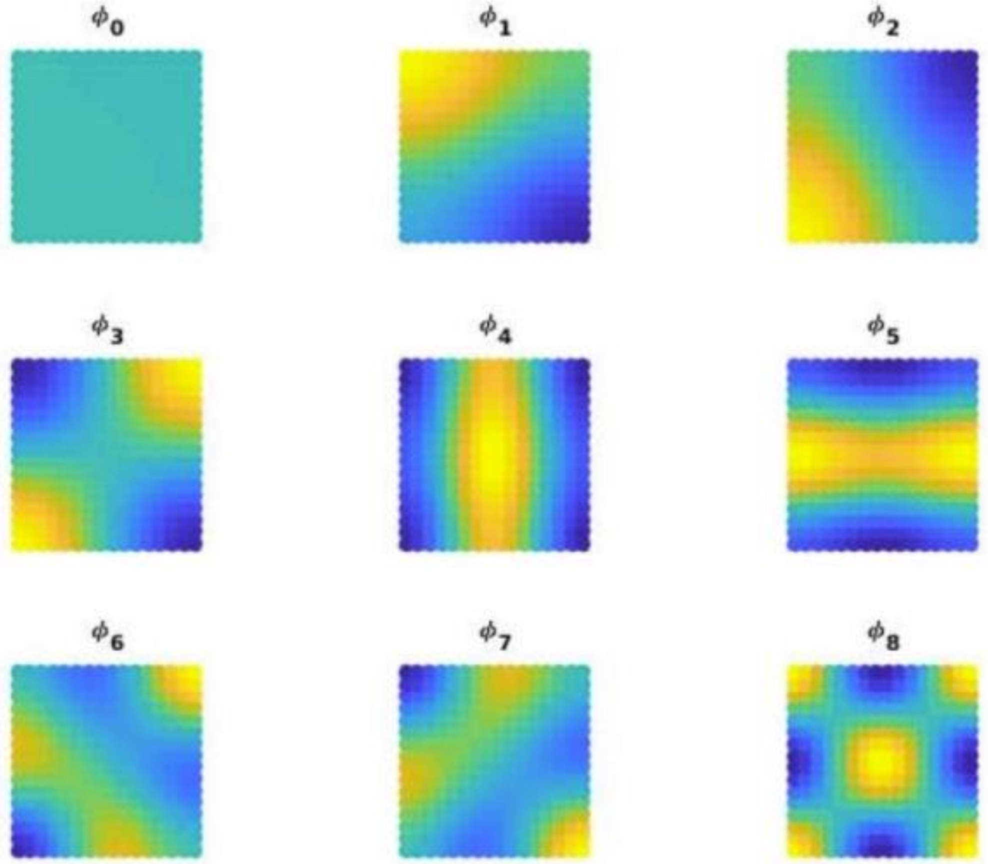
$$L\phi_i = \lambda_i \phi_i$$



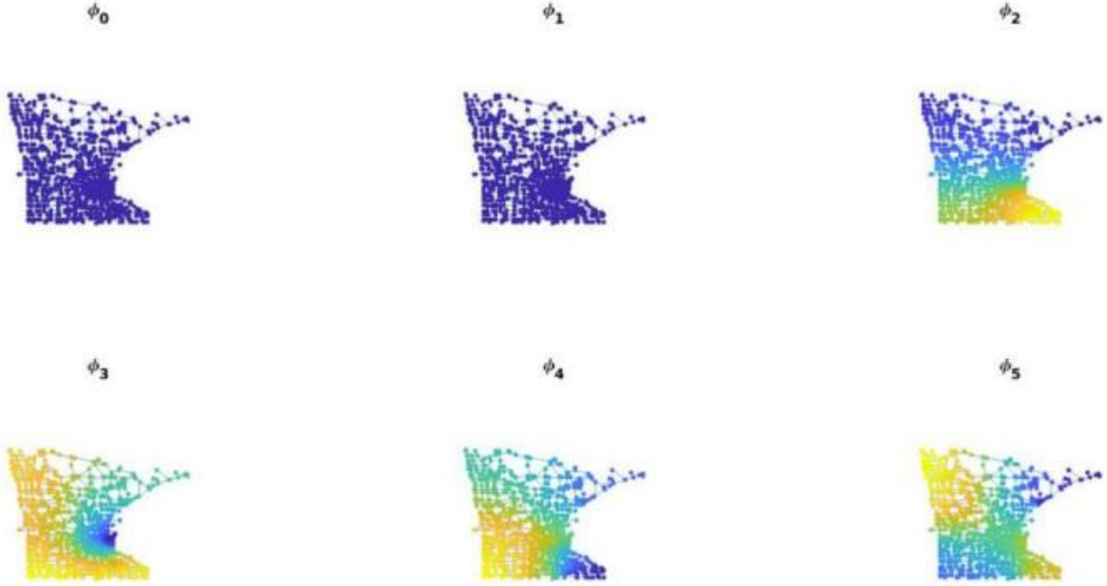
Laplacian Eigenvectors – 1D



Laplacian Eigenvectors – 2D



Laplacian Eigenvectors – Generic Graph



Minnesota Road Map, D. Gleich, 2010,https://www.cise.ufl.edu/research/sparse/matrices/Gleich/minnesota.html

Summary so far:

- **Graphs**: general object, with nodes and a connectivity
- **Functions on Graphs**: vectors (scalar value for each node)
- **Euclidean analysis tools**: Fourier analysis
- **Fourier basis:** Eigenfunctions of the Laplacian
- **Discretization of the Laplacian:** a sparse square matrix
- **Discrete Fourier basis:** Eigenvectors of the Laplacian

Next Goal

use it in applications on graphs and 3D data

Graph Laplacian for 3D shapes (LBO)

Bruno Levy

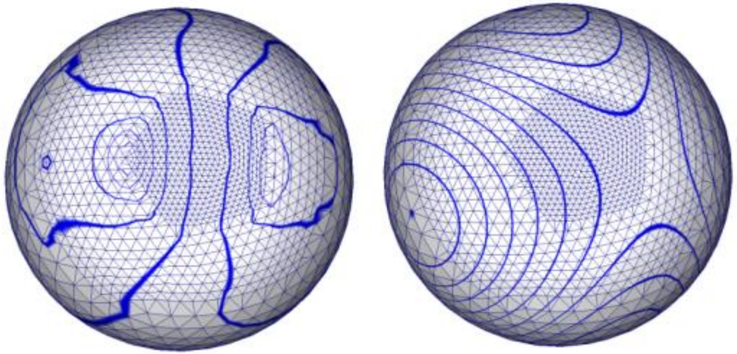
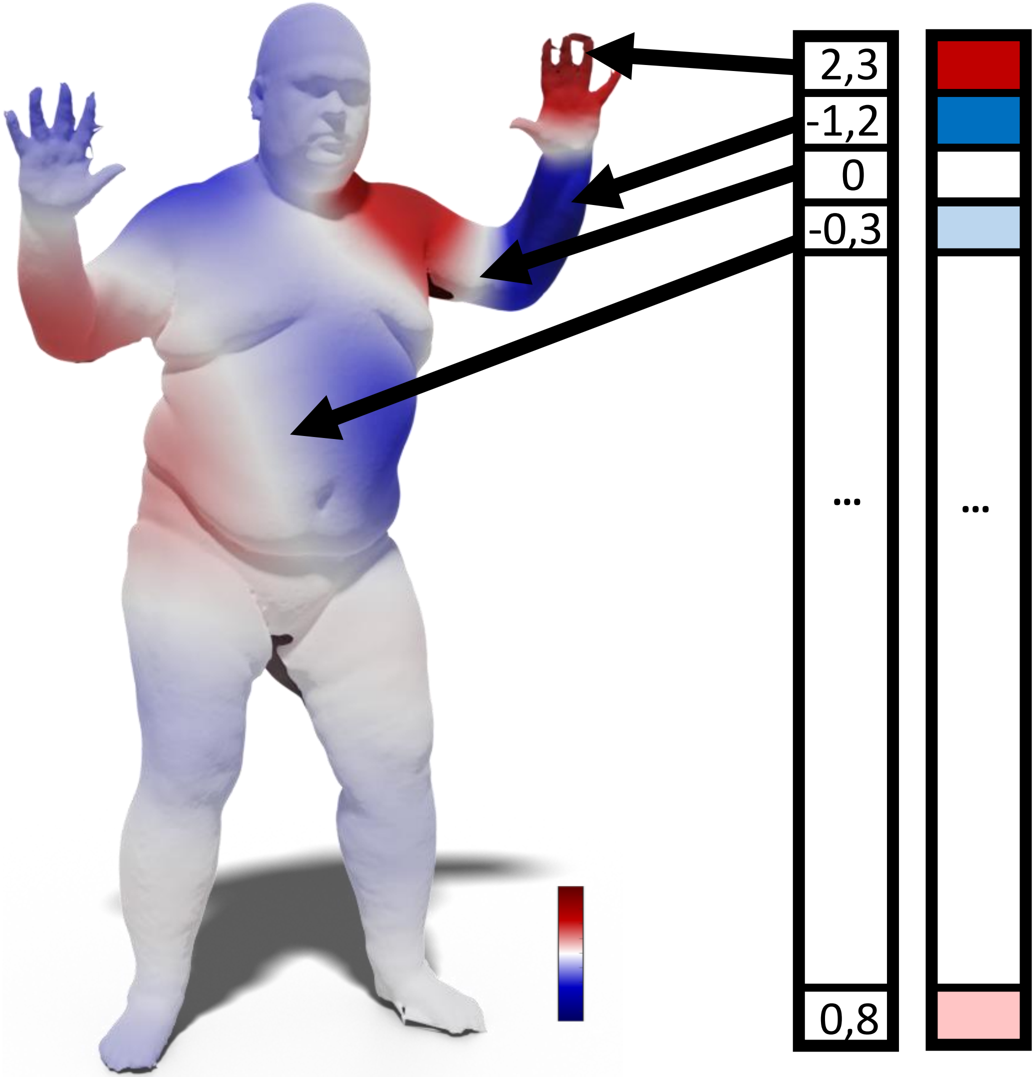


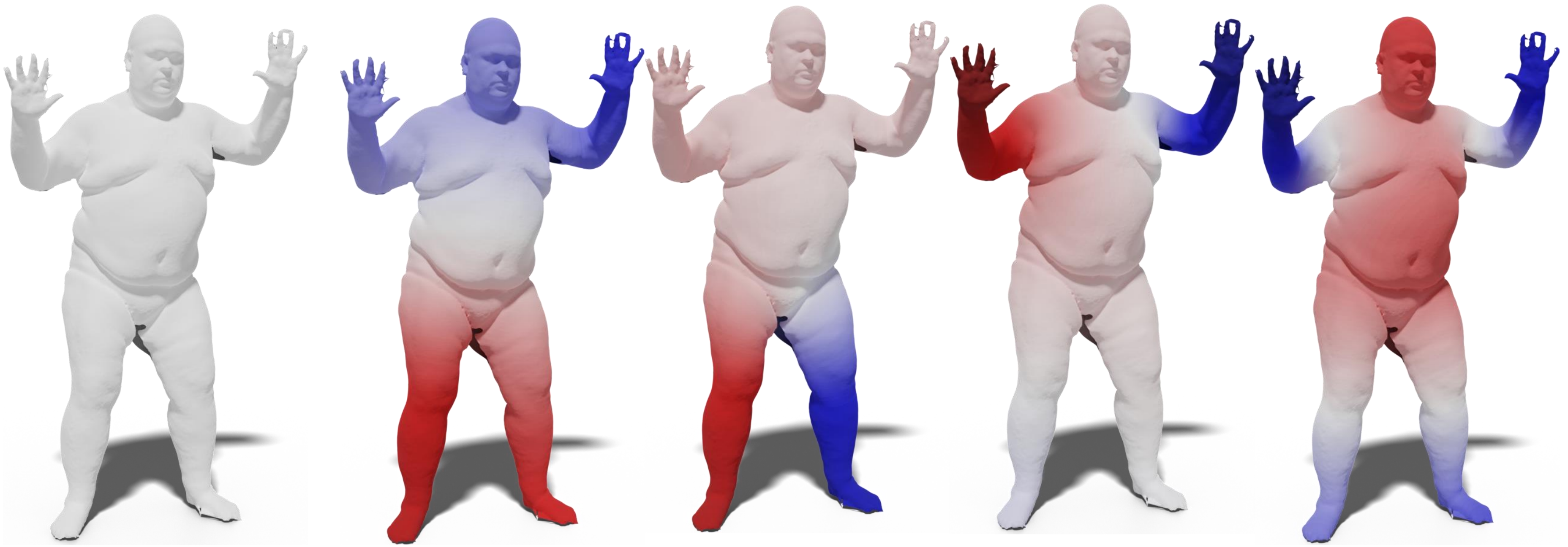
Figure 4. Contours of the 4th eigenfunction, computed from the Graph Laplacian (left) and cotangent weights (right) on an irregular mesh.

Key Features: Global operator defined by local relations, fully intrinsic, theoretical invariant to discretizations and isometries.

Levy B., «Lapalce-Beltrami eigenfunction towards an algorithm that understands geometry», 2006.


2,3
-1,2
0
-0,3
...
0,8
...

Example: Human Eigenfunctions

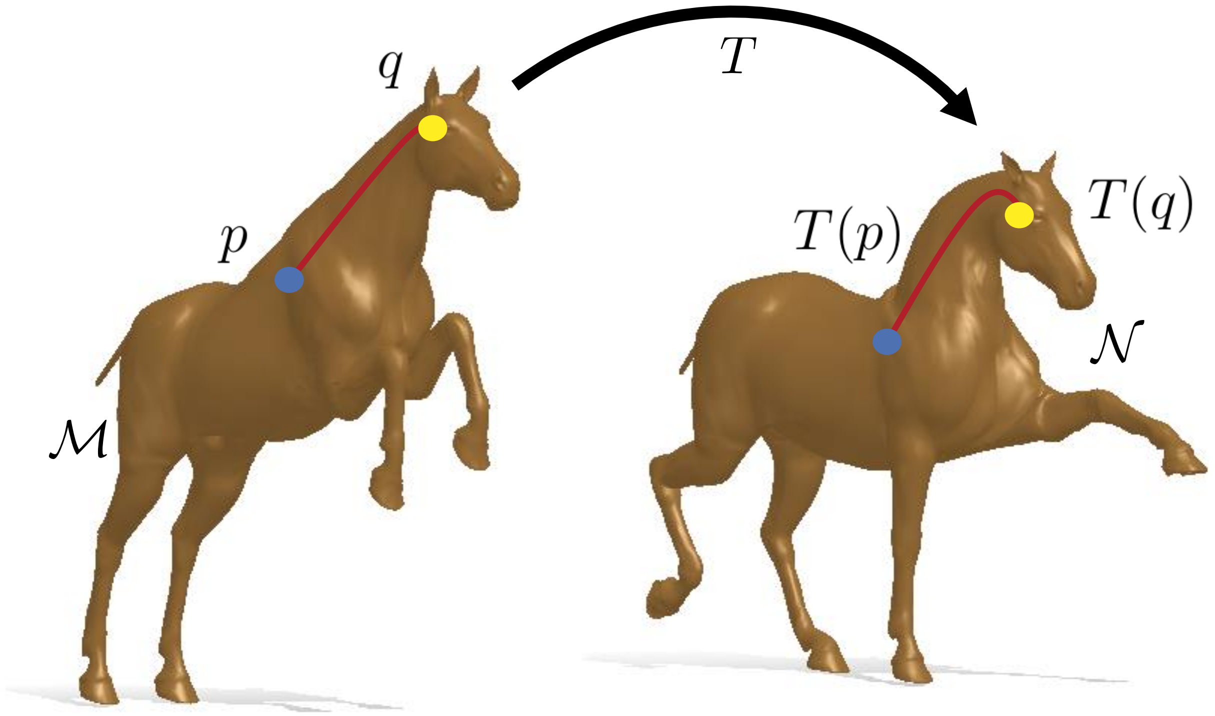


LBO and isometries

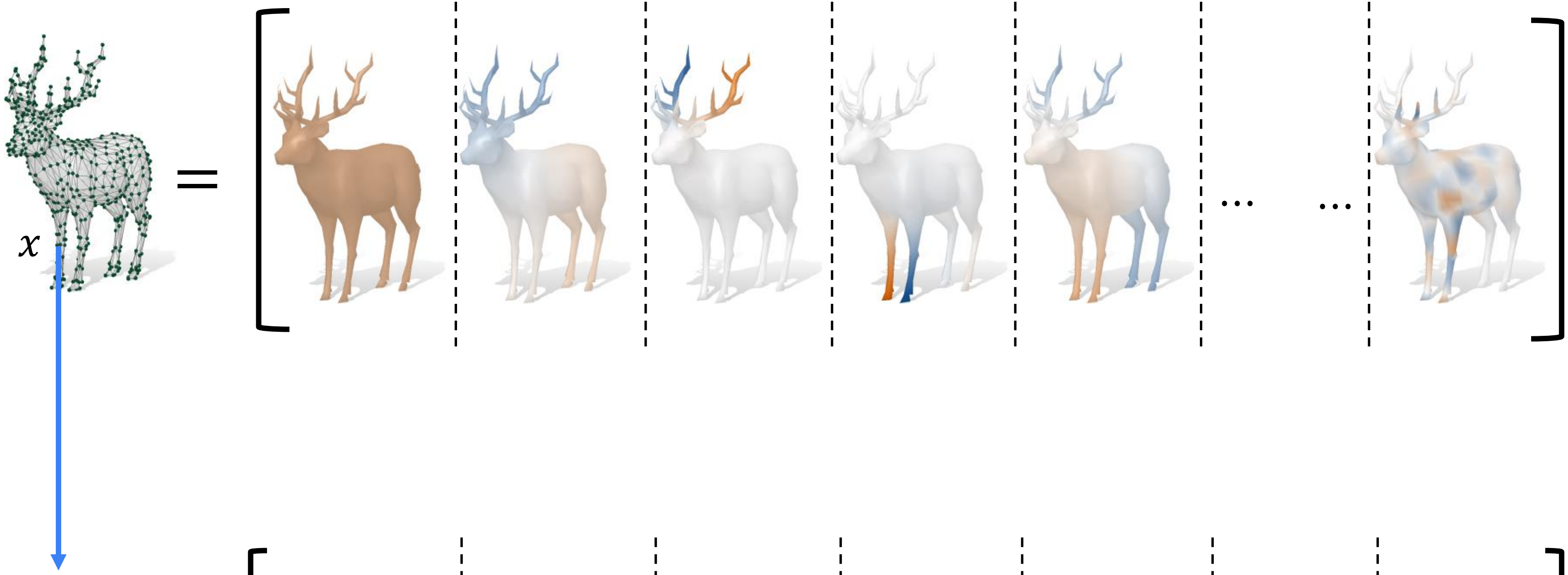
Two shapes are isometric $\Leftrightarrow$ their LBO agree

T is an isometry $\iff d_{\mathcal{M}}(p,q) = d_{\mathcal{N}}(T(p),T(q))\ \forall p,q \in \mathcal{M}$

Any quantity derived
from the LBO
is invariant to isometry



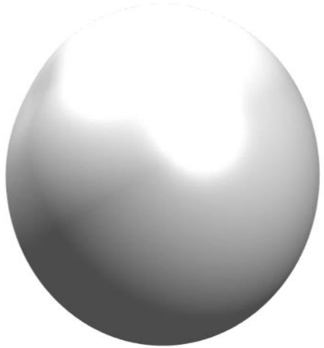
Spectral embedding

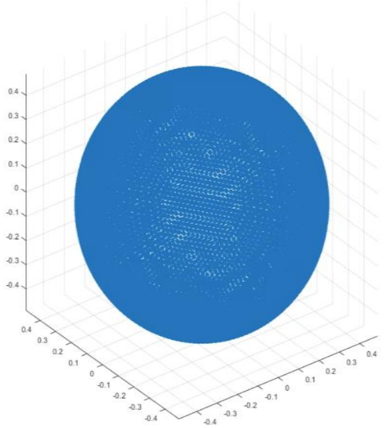


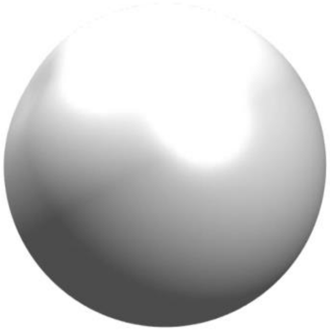
$$\boldsymbol{\Phi}(x) = \left[\ \phi_1(x) \ \vdots\ \phi_2(x) \ \vdots\ \phi_3(x) \ \vdots\ \phi_4(x) \ \vdots\ \phi_5(x) \ \vdots\ \cdots \quad \cdots \ \vdots\ \phi_k(x)\ \right]$$

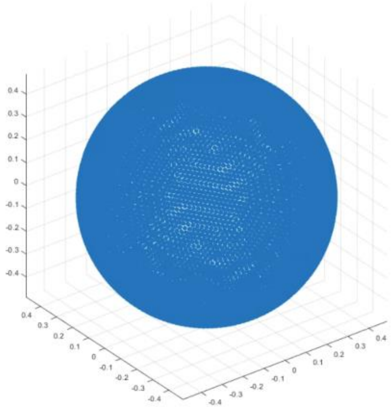
Example: sphere eigenfunctions

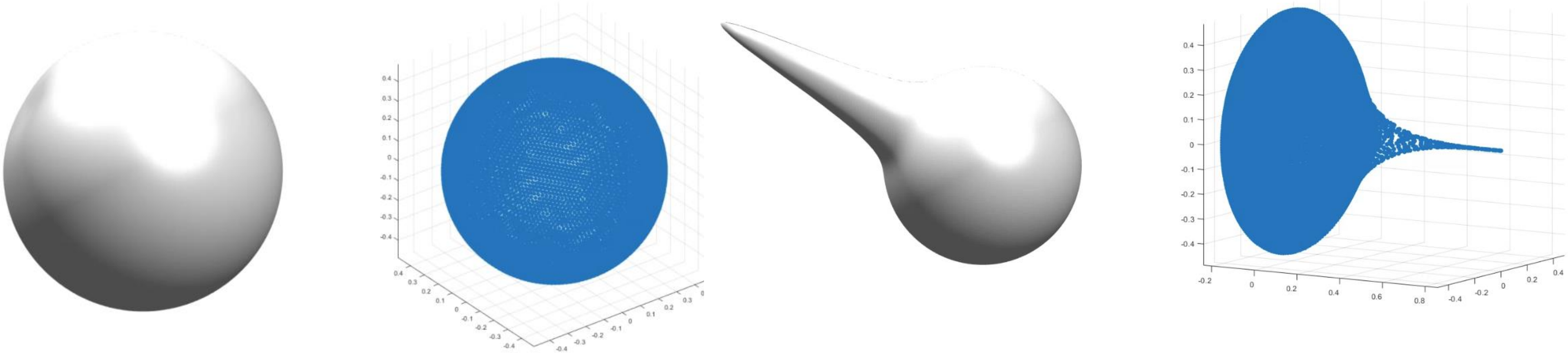
Consider 2,3 and 4 and plot them in the feature space

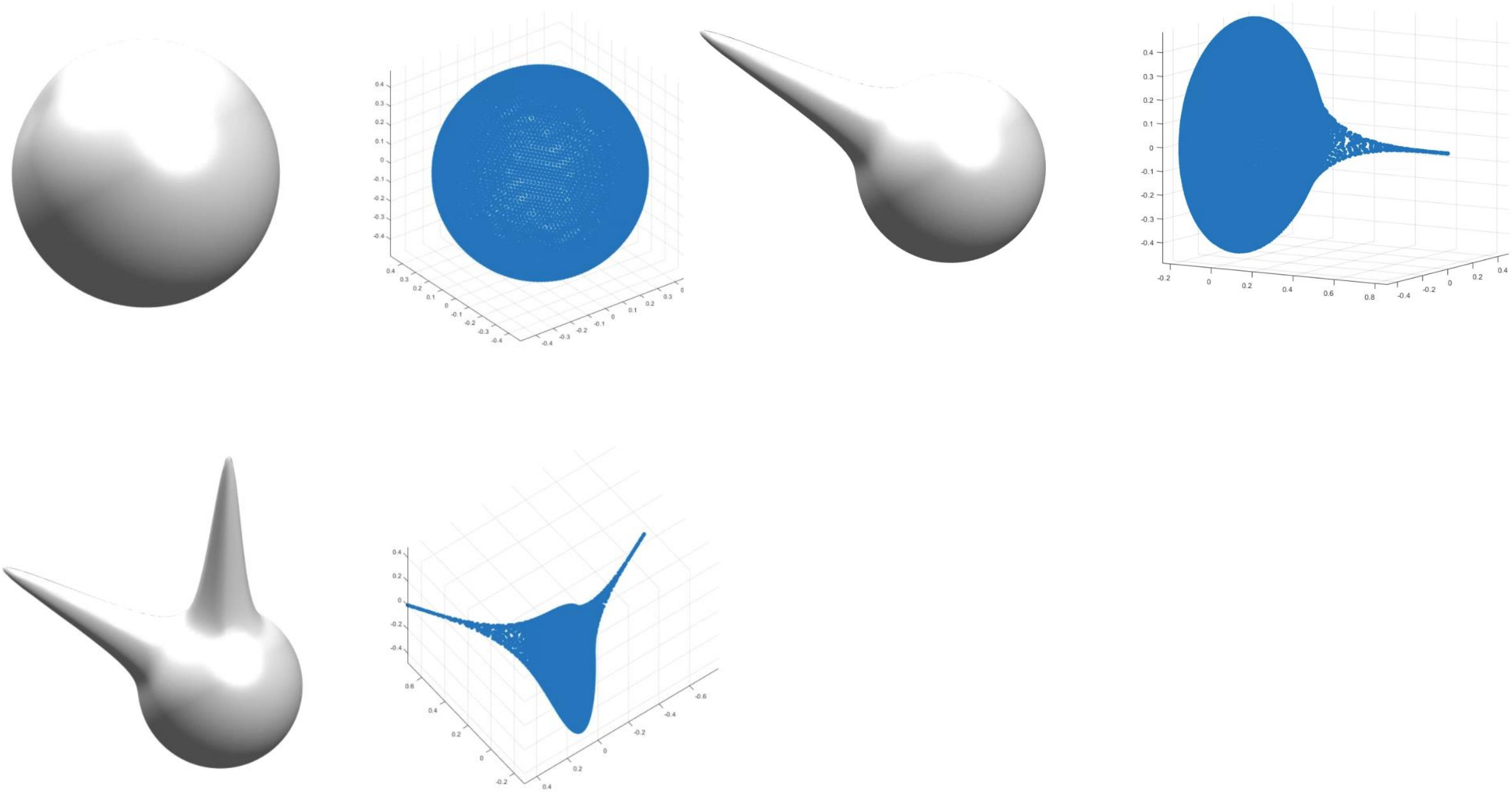


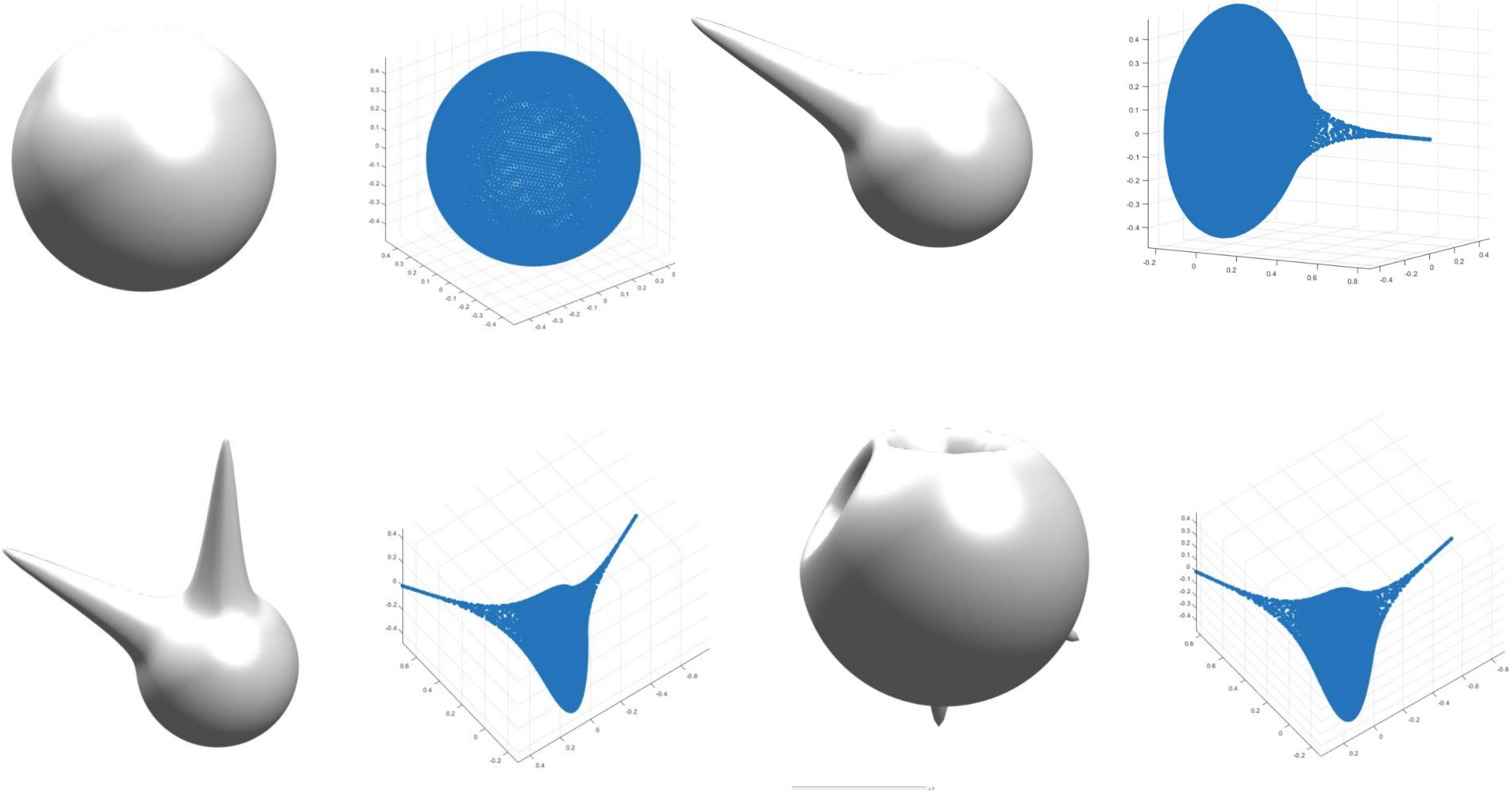





0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4


0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.2
0
0.2
0.4
0.6
0.8
-0.4
-0.2
0
0.2
0.4



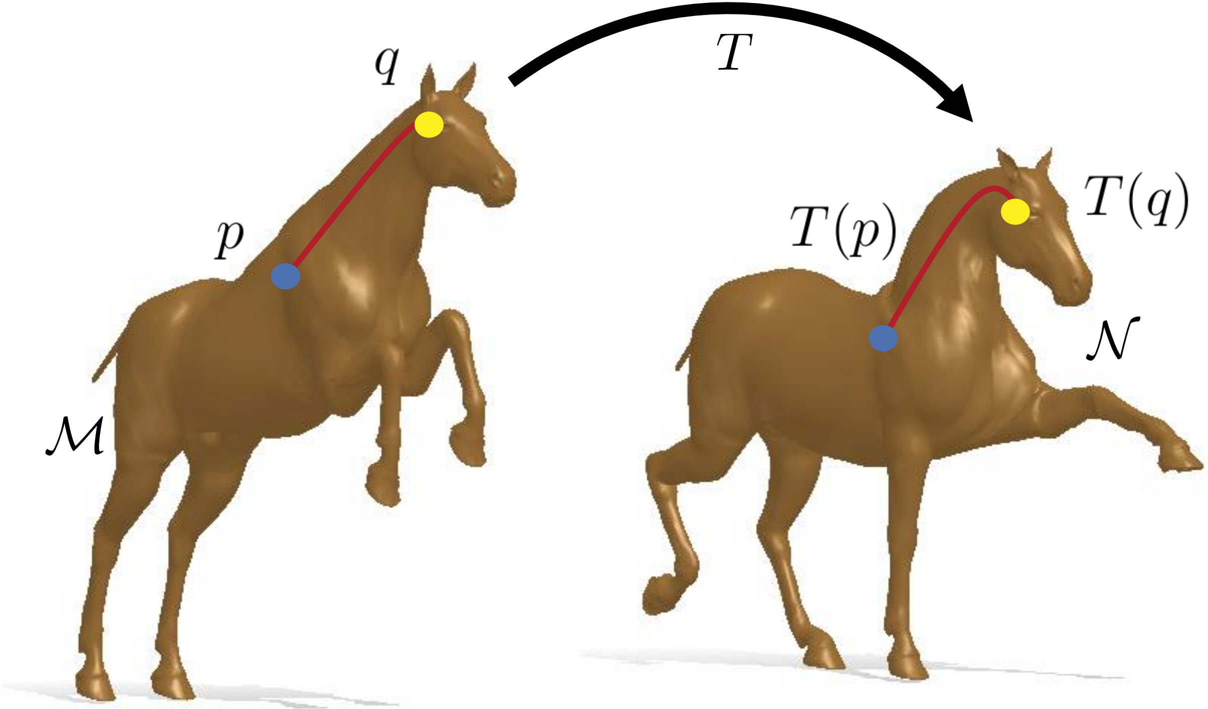


LBO and isometries

Two shapes are isometric $\Leftrightarrow$ their LBO agree

T is an isometry $\iff d_{\mathcal{M}}(p,q) = d_{\mathcal{N}}(T(p),T(q)) \; \forall p,q \in \mathcal{M}$

Any quantity derived
from the LBO
is invariant to isometry

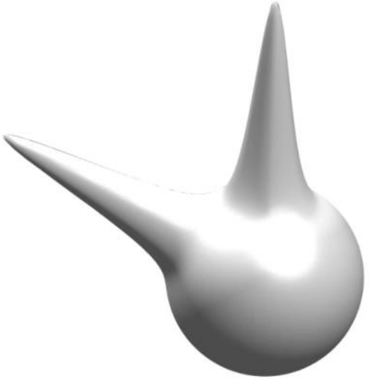


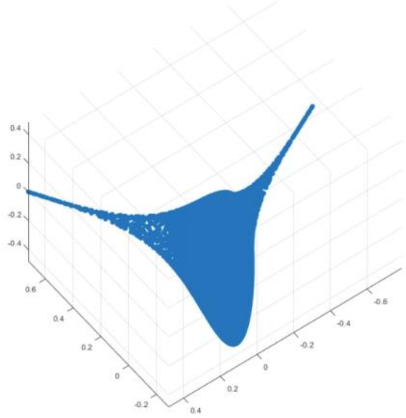
LBO and isometries

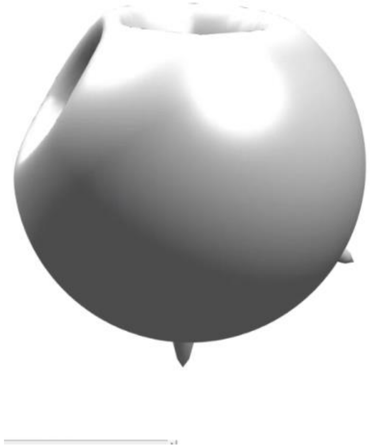
Two shapes are isometric $\Leftrightarrow$ their LBO agree

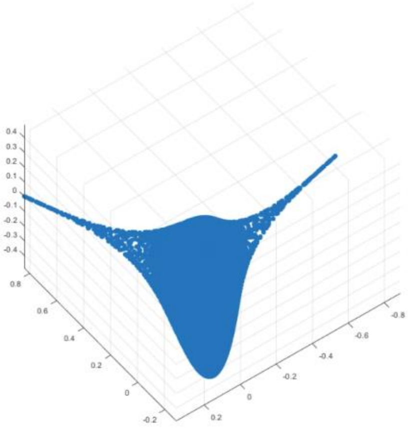
T is an isometry $\iff d_{\mathcal{M}}(p,q) = d_{\mathcal{N}}(T(p),T(q)) \ \forall p,q \in \mathcal{M}$

Any quantity derived from the LBO is invariant to isometry







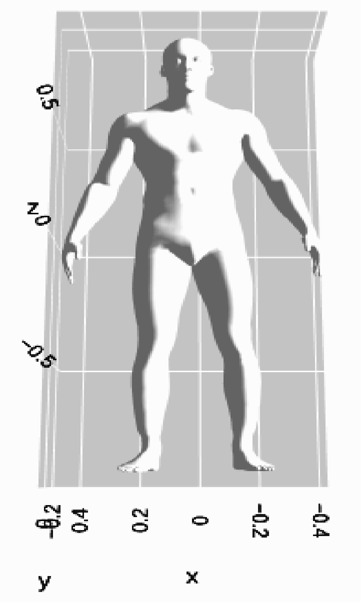


LBO and isometries

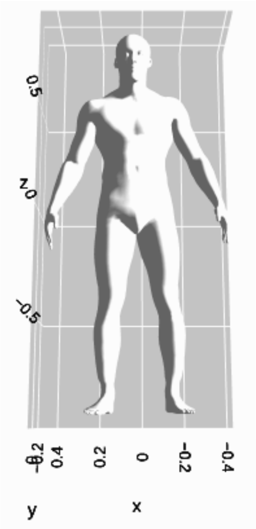
Two shapes are isometric ⟺ their LBO agree

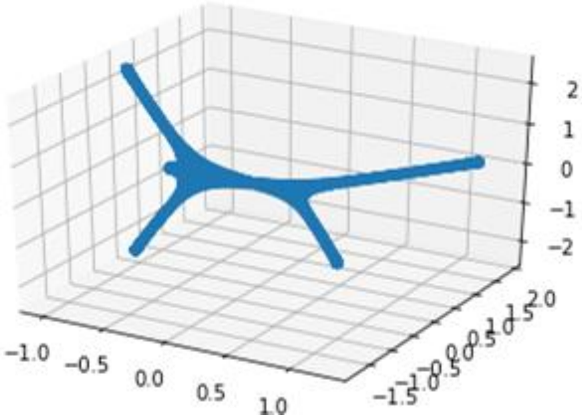
T is an isometry $\iff d_{\mathcal{M}}(p,q) = d_{\mathcal{N}}(T(p),T(q)) \ \forall p,q \in \mathcal{M}$

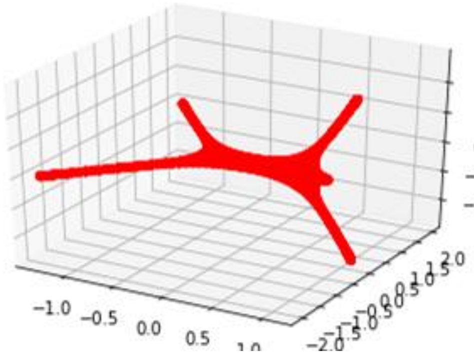
Any quantity derived from the LBO is invariant to isometry

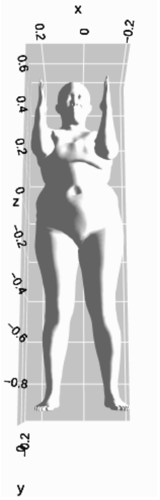


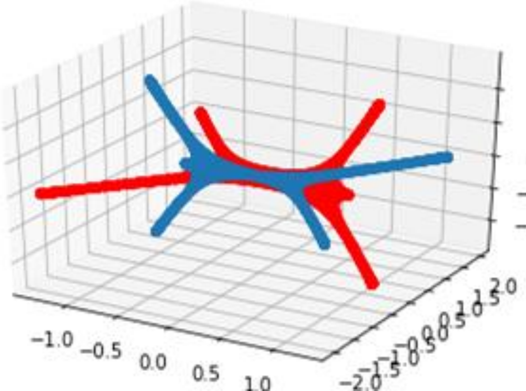
Example: Human Eigenfunctions (In the feature space)









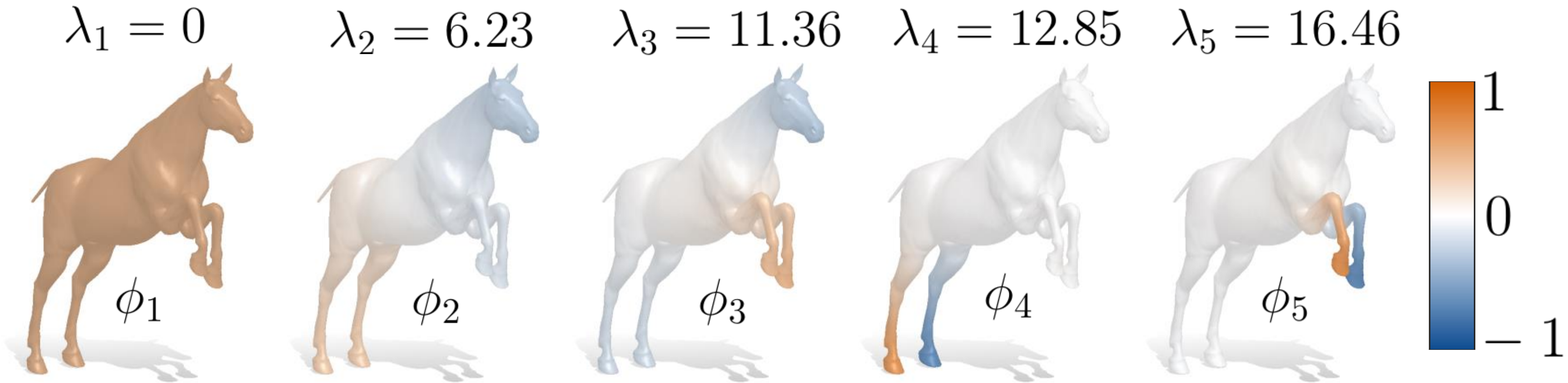


Functional Maps

Fourier basis

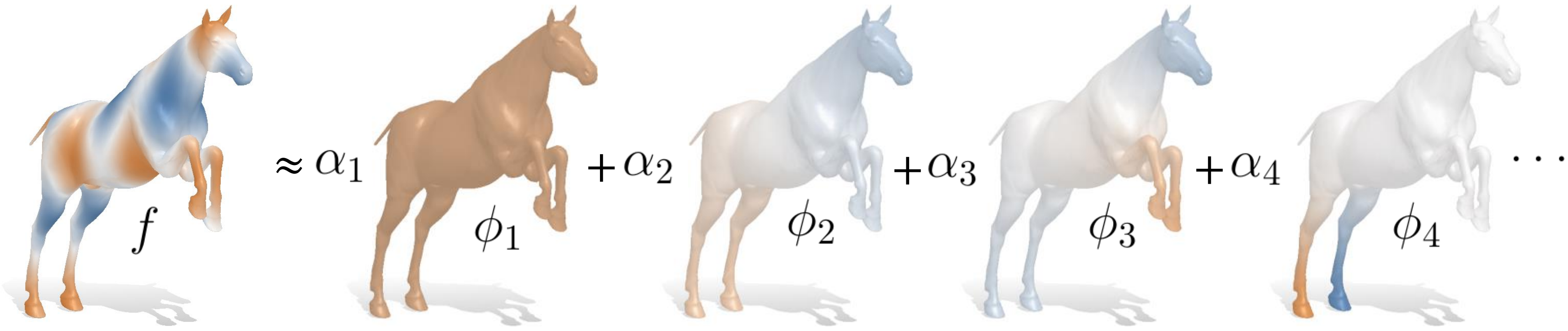
The eigenfunctions of the Laplace Beltrami Operator (LBO)

$$\mathbf{\Delta}_{\mathcal{M}}\phi_l = \lambda_l \phi_l \qquad \langle \phi_l, \phi_k \rangle_{\mathcal{M}} = \delta_l^k \qquad \lambda_l = \int_{\mathcal{M}} \|\nabla \phi_l\|^2 d\mu(x)$$



Laplace-beltrami eigenfunctions towards an algorithm that understands geometry, *Levy*, 2006

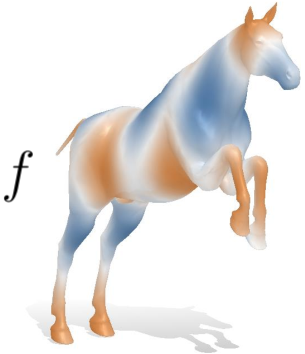
Fourier representation



Laplace-beltrami eigenfunctions towards an algorithm that understands geometry, *Levy*, 2006

Synthesis and analysis

Given a signal: f



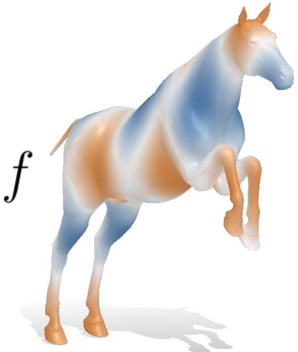
The analysis: $$\alpha_l = \langle f, \phi_l \rangle_{\mathcal{M}} = \int_{\mathcal{M}} f(x)\phi_l(x)d\mu(x)$$

The synthesis: $$f = \sum_{l=1}^{n} \alpha_l \phi_l = \sum_{l=1}^{n} \langle f, \phi_l \rangle_{\mathcal{M}} \phi_l \approx \sum_{l=1}^{k<n} \alpha_l \phi_l$$

Laplace-beltrami eigenfunctions towards an algorithm that understands geometry, *Levy*, 2006

Synthesis and analysis: dicrete setting

Given a signal: f



$$\langle f, g \rangle_{\mathcal{M}} = f^{\top} \mathbf{\Omega}_{\mathcal{M}} g$$

$$\mathbf{\Phi}_{\mathcal{M}} = [\phi_1,\ \phi_2,\ \ldots\ \phi_{k-1},\ \phi_k]$$

$$\mathbf{\Phi}^{\dagger}_{\mathcal{M}}\ s.t.\ \mathbf{\Phi}^{\dagger}_{\mathcal{M}} \mathbf{\Phi}_{\mathcal{M}} = I$$

$$\mathbf{\Phi}_{\mathcal{M}}\ s.t.\ \mathbf{\Phi}^{\top}_{\mathcal{M}} \mathbf{\Omega}_{\mathcal{M}} \mathbf{\Phi}_{\mathcal{M}} = I$$

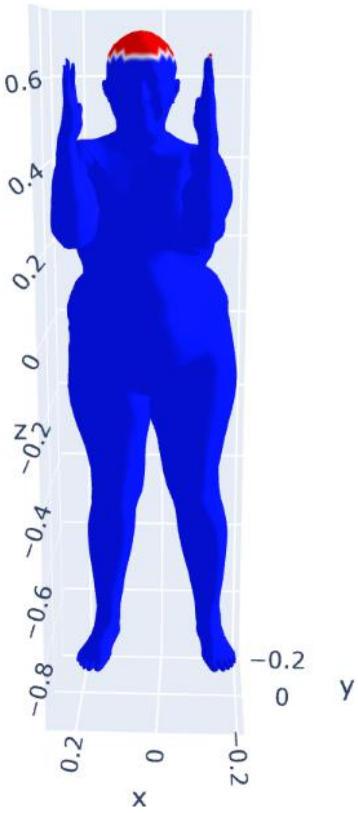
$$\mathbf{\Phi}^{\dagger}_{\mathcal{M}} = \mathbf{\Phi}^{\top}_{\mathcal{M}} \mathbf{\Omega}_{\mathcal{M}}$$

The analysis: $\boldsymbol{\alpha} = \mathbf{\Phi}^{\dagger}_{\mathcal{M}} f$

The synthesis: $f = \mathbf{\Phi}_{\mathcal{M}} \boldsymbol{\alpha} = \mathbf{\Phi}_{\mathcal{M}} \mathbf{\Phi}^{\dagger}_{\mathcal{M}} f$

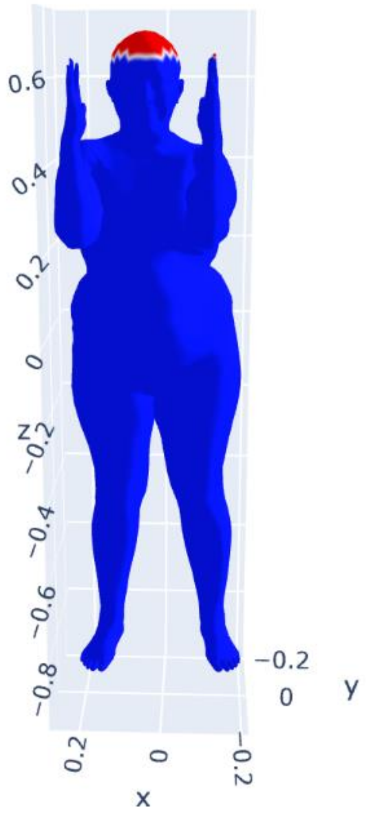
Laplace-beltrami eigenfunctions towards an algorithm that understands geometry, *Levy*, 2006

Application: Signal smoothing

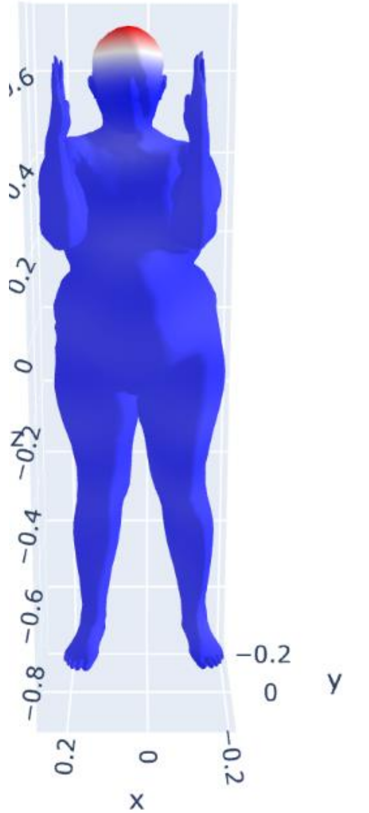


f

Application: Signal smoothing

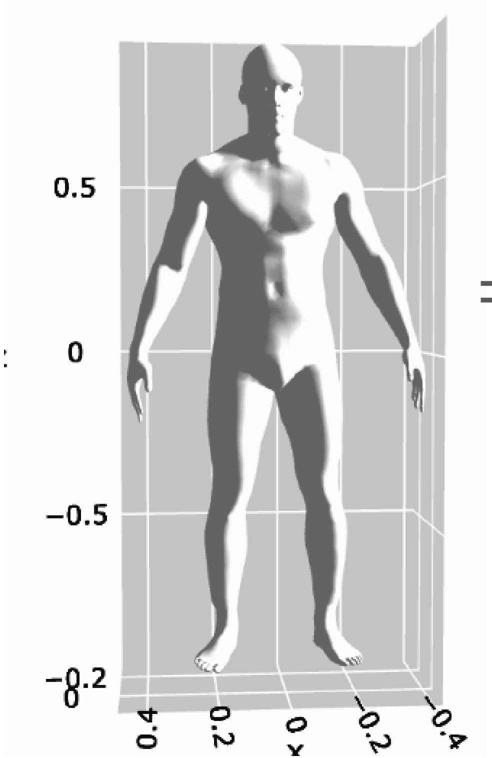


$$f$$



$$\mathbf{\Phi}(f^T\mathbf{\Phi})^T$$

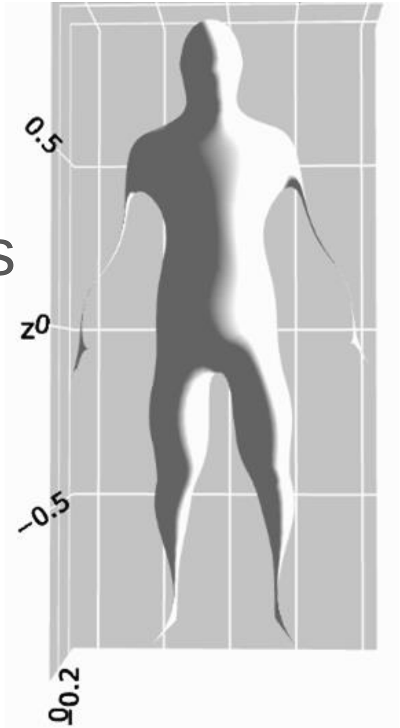
Application: Signal smoothing



f

f

= Coordinates



$\boldsymbol{\Phi}(f^T\boldsymbol{\Phi})^T$

Oh no, I am lost!

Oh no, I am lost!

DON'T
PANIC

Oh no, I am lost!

N
Vertices

Given a
mesh

→ Δ

a matrix

DON'T
PANIC

Oh no, I am lost!

N

Vertices

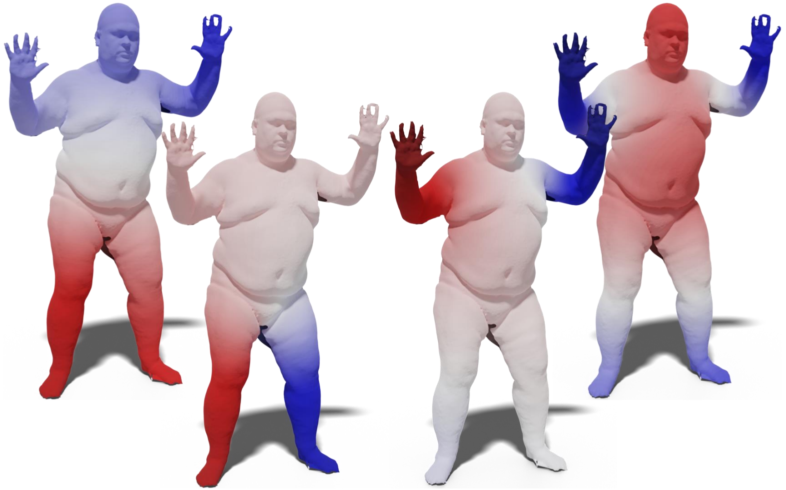
Given a mesh

Δ

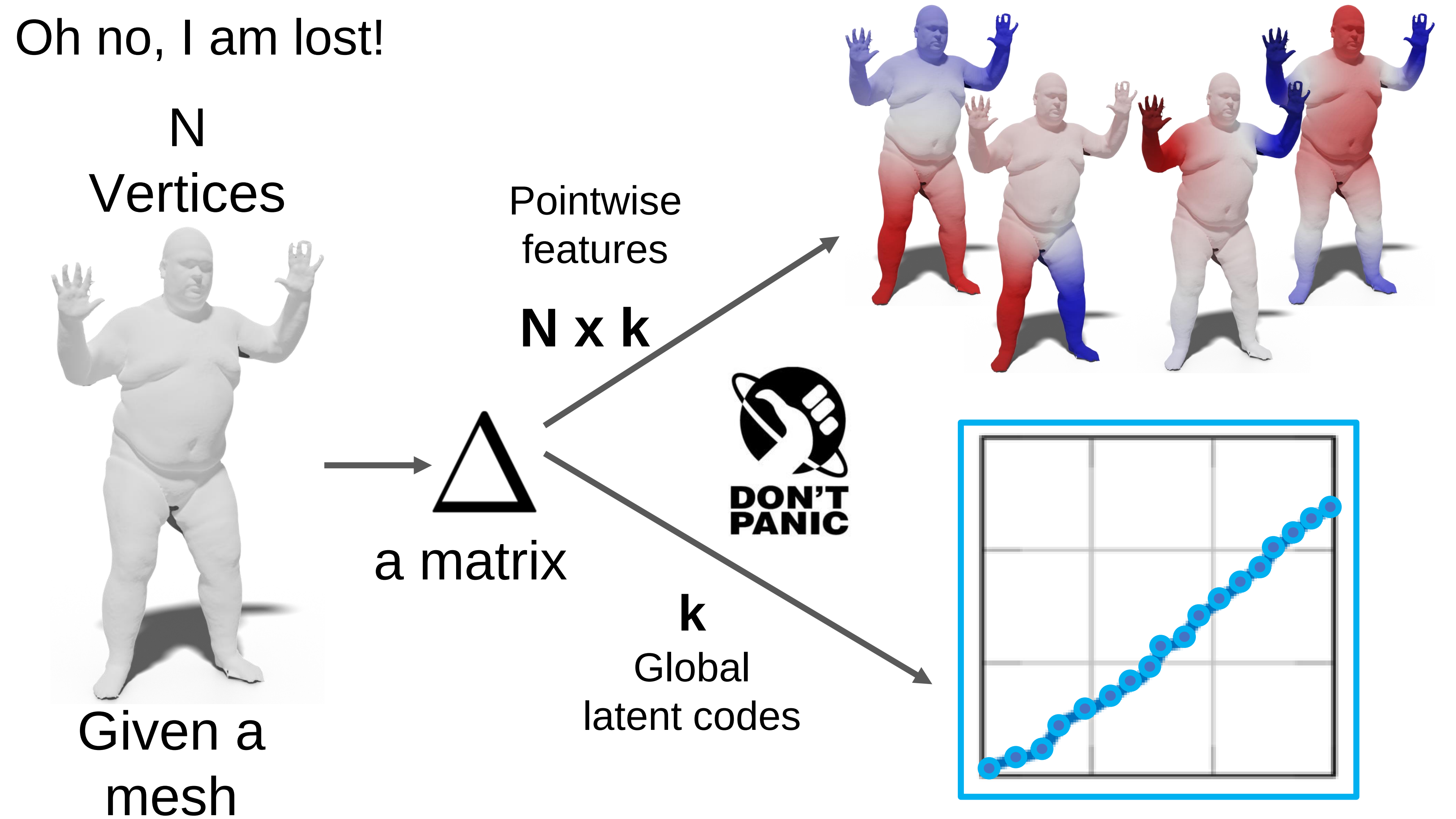
a matrix

Pointwise features

N x k



DON'T PANIC


Oh no, I am lost!
N
Vertices
Given a
mesh
a matrix
Pointwise
features
N x k
DON'T
PANIC
k
Global
latent codes

Oh no, I am lost!

N
Vertices

Given a
mesh

a matrix

Pointwise
features

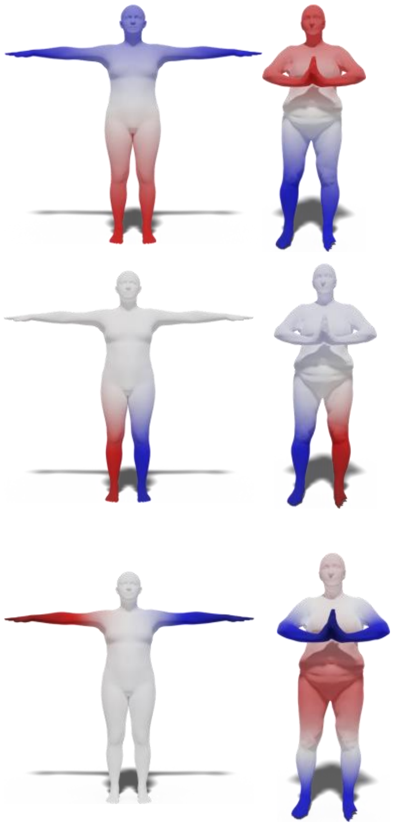
N x k

k
Global
latent codes

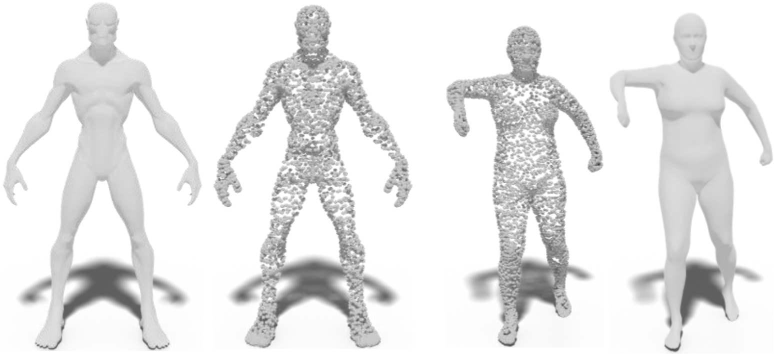
DON'T
PANIC

Main issues

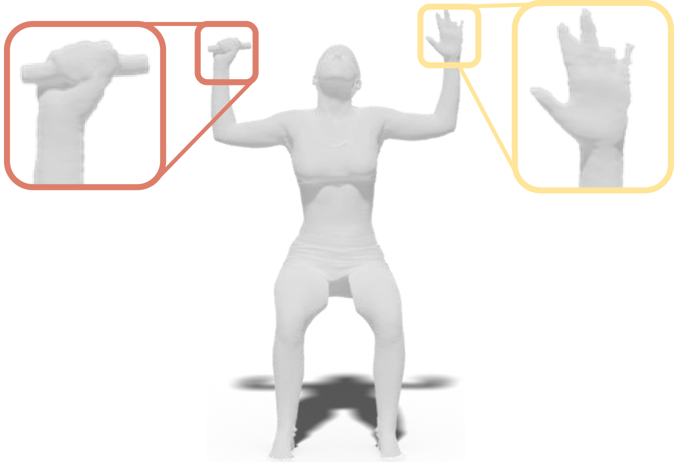
a) Topological noise



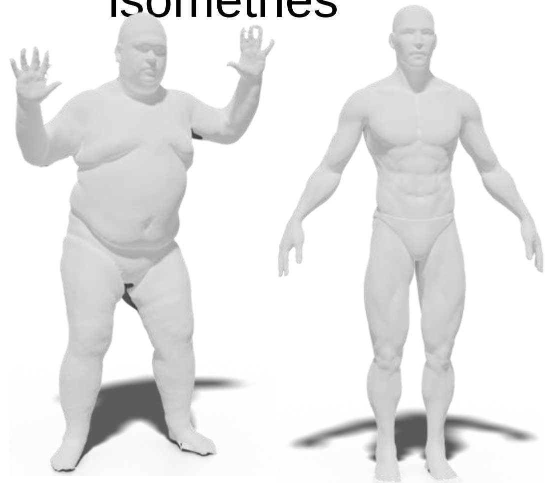
b) Pointclouds



c) Clutter\Partiality



c) Heavy non-isometries



Summary so far:

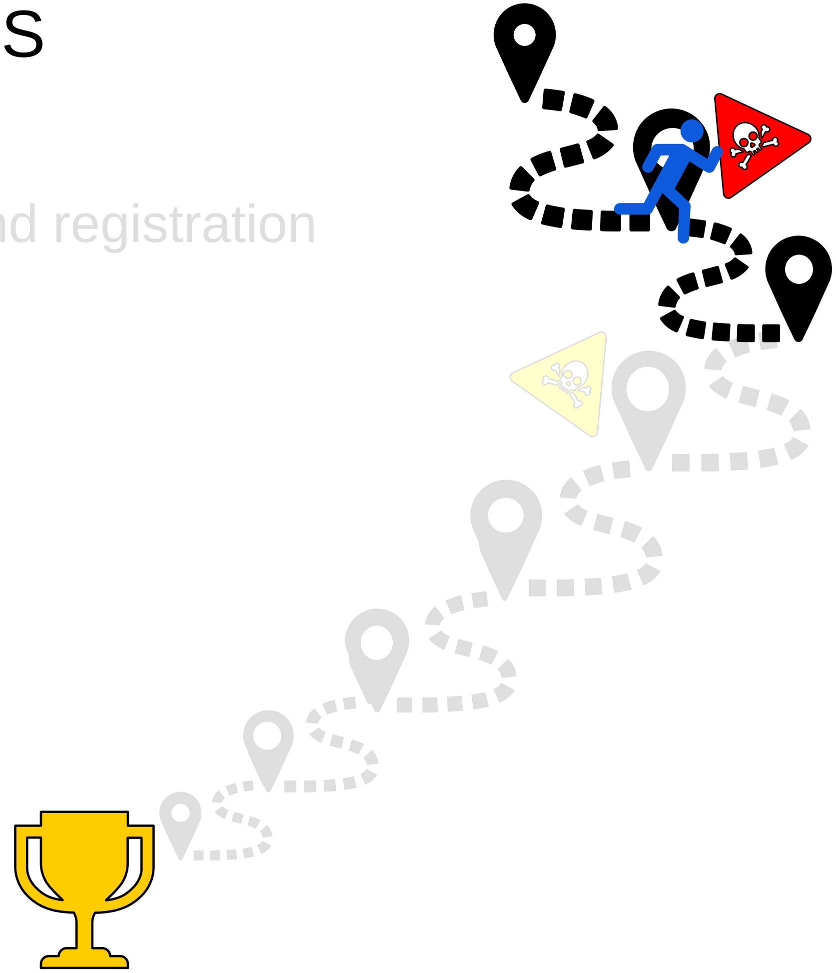
- **Graphs**: general object, with nodes and a connectivity
- **Functions on Graphs**: vectors (scalar value for each node)
- **Euclidean analysis tools**: Fourier analysis
- **Fourier basis:** Eigenfunctions of the Laplacian
- **Discretization of the Laplacian:** a sparse square matrix
- **Discrete Fourier basis:** Eigenvectors of the Laplacian
- **This is general enough:** it works on graph and meshes (point clouds)
- **Nice theoretical properties:** isometries, near isometries, low-pass filtering
- **Connection with Fourier Analysis:** Non-Euclidean signal processing
- **Many applications:** Classification, Segmentation, Positional encoding
- **Weakness:** many real case scenarios ruin the theoretical premises

Next Goal

use it in 3D non-rigid registration

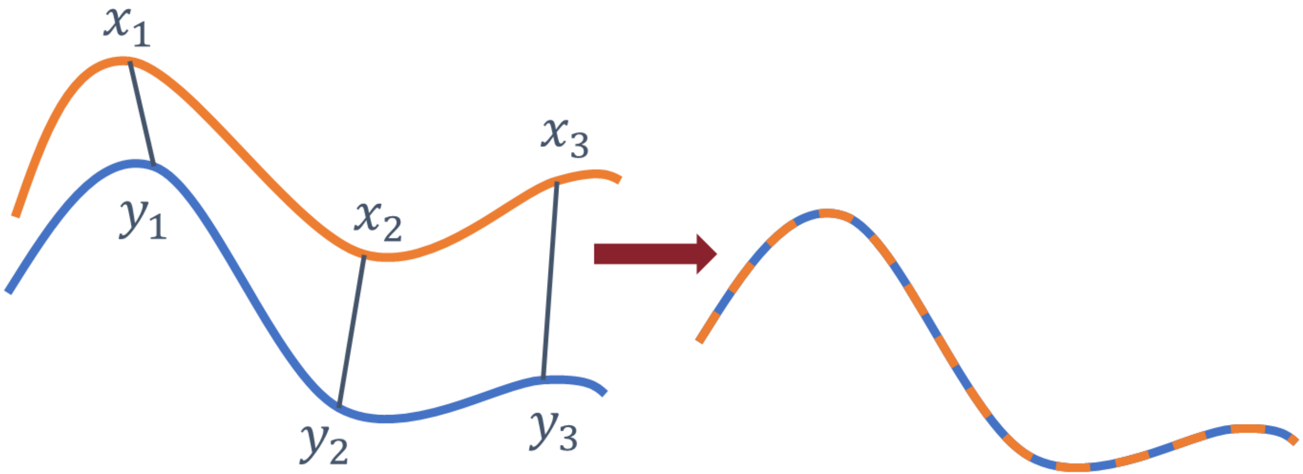
AXIOMATIC APPROACHES

1. Introduction: 3D Non-Rigid shapes and registration
2. Spectral representation
3. **Axiomatic approaches**
4. Functional maps
5. Learning on geometric data
6. Learning–based Functional maps
7. Other learning-based approaches
8. Transformers



ICP (1D case)

The solution is a rigid transformation $\mathbb{R}^2 \rightarrow \mathbb{R}^2$



ICP approach = iterate alternating:
(1) finding correspondences;
(2) finding optimal transformation.

Slide credits to M. Ovsjanikov

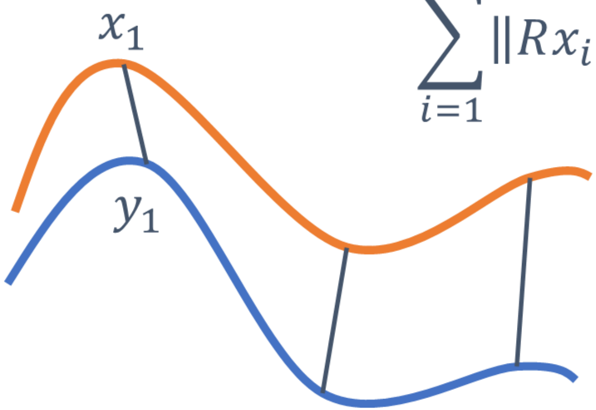
ICP (1D case)

ICP = Iterative Closest Point

Given 2 shape $\mathcal{X}$ and $\mathcal{Y}$
Iterate (a stop criteria is satisfied):

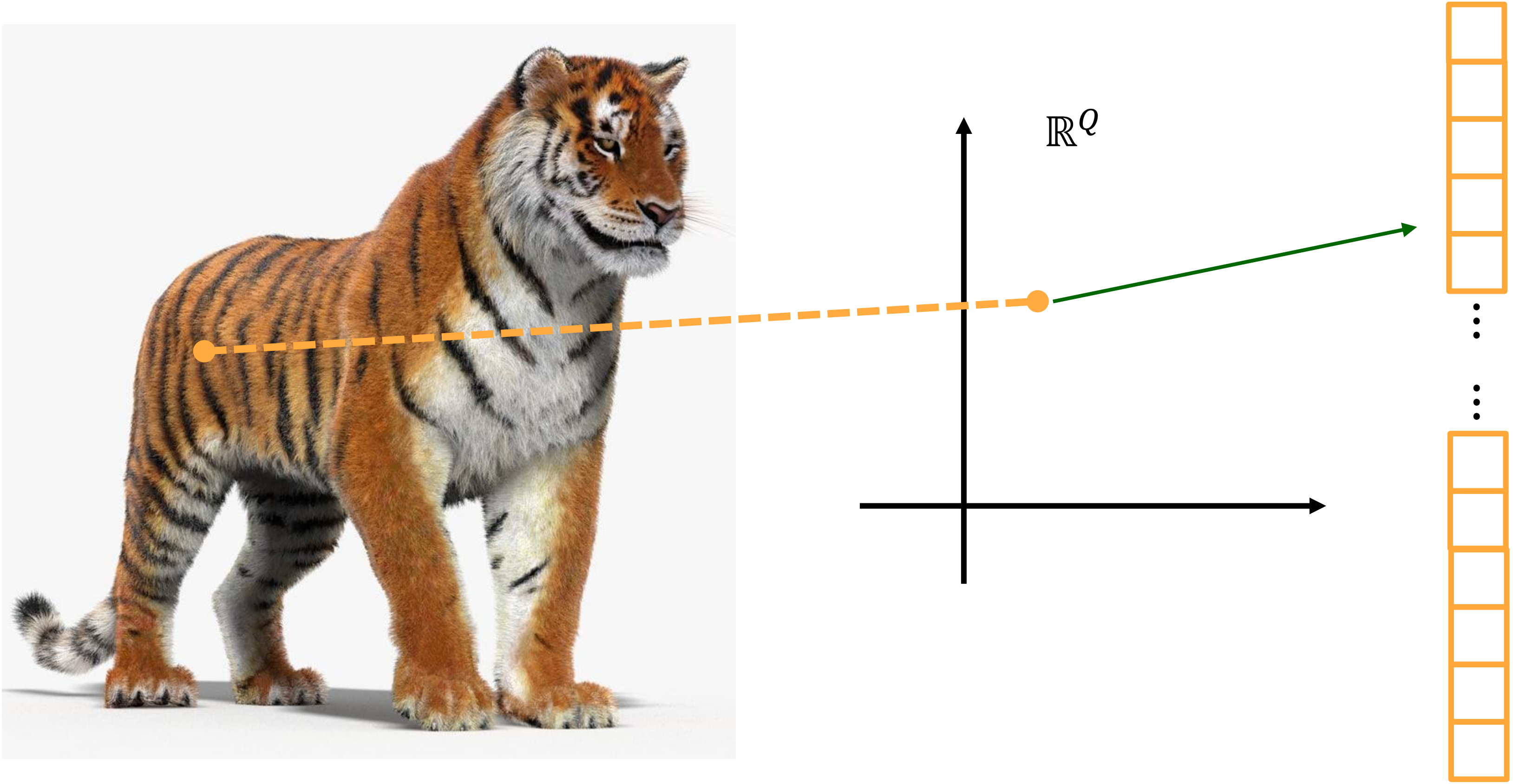
1. $\forall x_i \in \mathcal{X}$ find the **nearest** neighbor $y_i \in \mathcal{Y}$;
2. Find R optimal rotation and t translation s.t.:

$$\sum_{i=1}^{N} \|Rx_i + t - y_i\|_2^2$$

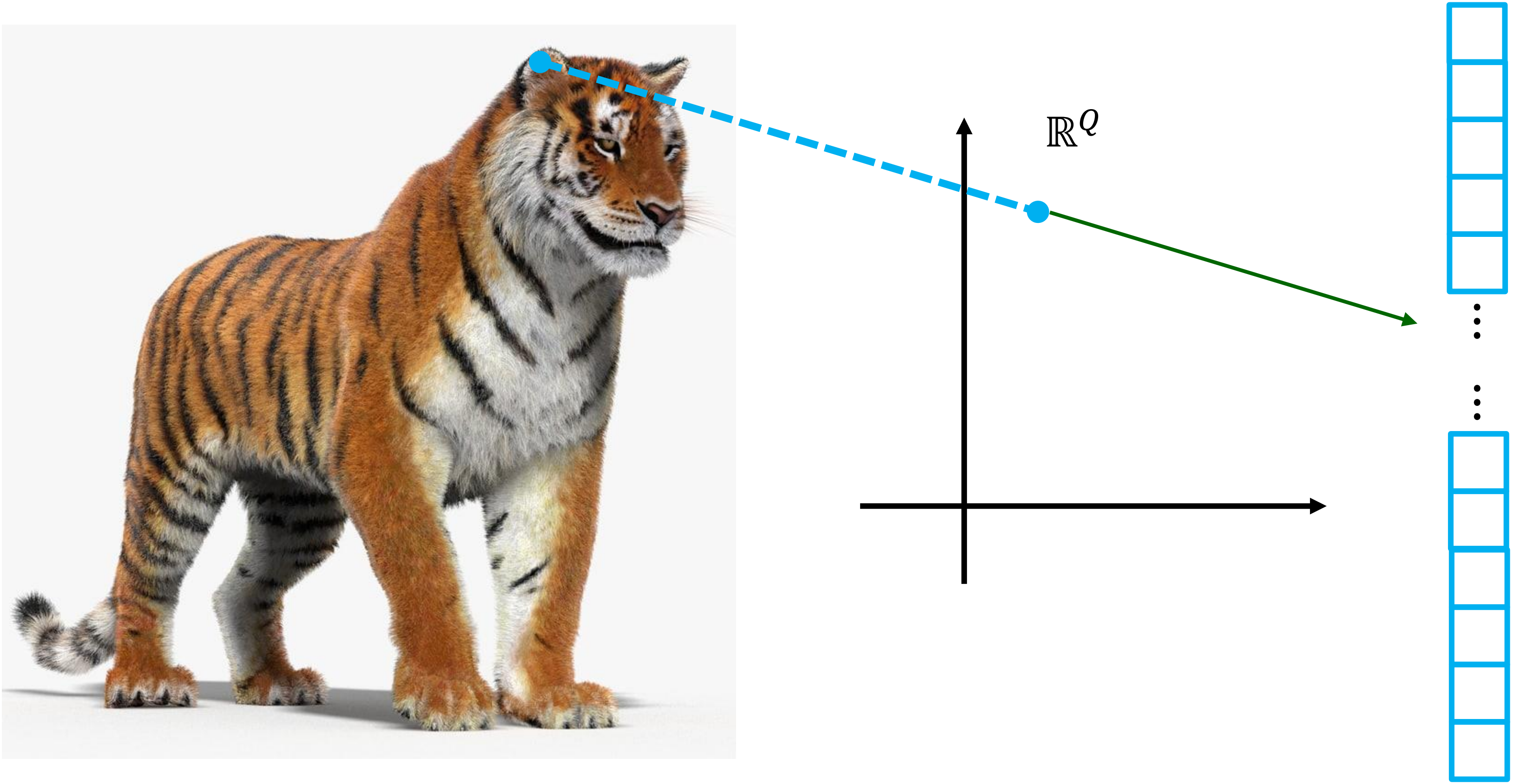


Slide credits to M. Ovsjanikov

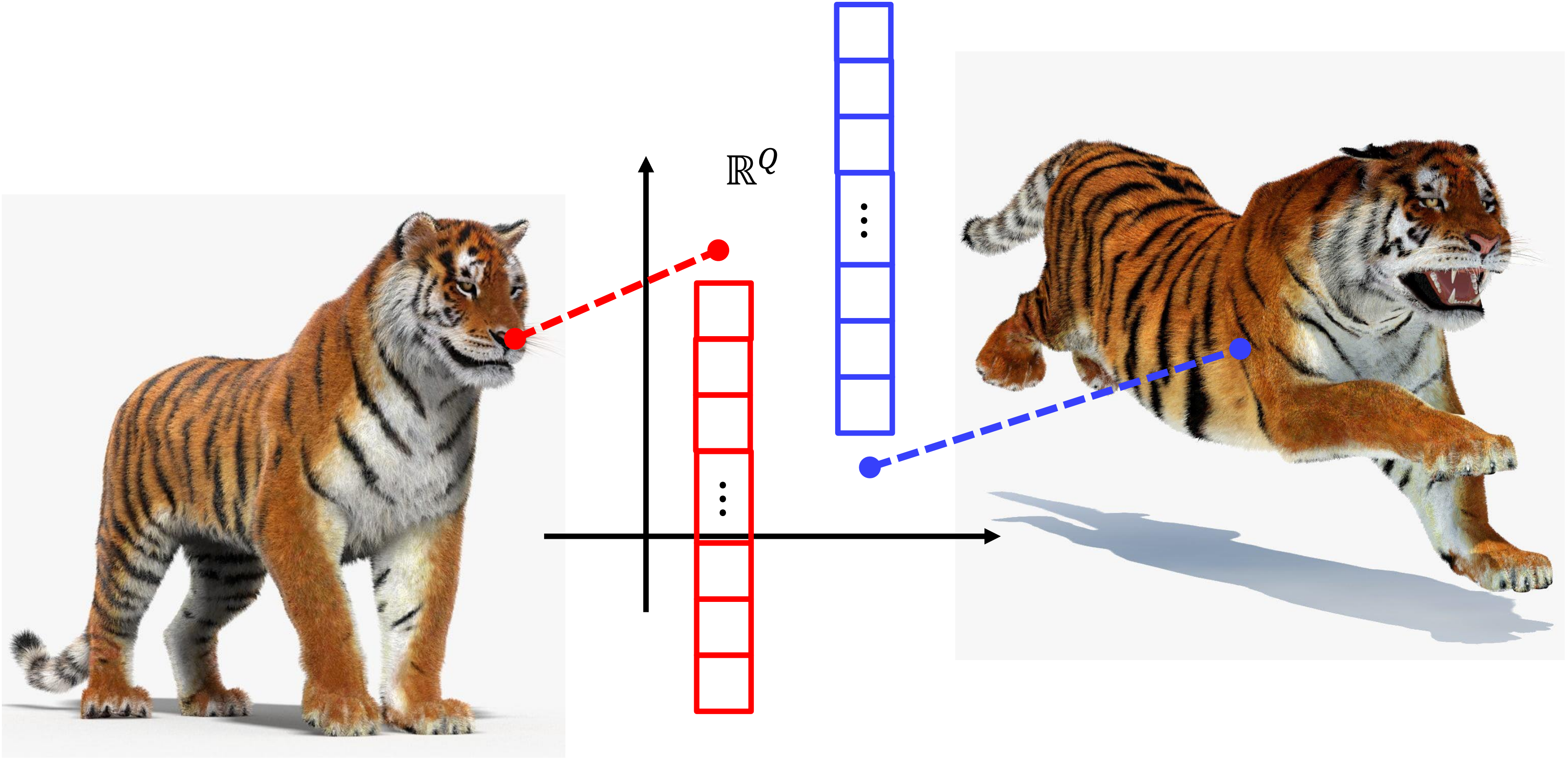
Features-based



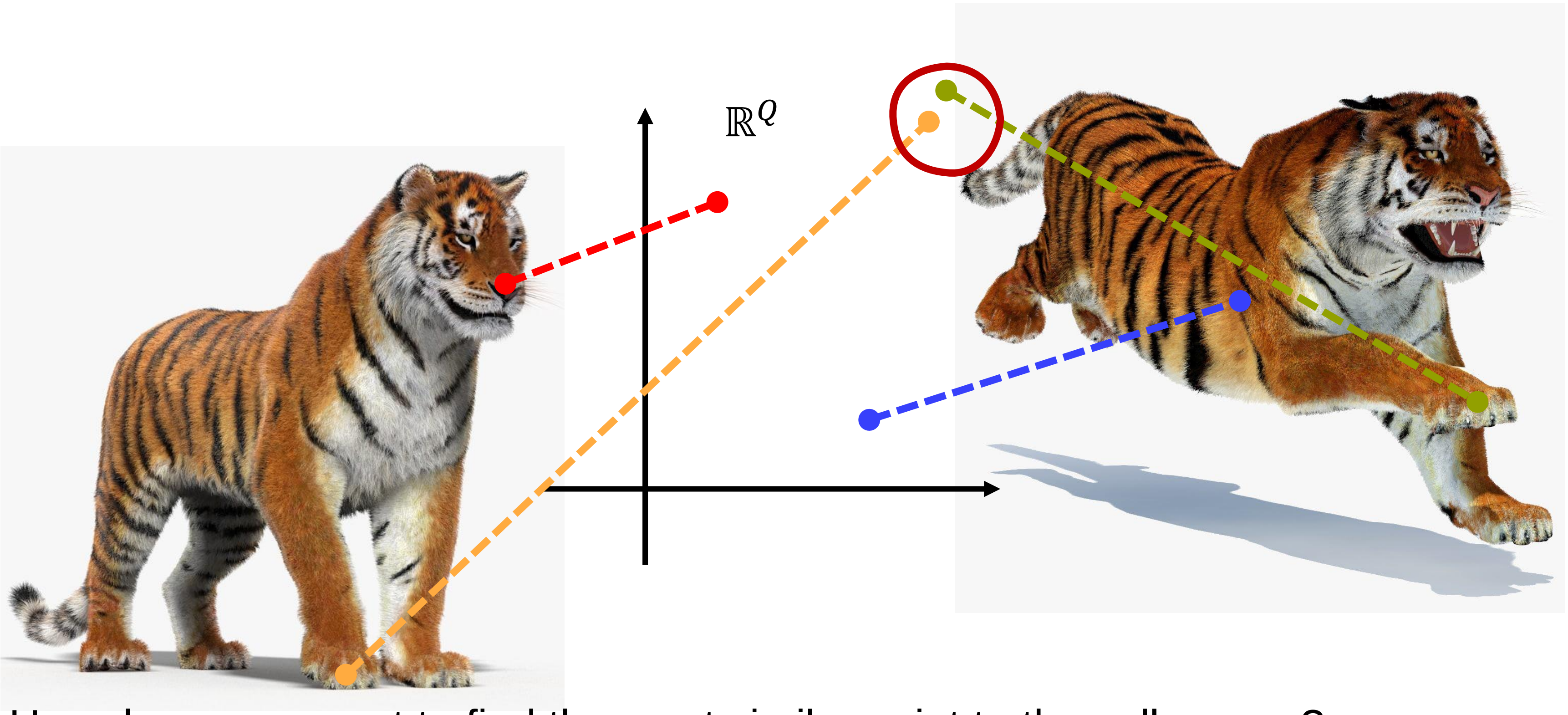
Features-based



Features-based

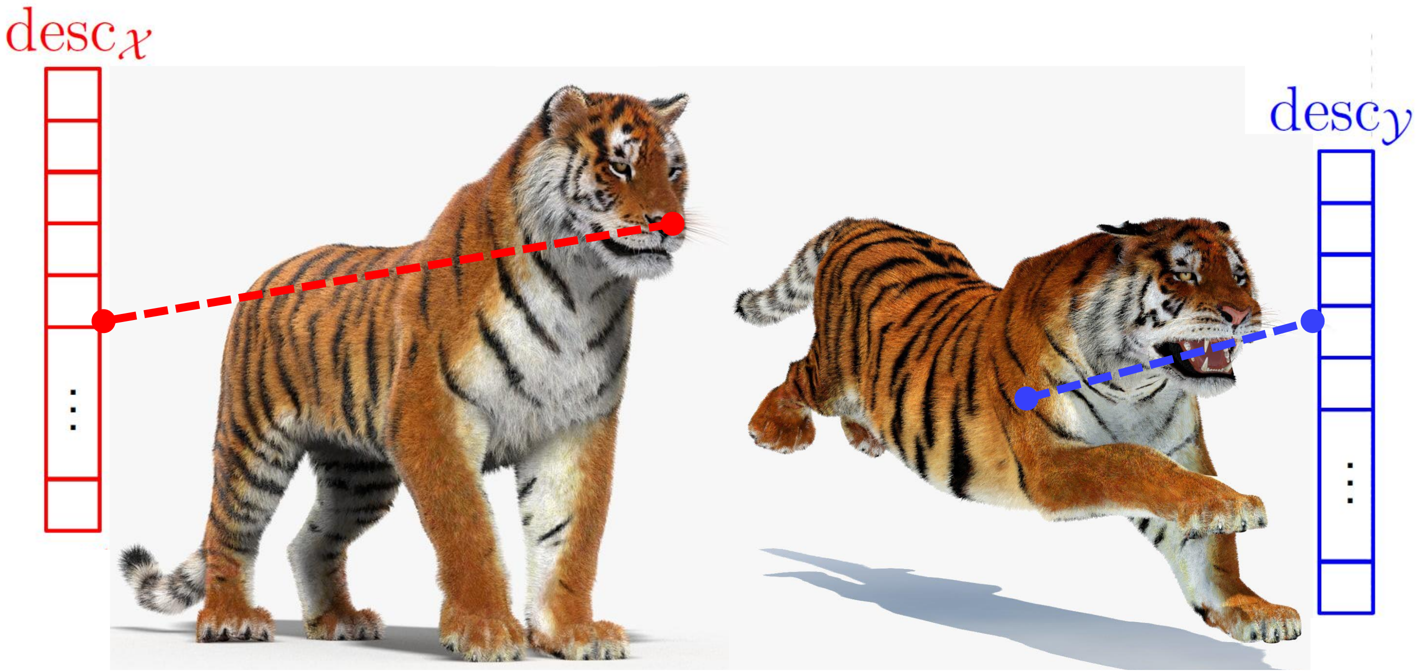


Features-based



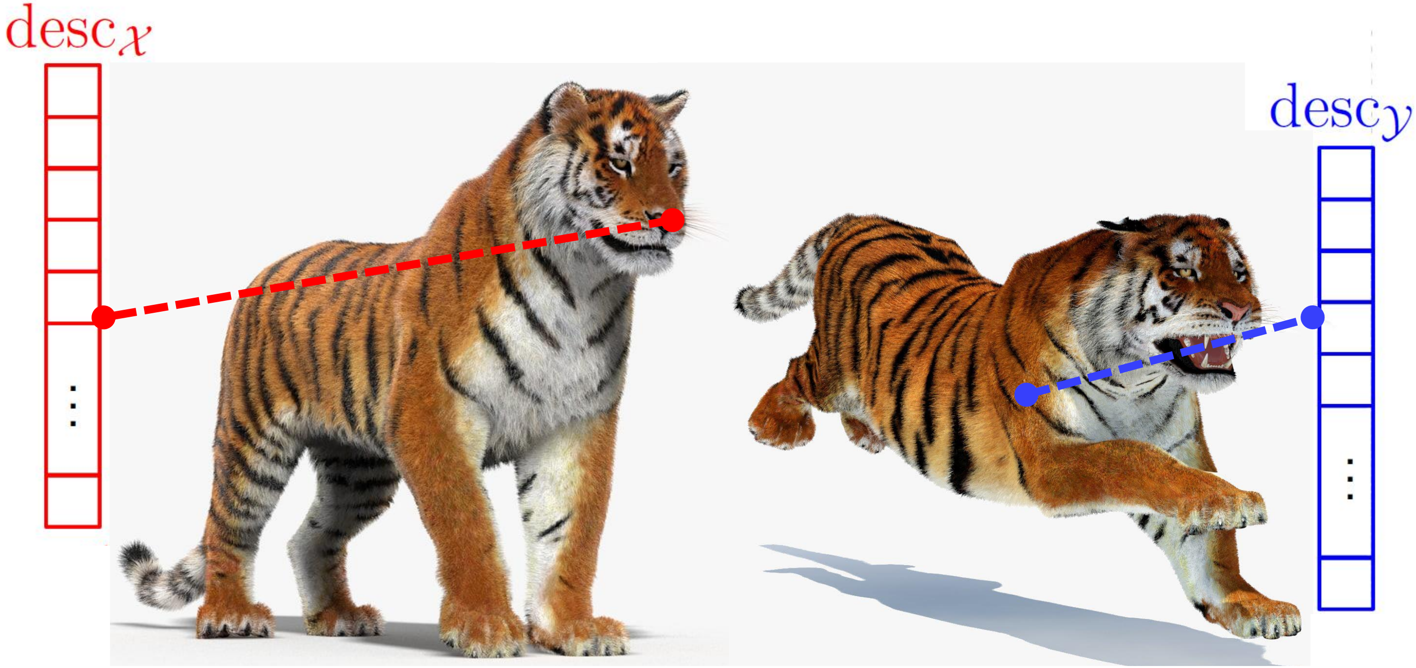
How do you suggest to find the most similar point to the yellow one?

Features-based



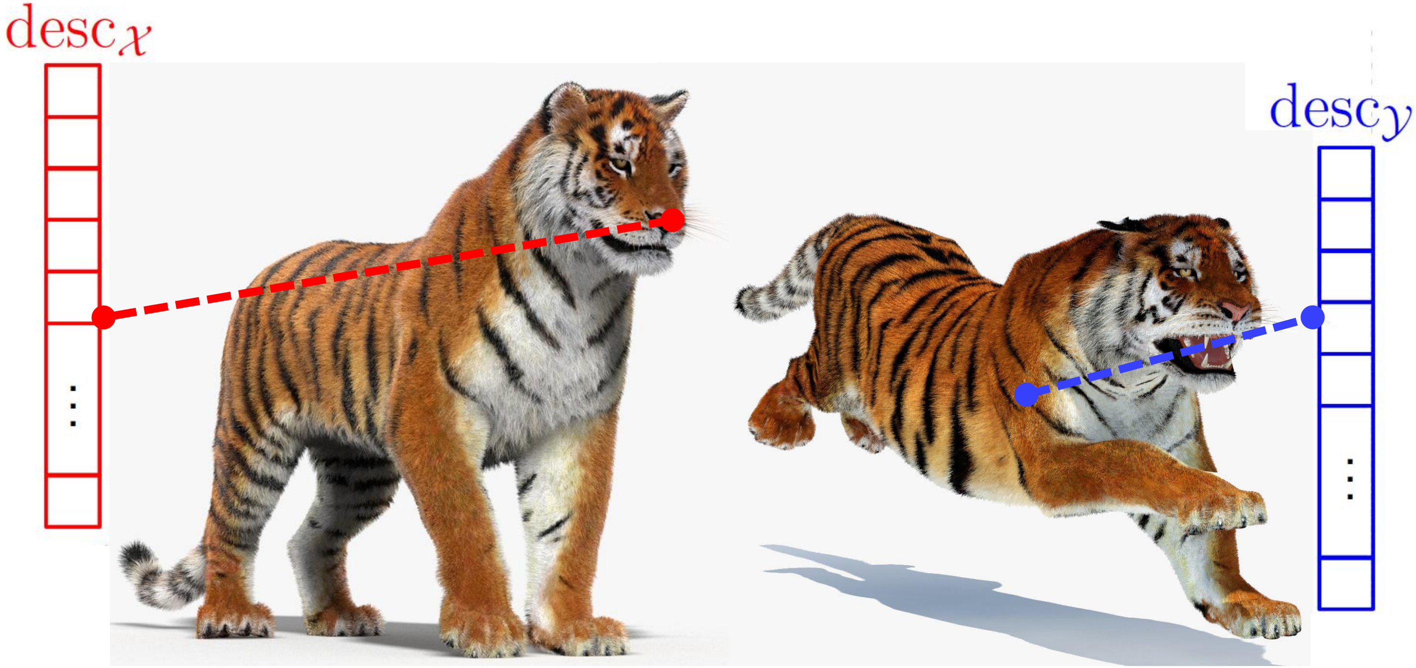
$$\text{Distance} = \mathcal{D}(desc_{\mathcal{X}}, desc_{\mathcal{Y}})$$

Features-based



$$\text{Distance} = \left\| desc_{\mathcal{X}} - desc_{\mathcal{Y}} \right\|_2$$

Features-based



$$\Pi(x) = y = \underset{y \in \mathcal{Y}}{argmin} \left\| desc_{\mathcal{X}}(x) - desc_{\mathcal{Y}}(y) \right\|_2$$

Desired properties

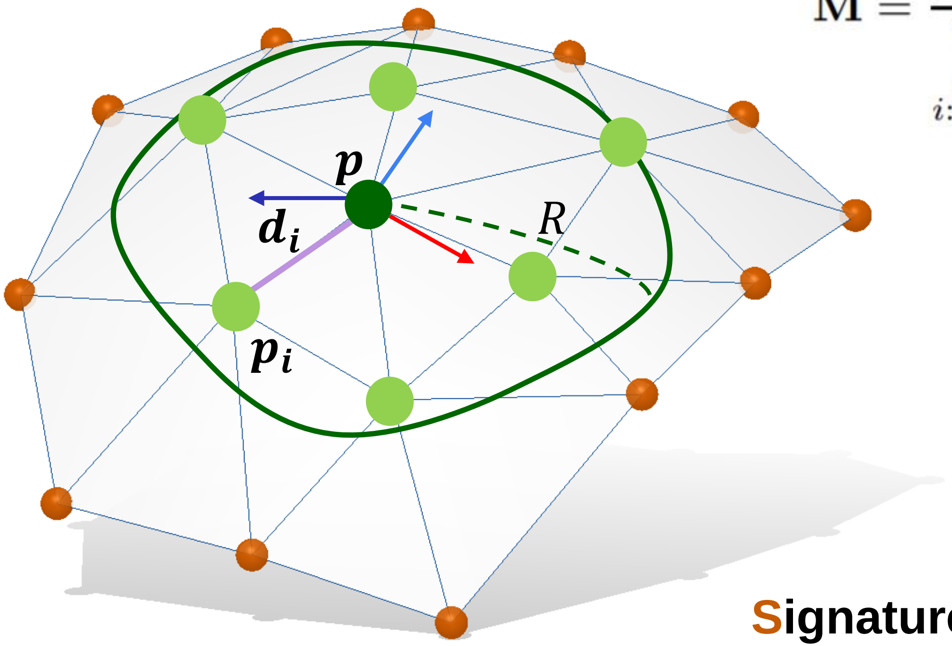
A descriptor (signature) should be:

1. Effective
2. Concise\compact
3. Repeatable
4. Robust

SHOT

For all $\boldsymbol{p}$ we define the covariance matrix:

$$\mathbf{M} = \frac{1}{\sum_{i:d_i \le R} (R-d_i)} \sum_{i:d_i \le R} (R-d_i)(\mathbf{p}_i - \mathbf{p})(\mathbf{p}_i - \mathbf{p})^T$$



From the eigenvectors of M we obtain a LRF (x, y, z) that is then used to define:

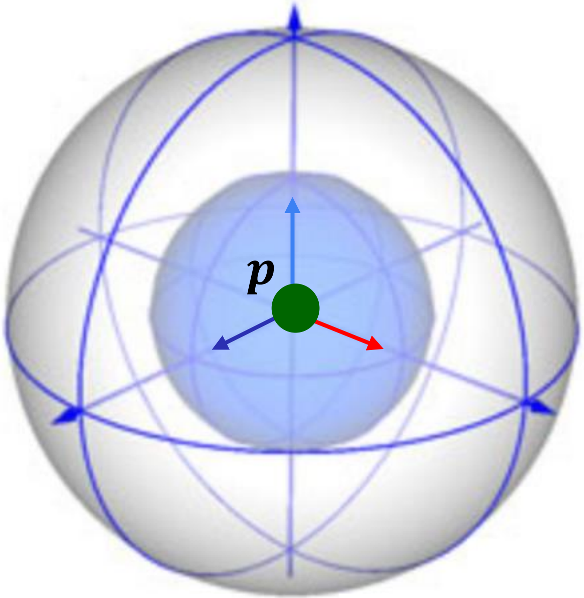
SHOT

Signature of Histograms of OrienTations

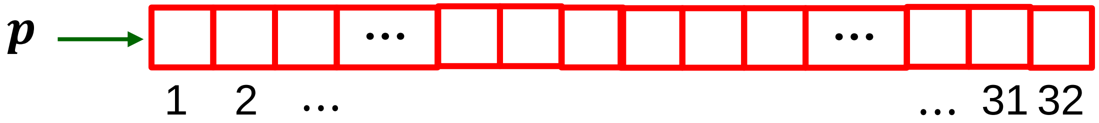
[Unique Signatures of Histograms for Local Surface Description, *Tombari et al.*, ECCV 2010]

SHOT: construction

Once we have the LRF for every point $\boldsymbol{p}$ we can define a **coherent 3D grid**



The 3D space around $\boldsymbol{p}$ is subdivided in 32 regions each of wich is a different bin of the histogram that describes the point.

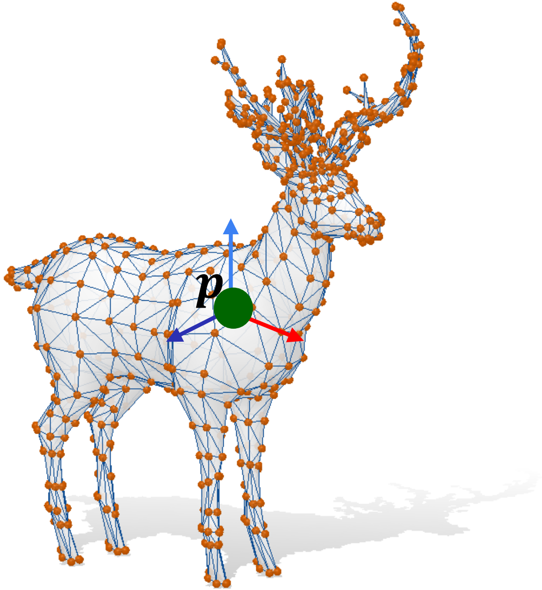


The value of each bin is a weighted sum of $cos\theta_i$ where θ_i is the angle between the normals of the point $\boldsymbol{p}$ and the point within each region of the 3D grid.

Unique Signatures of Histograms for Local Surface Description, *Tombari et al.*, ECCV 2010

SHOT: a comment

SHOT is an extrinsic descriptor: it depends on the 3D embedding of the shape



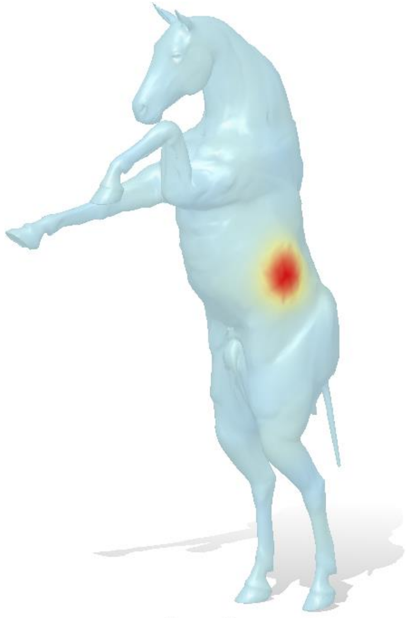
The analysis for the point p is performed looking at how the shape behaves around the point.

To obtain a coherent description of similar points and to be invariant to rigid deformations the LRF is necessary.

The SHOT descriptors is not invariant to non-rigid deformations.

Heat Diffusion

$\mathcal{X}$ is a Riemannian surface, $u(x,t)$ is the amount of heat in a point $x \in \mathcal{X}$ at time $t \in \mathbb{R}$


$t = 0$

Given a initial distribution u_0

of heat on $\mathcal{X}$ at time $t = 0$, ($u_0(x) = u(x,0)$)

How is it diffused over time on the surface?

A Concise and Provably Informative Multi-scale Signature Based on Heat Diffusion, *Sun et al.*, 2009.

Heat Diffusion

From physics the heat diffusion is governed by the **heat equation**:

$$\Delta_{\mathcal{X}} u(x,t) = -\frac{\partial u(x,t)}{\partial t}$$

The LBO
=
derivatives in space

derivative in time

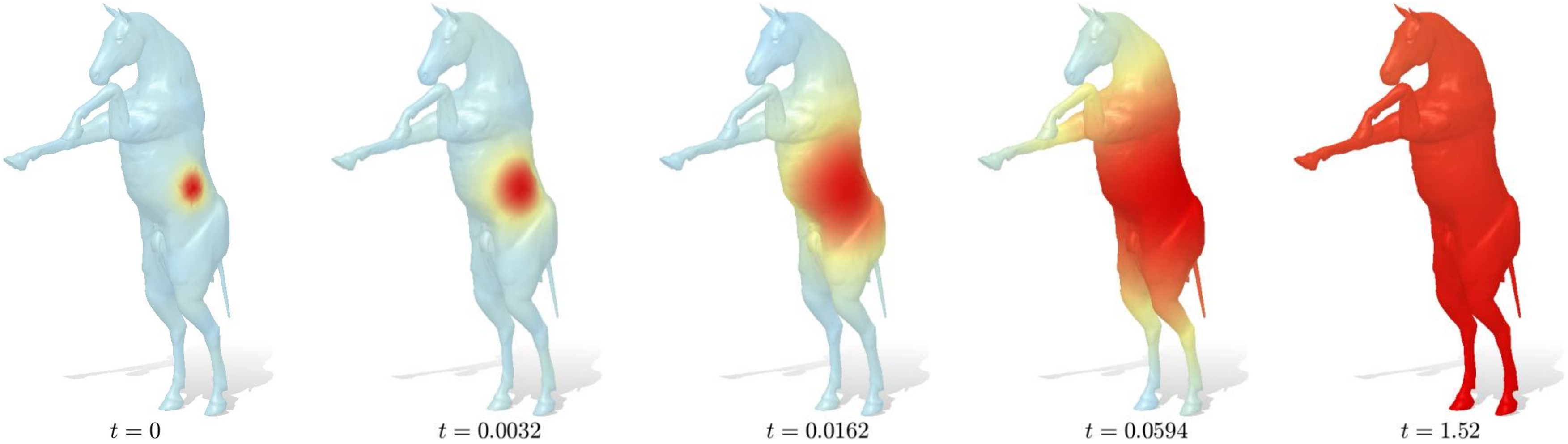
$u(x,t)$ solution of the heat equation is a function of $x \in \mathcal{X}$ and time $t \in \mathbb{R}$ which satisfies the **heat equation** for a given initial condition: $u_0(x) = u(x,0)$

Heat Diffusion solution

For an initial delta distribution of heat $\delta_x, \ x \in \mathcal{X}$

the heat kernel $k_t(x, y) = \sum_{l=0}^{\infty} e^{-t\lambda_l} \phi_l(x) \phi_l(y)$

Is the amount of heat moving from x to y



HKS: Heat Kernel Signature

For an initial delta distribution of heat $\delta_x, \; x \in \mathcal{X}$

$$k_t(x,x) = \sum_{l=0}^{\infty} e^{-t\lambda_l} \phi_l(x) \phi_l(x)$$

Is the amount of heat remaining at x after the time $t \in \mathbb{R}$

$$\boldsymbol{HKS}(x) = [k_{t_1}(x,x), k_{t_2}(x,x), \ldots, k_{t_Q}(x,x)] \quad t_1 < t_2 < \ldots t_Q \in \mathbb{R}$$

is the heat kernel signature (HKS) at the point $x \in \mathcal{X}$

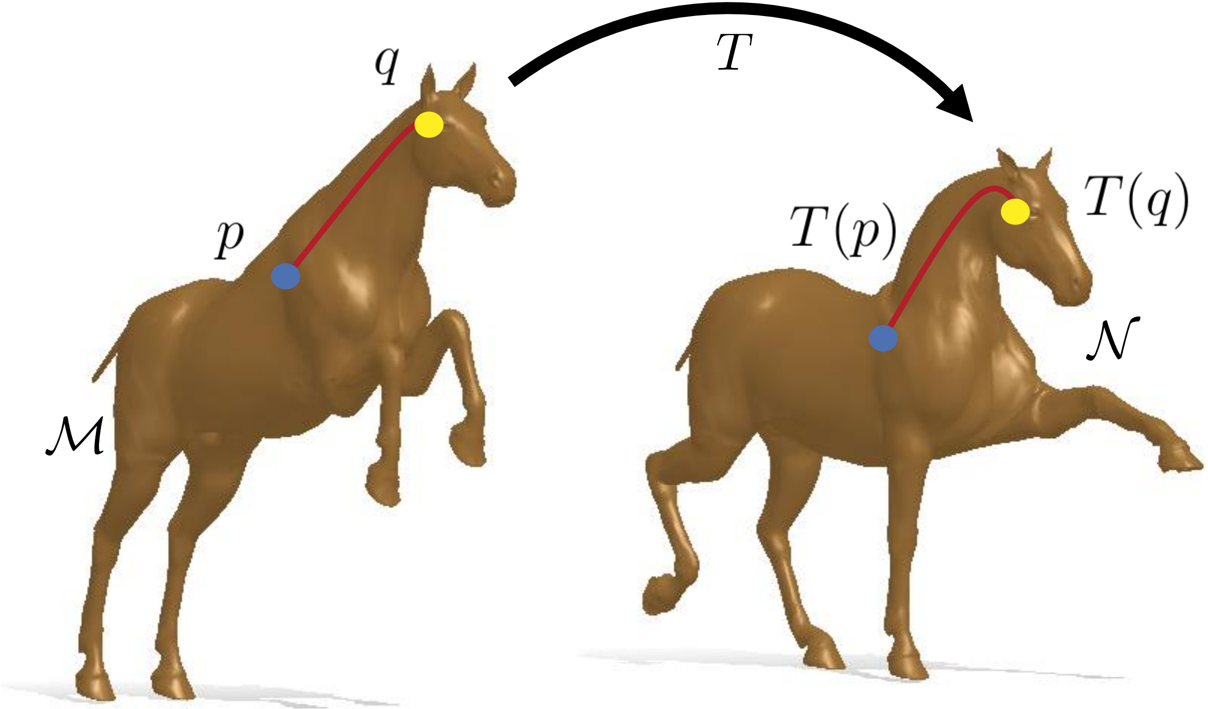
for a given set of time scales $t_1, \ldots, t_Q$

LBO and isometries

Two shapes are isometric ⇔ their LBO agree

T is an isometry $\iff d_{\mathcal{M}}(p,q) = d_{\mathcal{N}}(T(p),T(q)) \ \forall p,q \in \mathcal{M}$

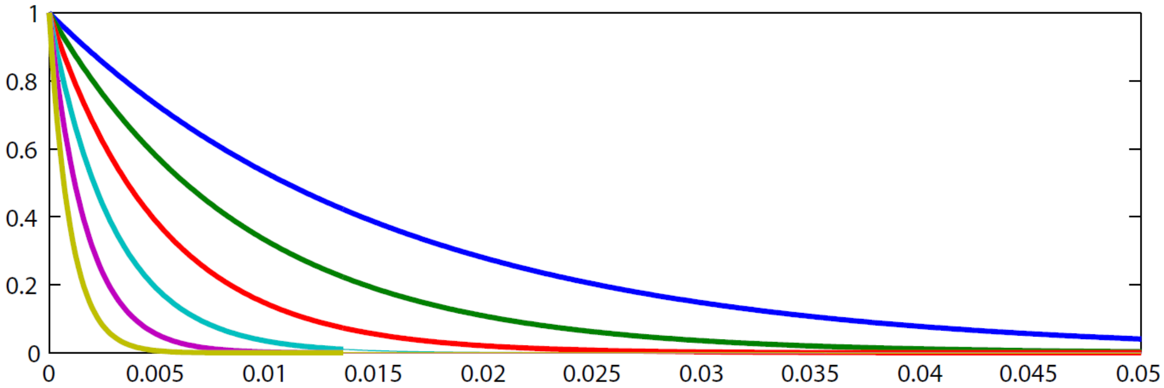
Any quantity derived
from the LBO
is invariant to isometry



HKS: as a filter on the frequencies

$$k_t(x,x) = \sum_{l=1}^{\infty} e^{-t\lambda_l}\phi_l(x)\phi_l(x) = \sum_{l=1}^{\infty} e^{-t\lambda_l}\phi_l(x)^2$$

$$g_t(\lambda_l) = e^{-t\lambda_l}$$



A low-pass filter applied to the frequencies to produce the HKS

The wave equation (Schrödinger)

Heat equation: $$\Delta_{\mathcal{X}} u(x,t) = -\frac{\partial u(x,t)}{\partial t}$$

Wave equation: $$i\Delta_{\mathcal{X}} u(x,t) = \frac{\partial u(x,t)}{\partial t}$$

presence of the *i*

missing a minus

It governs the temporal evolution of a quantum particle

It encodes oscillation rather than dissipation as done by the heat equation

Idea: point $x \leftrightarrow$ the average probabilities of quantum particles of different energies to be measured at x

[The wave kernel signature: A quantum mechanical approach to shape analysis, *Aubry et al.*, 2011.]

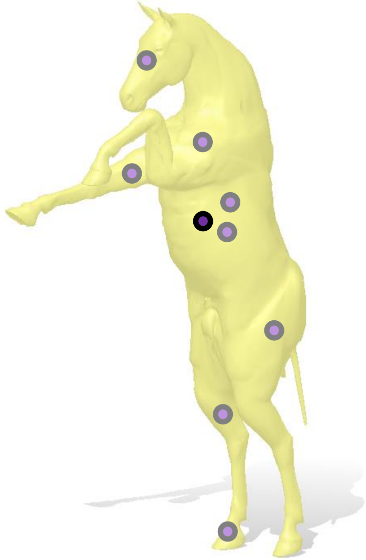
WKS: Wave Kernel Signature

- a quantum particle with unknown position on the surface

f_E^2 the probability distribution with expectation value E estimated at time $t = 0$

$$WKS(E, x) = \sum_{l=1}^{\infty} f_E(E_l)^2 \phi_l(x)^2$$

the average probability over the time to find the particle at position $x \in \mathcal{X}$ given the initial energy E



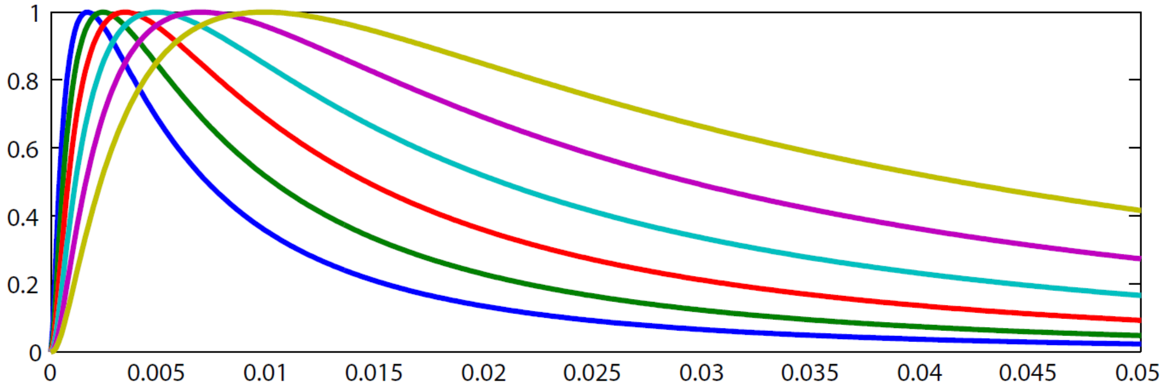
$$WKS(x) = [WKS(E_1, x), WKS(E_2, x), \ldots, WKS(E_Q, x)]$$

WKS: as a filter on the frequencies

$$k_E(x,x) = \sum_{l=1}^{\infty} e^{-\frac{(log(E) - log(\lambda_l))^2}{2\sigma^2}} \phi_l(x)^2$$

$$g_t(\lambda_l) = e^{-\frac{(log(E) - log(\lambda_l))^2}{2\sigma^2}}$$

A band-pass filter applied to the frequencies to produce the WKS



Spectral descriptors

A common structure is shared by the spectral descriptors **HKS** and **WKS**

$$desc_q(x) = \sum_{l=1}^{k} g_{t_q}(\lambda_l)\phi_l^2(x), \quad \forall q \in 1, \ldots, Q$$

A set of filters on the frequencies

=

functions of the eigenvalues

The square of each dimension of the spectral embedding

Spectral descriptors: as filter on the frequencies

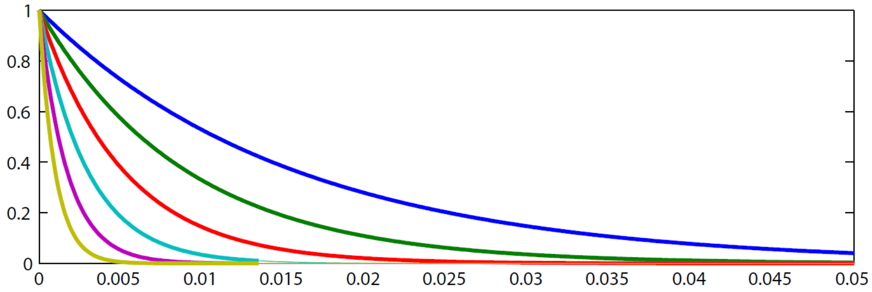
$$desc_q(x) = \sum_{l=1}^{k} g_{t_q}(\lambda_l)\phi_l^2(x), \quad \forall q \in 1, \dots, Q$$

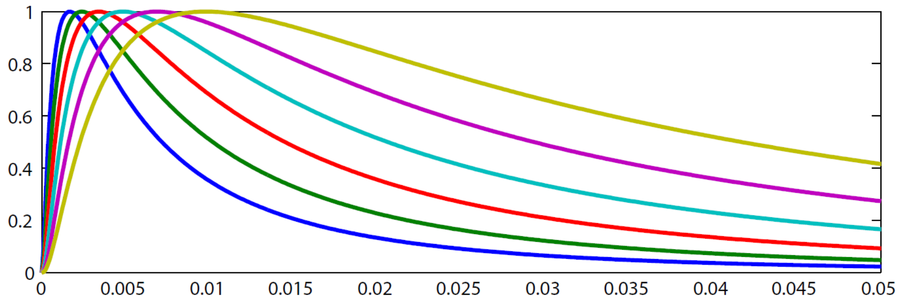
HKS:

$$g_t(\lambda_l) = e^{-t\lambda_l}$$

WKS:

$$g_t(\lambda_l) = e^{-\frac{(log(E) - log(\lambda_l))^2}{2\sigma^2}}$$





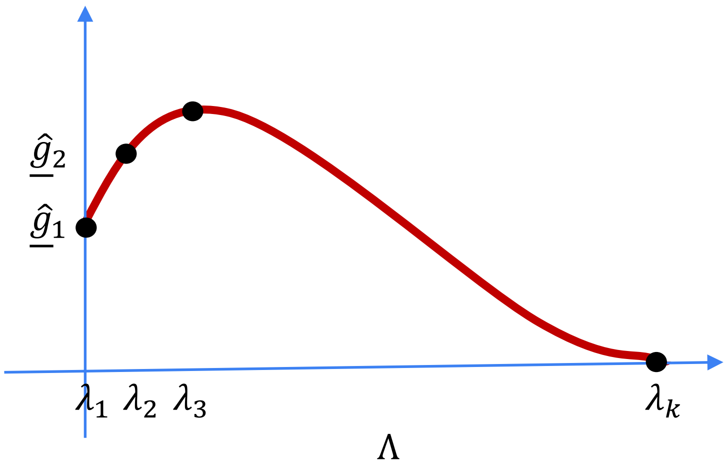
"Learning spectral descriptors for deformable shape correspondence", *Litman et al.*, 2014.

Spectral descriptors: learn the optimal filters

$$desc_q(x) = \sum_{l=1}^{k} g_{t_q}(\lambda_l)\phi_l^2(x), \quad \forall q \in 1,\ldots,Q$$

What are the best filters to apply in this equation to obtain the best descriptors?

These are just Q finite sets of k parameters, can we learn them?



Optimal spectral descriptors

We can compute a learned kernel signature by learning the matrix $\boldsymbol{A} \in \mathbb{R}^{Q \times Z}$

$$\boldsymbol{LKS}(x) = [desc_1^{\boldsymbol{A}}(x), desc_2^{\boldsymbol{A}}(x), \ldots, desc_q^{\boldsymbol{A}}(x)]$$

These explixitly depend on the learned matrix

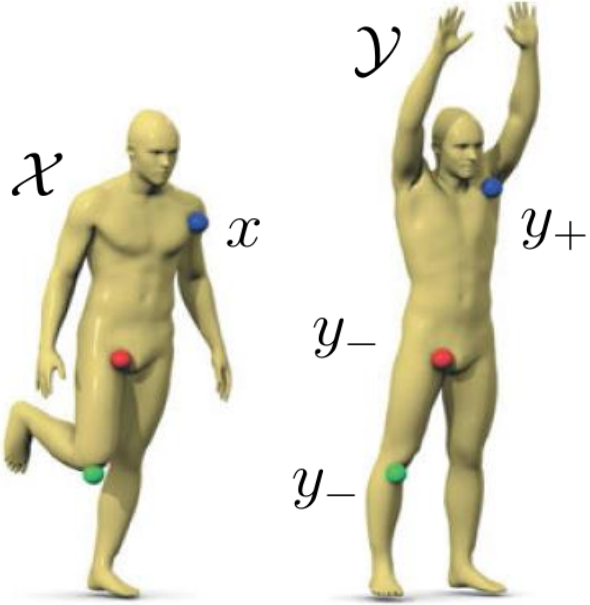
How could we learn this matrix A?

Optimal spectral descriptors: loss definition

Given a pair of shapes $\mathcal{X}$ and $\mathcal{Y}$

We consider a set of points X on $\mathcal{X}$ such that $\forall x \in X$ we can define a set of points Y on $\mathcal{Y}$ that is composed by:

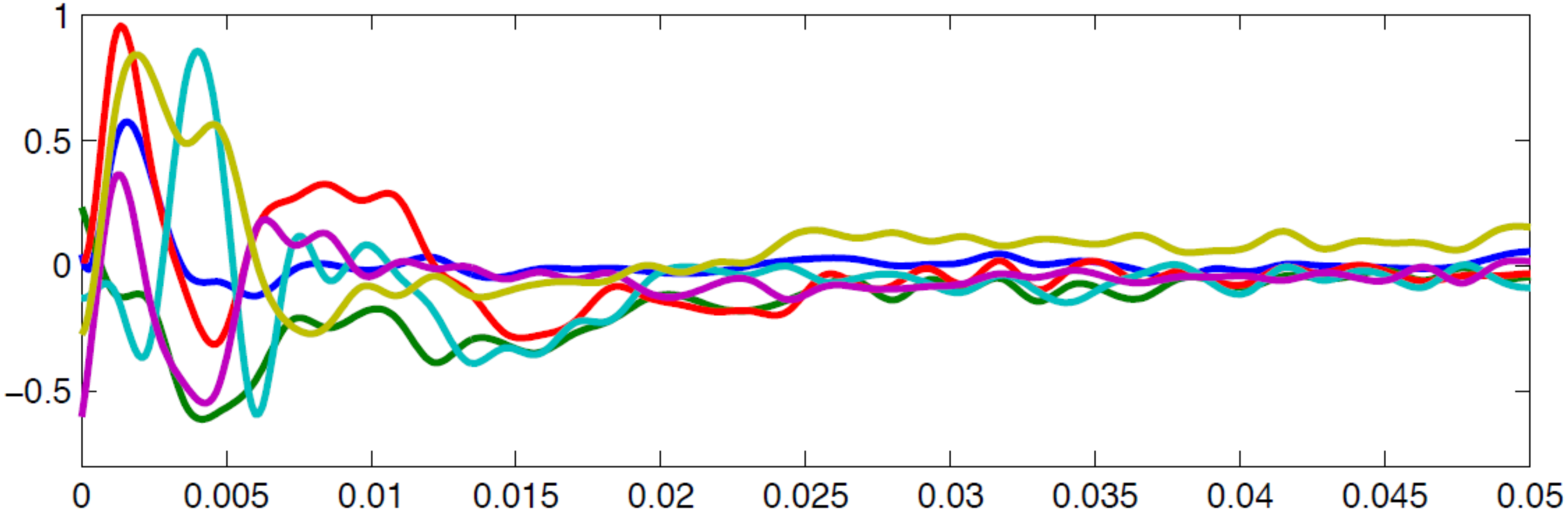
- similar points (**positive**) y_+
- dissimilar points (**negative**) y_-

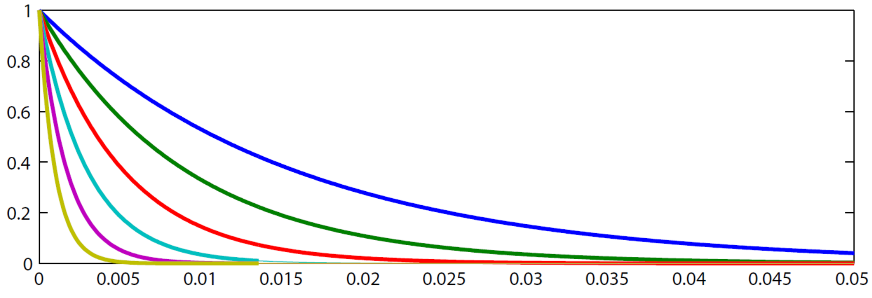


$$\underset{\boldsymbol{A}}{argmin} \sum_{x \in X} \gamma(\|LKS(x) - LKS(y_+)\|^2) \; - (1-\gamma)(\|LKS(x) - LKS(y_-)\|^2)$$

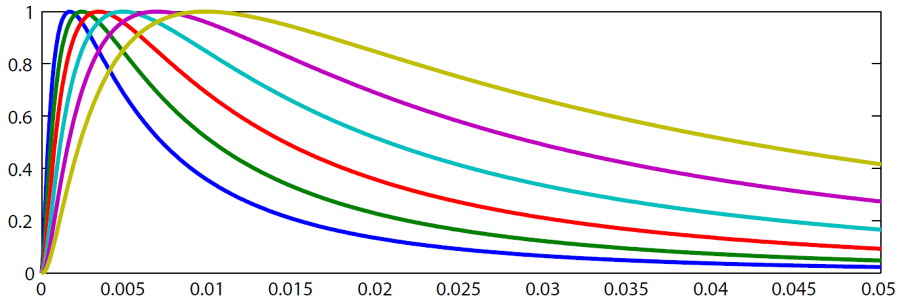
$$\boldsymbol{LKS}(x) = [desc_1^{\boldsymbol{A}}(x), desc_2^{\boldsymbol{A}}(x), \ldots, desc_q^{\boldsymbol{A}}(x)]$$

Optimal spectral descriptors: learned filters



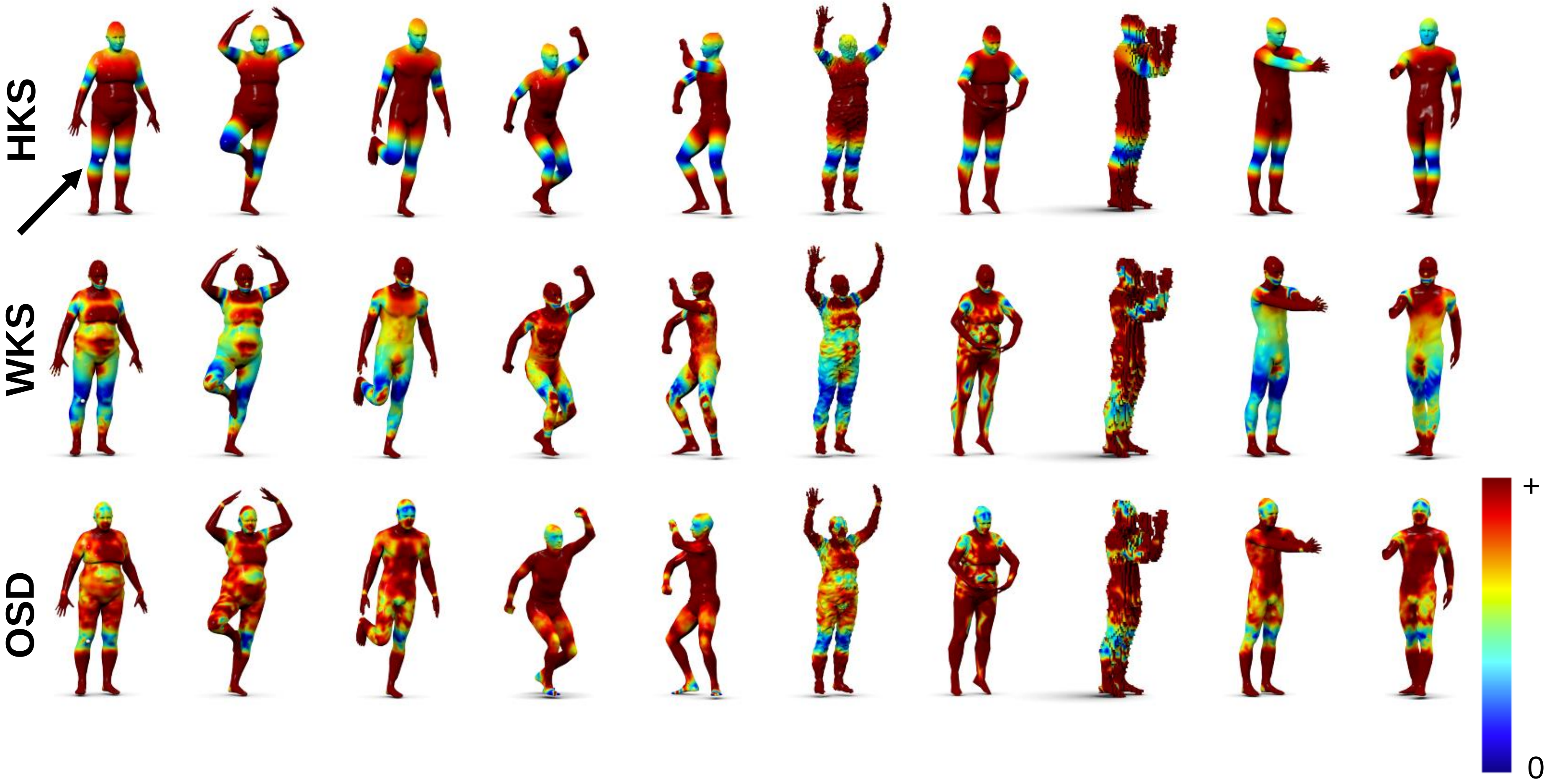


HKS:

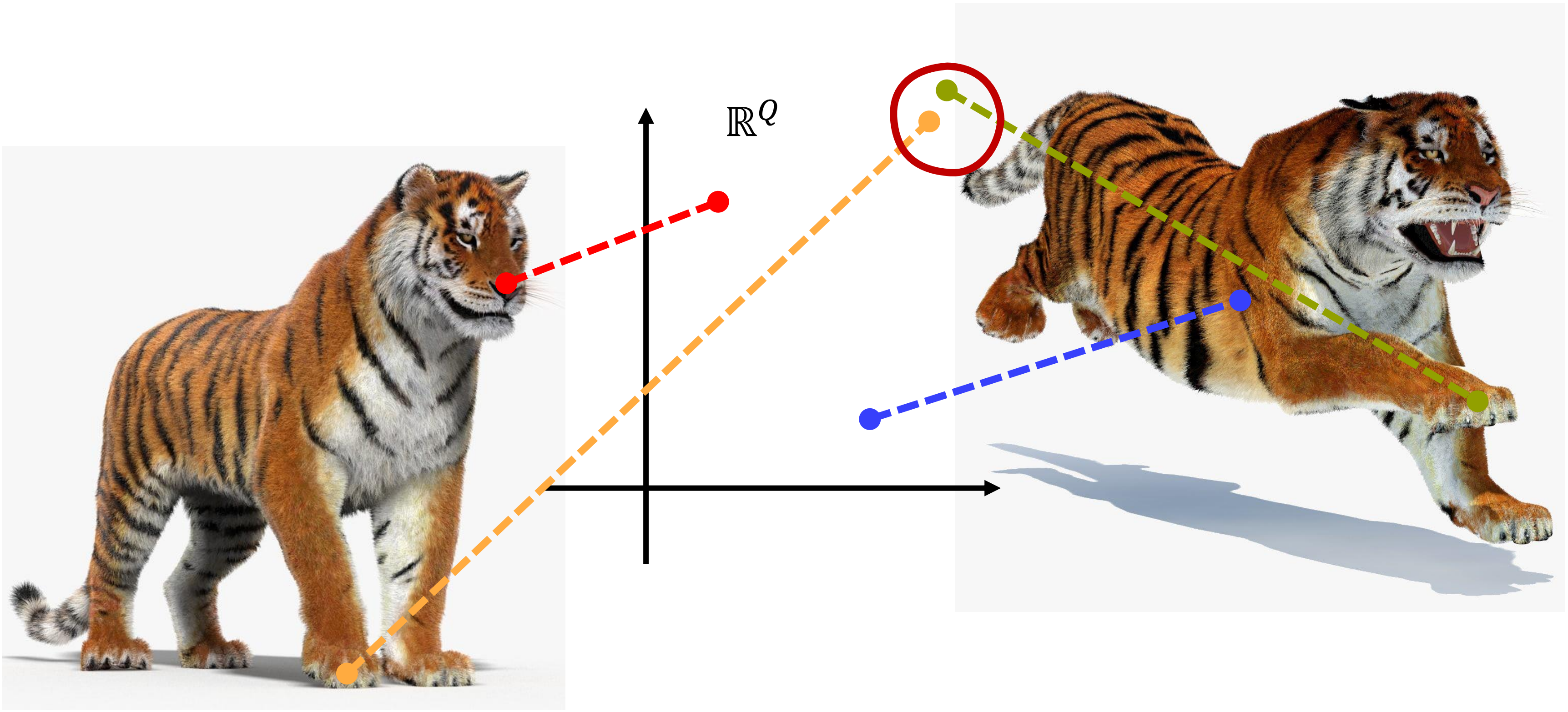


WKS:

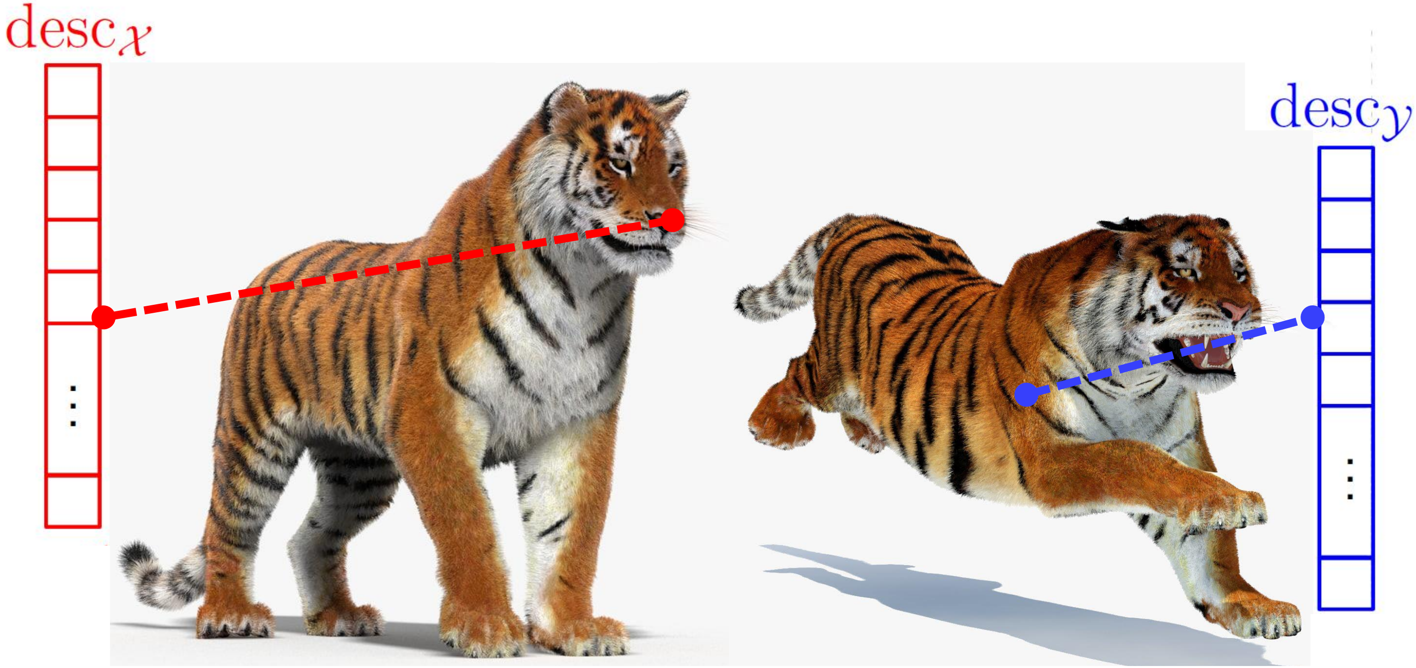
Qualitative comparison



Features-based

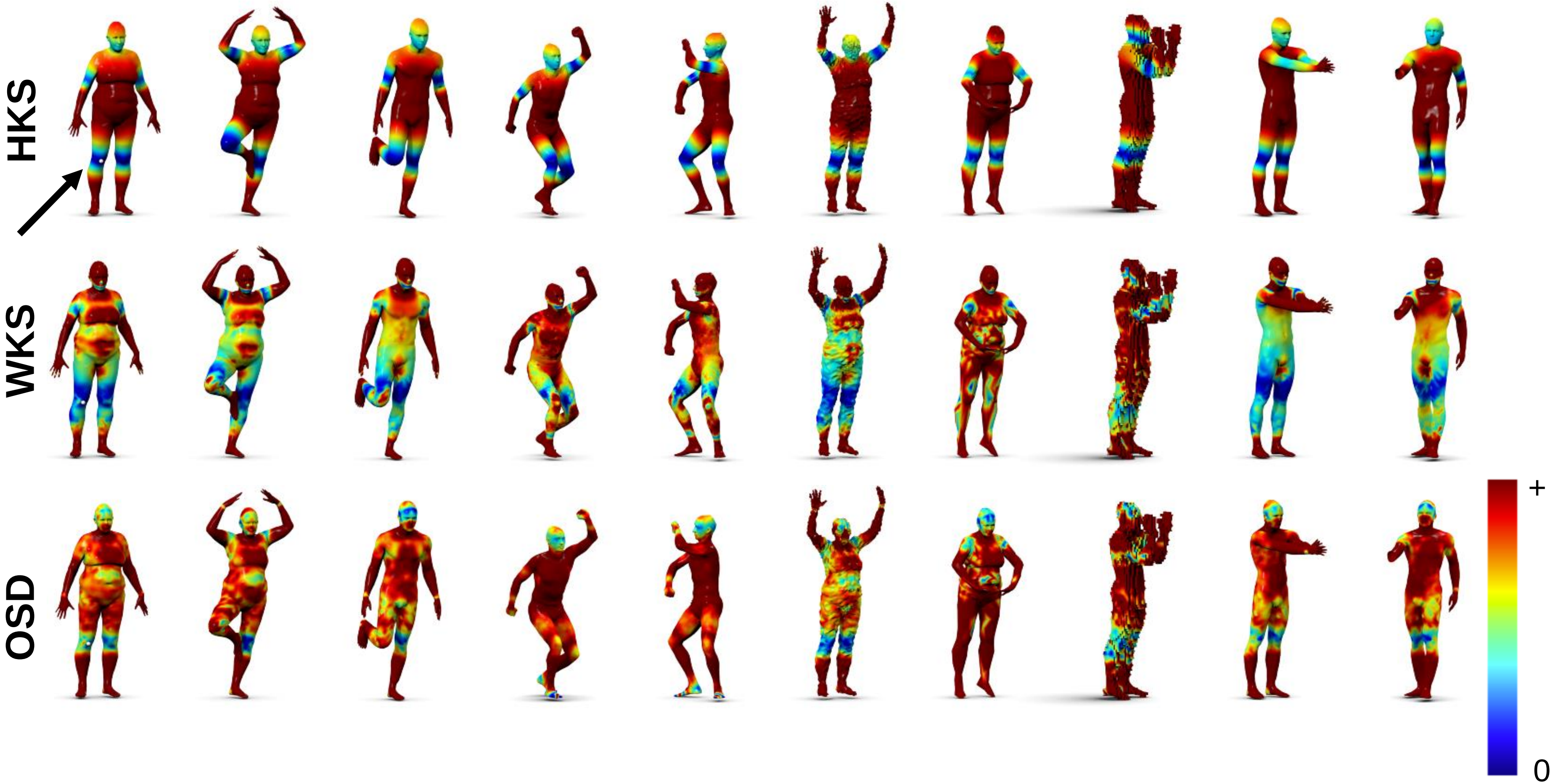


Features-based



$$\text{Distance} = \left\| desc_{\mathcal{X}} - desc_{\mathcal{Y}} \right\|_2$$

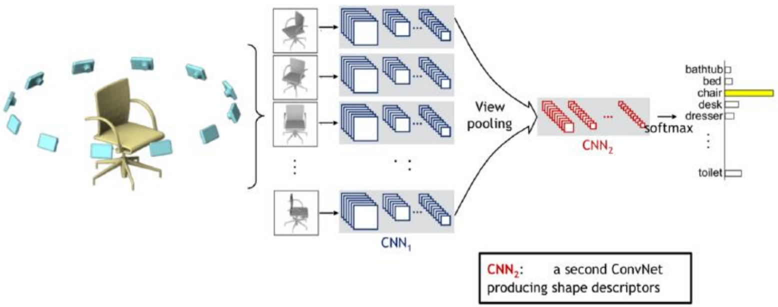
Qualitative comparison



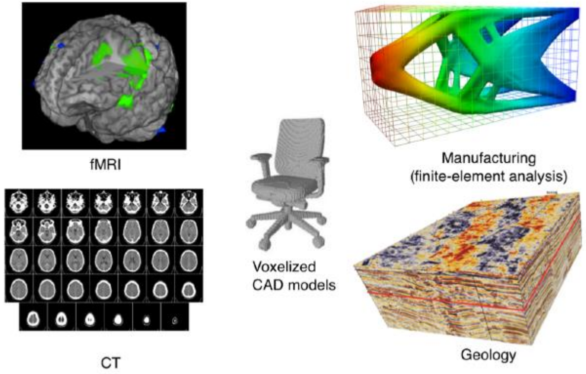
Some considerations

- SHOT can characterize the rigid geometry of a shape
- Spectral descriptors do not solve the symmetries
- Spectral descriptors can be generalized via data-driven approaches
- Spectral descriptors are invariant to isometric deformations
- The data-driven approaches outperform the standard spectral ones
- Other deformations (for from isometries) can not be faced

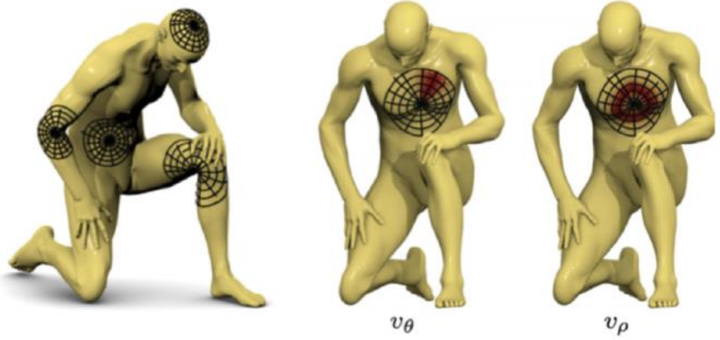
Geometric Deep Learning



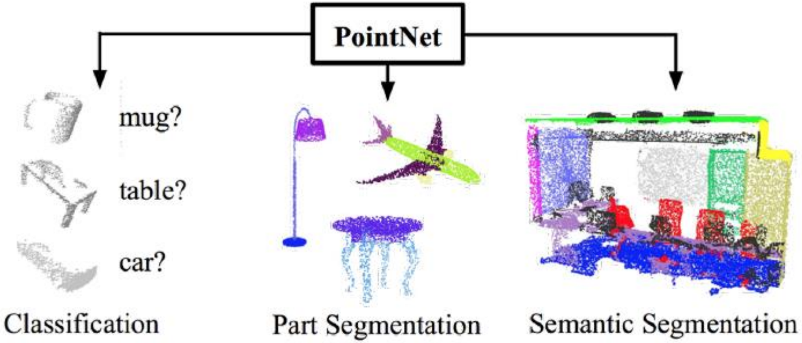
View-based



Volumetric



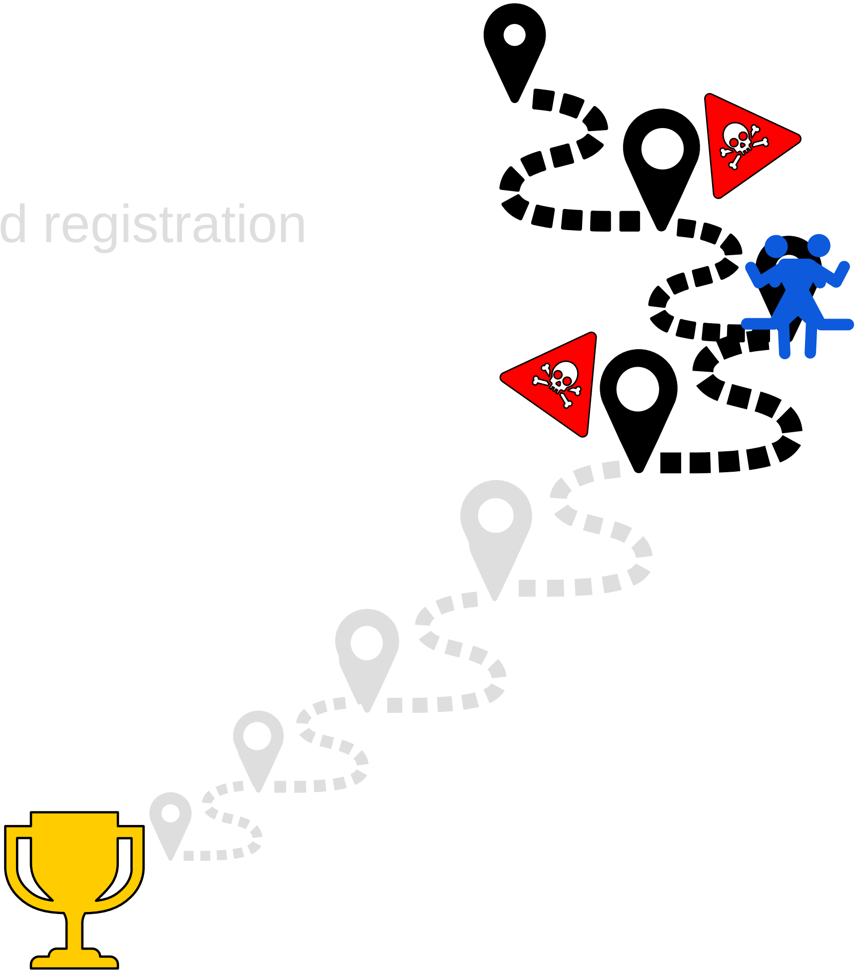
Intrinsic (surface-based)



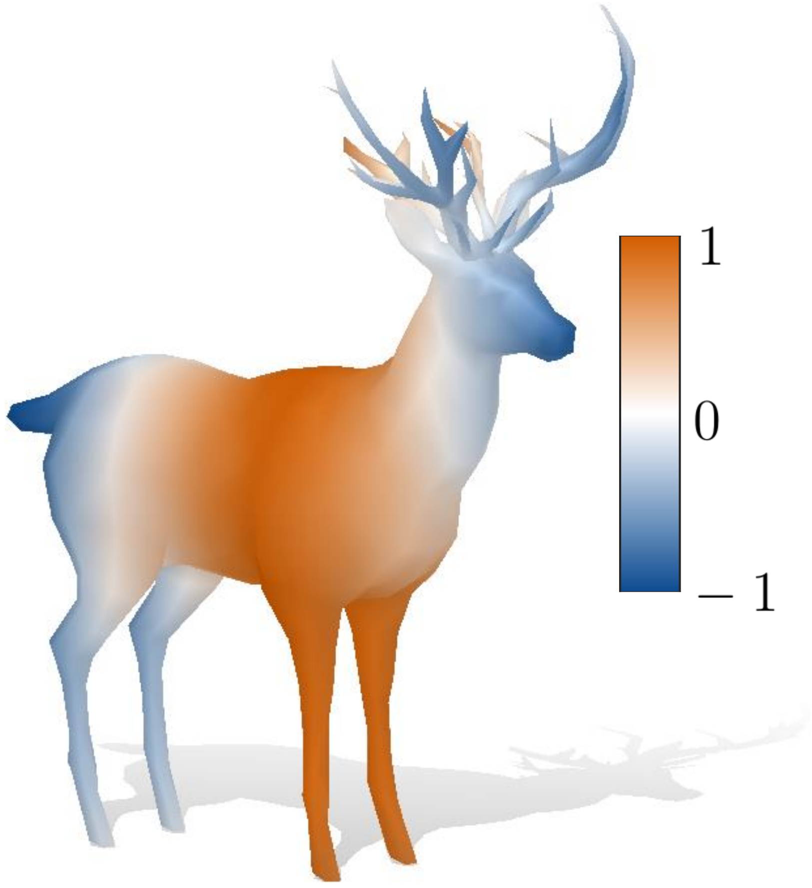
Point-based

FUNCTIONAL MAPS

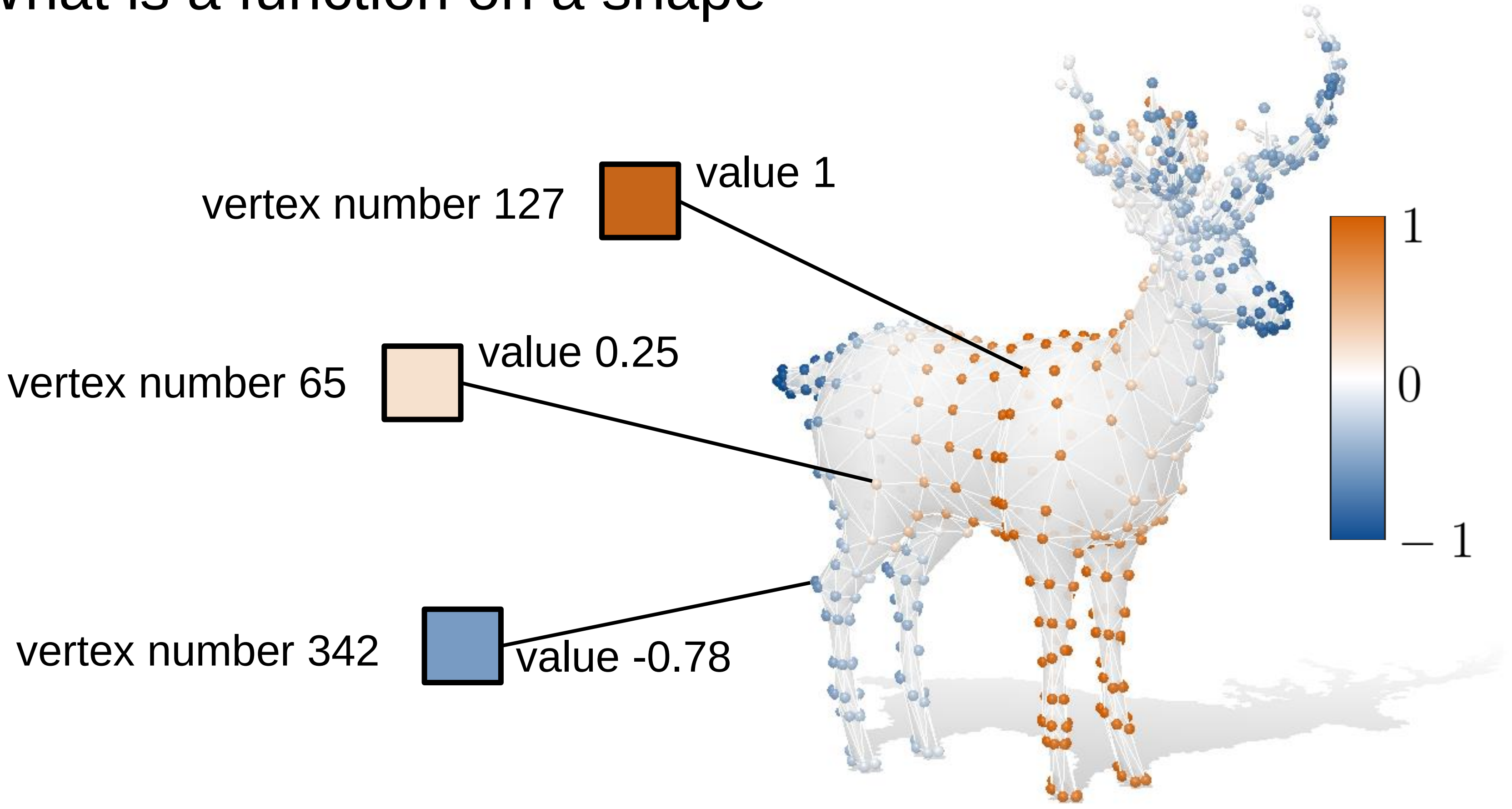
1. Introduction: 3D Non-Rigid shapes and registration
2. Spectral representation
3. Axiomatic approaches
4. **Functional maps**
5. Learning on geometric data
6. Learning–based Functional maps
7. Other learning-based approaches
8. Transformers



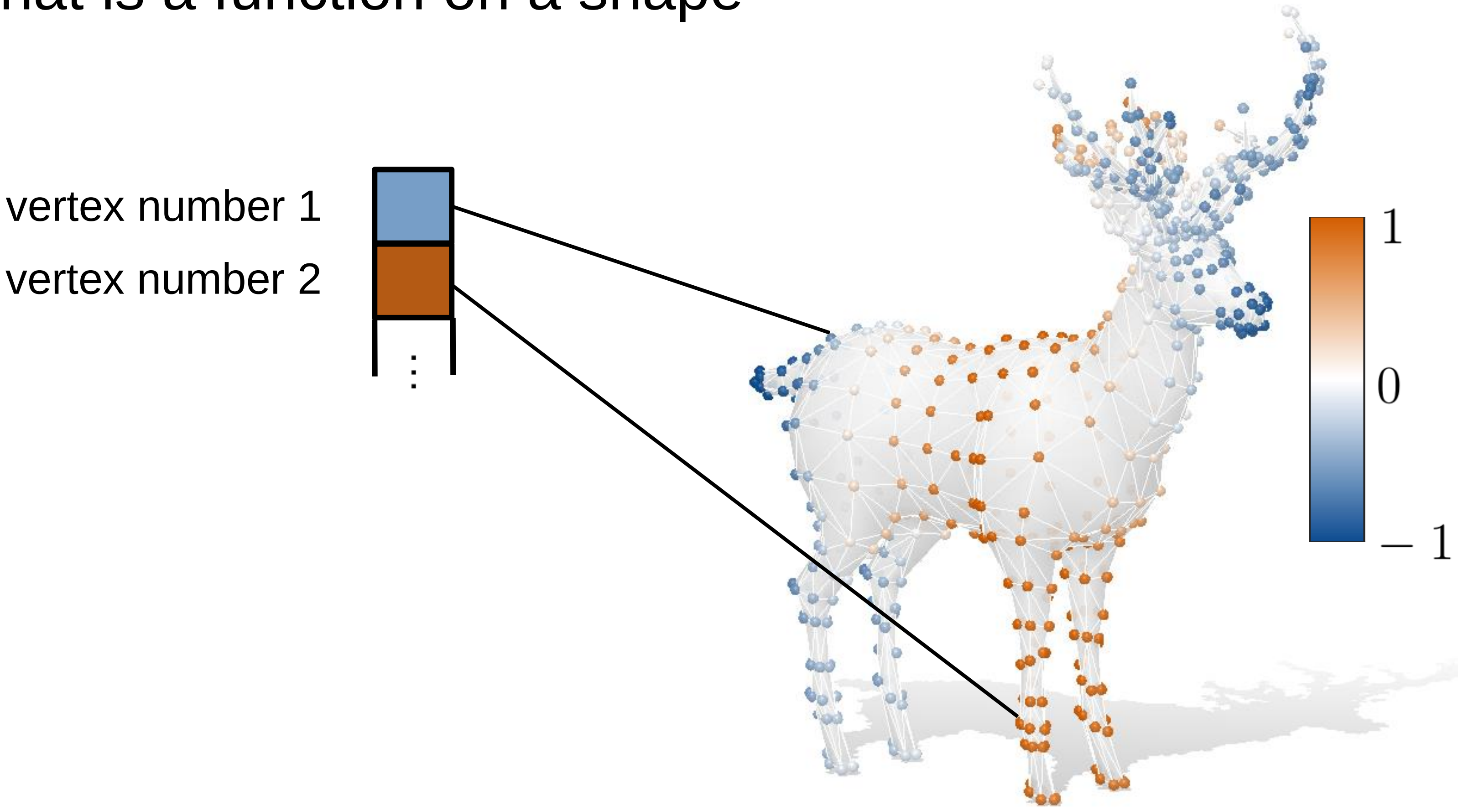
What is a function on a shape



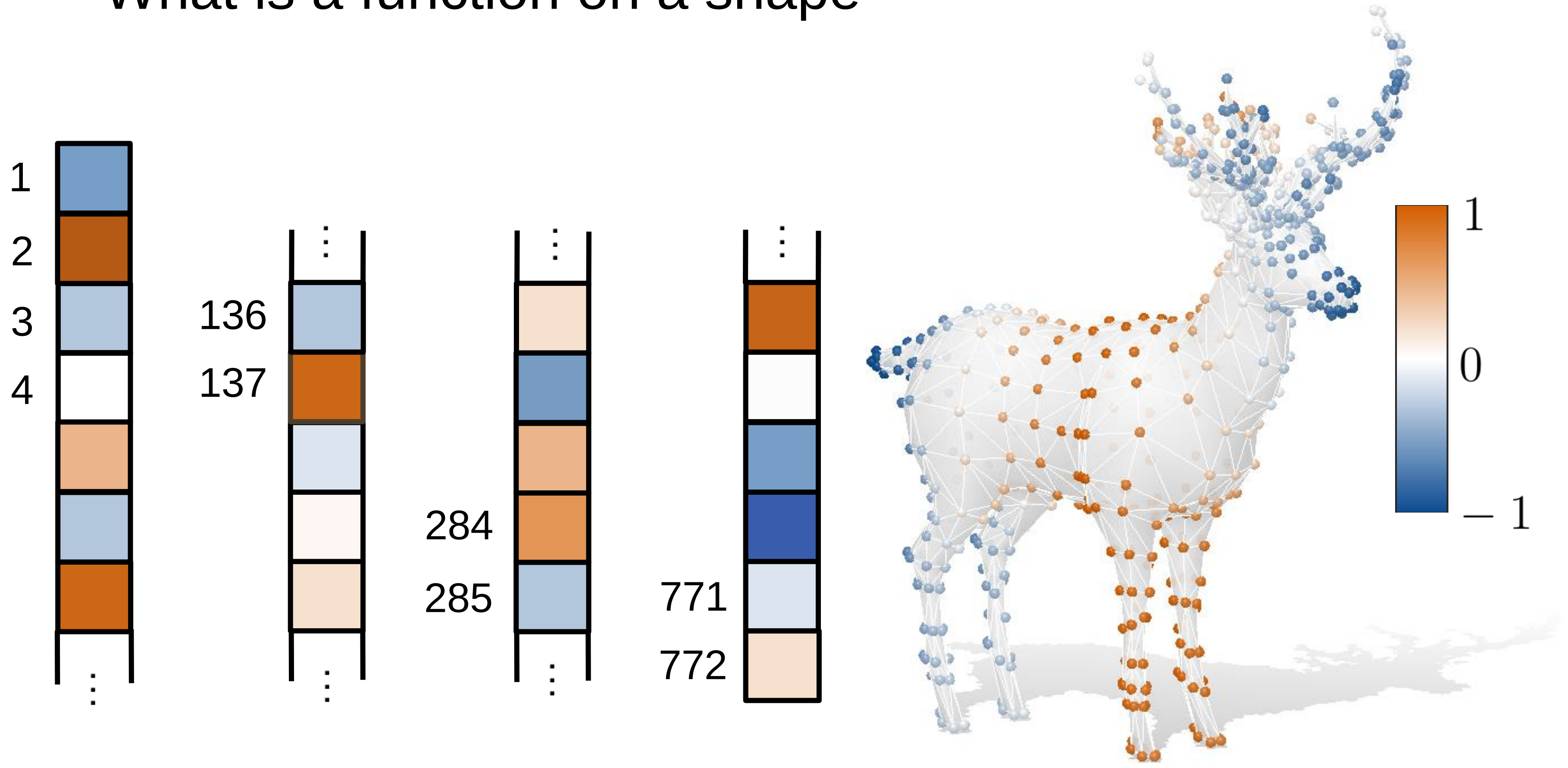
What is a function on a shape



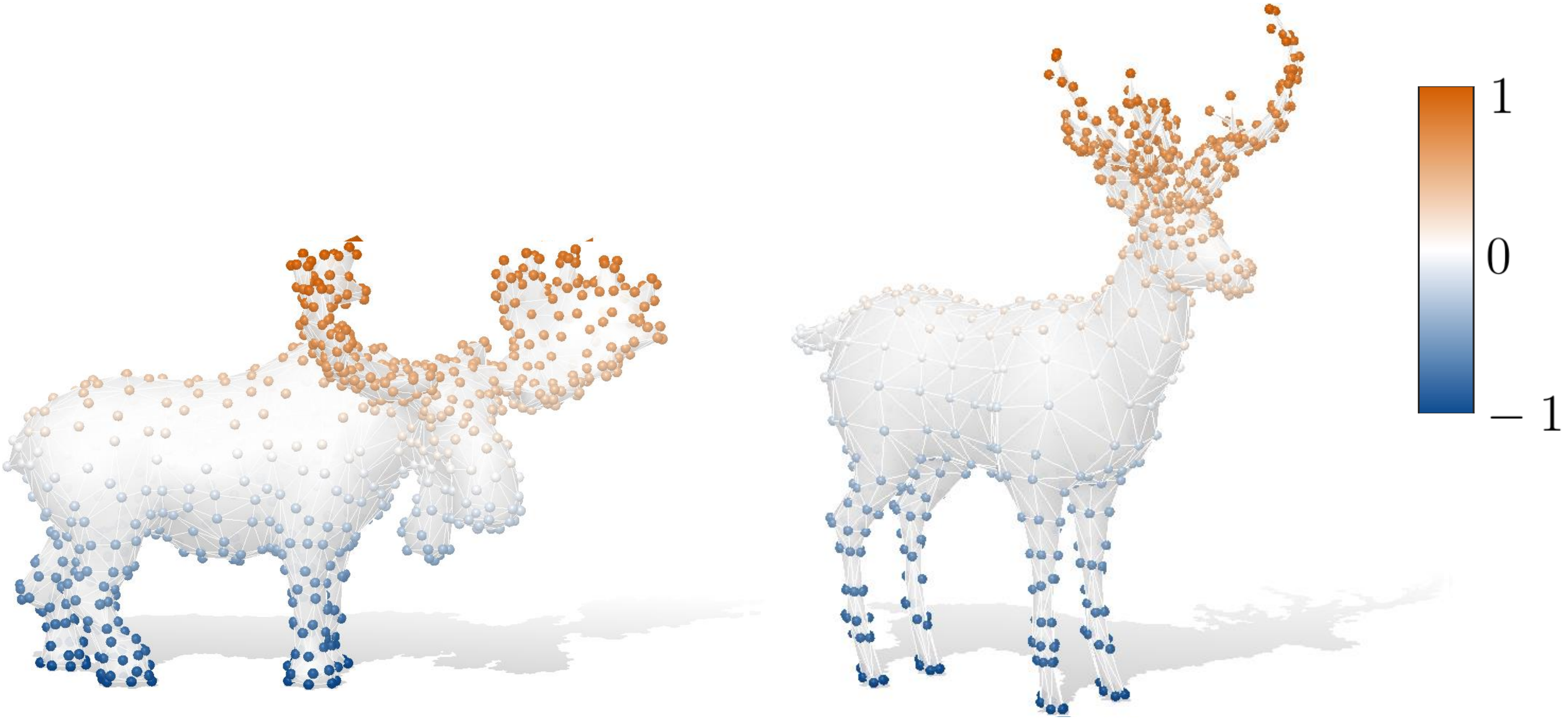
What is a function on a shape



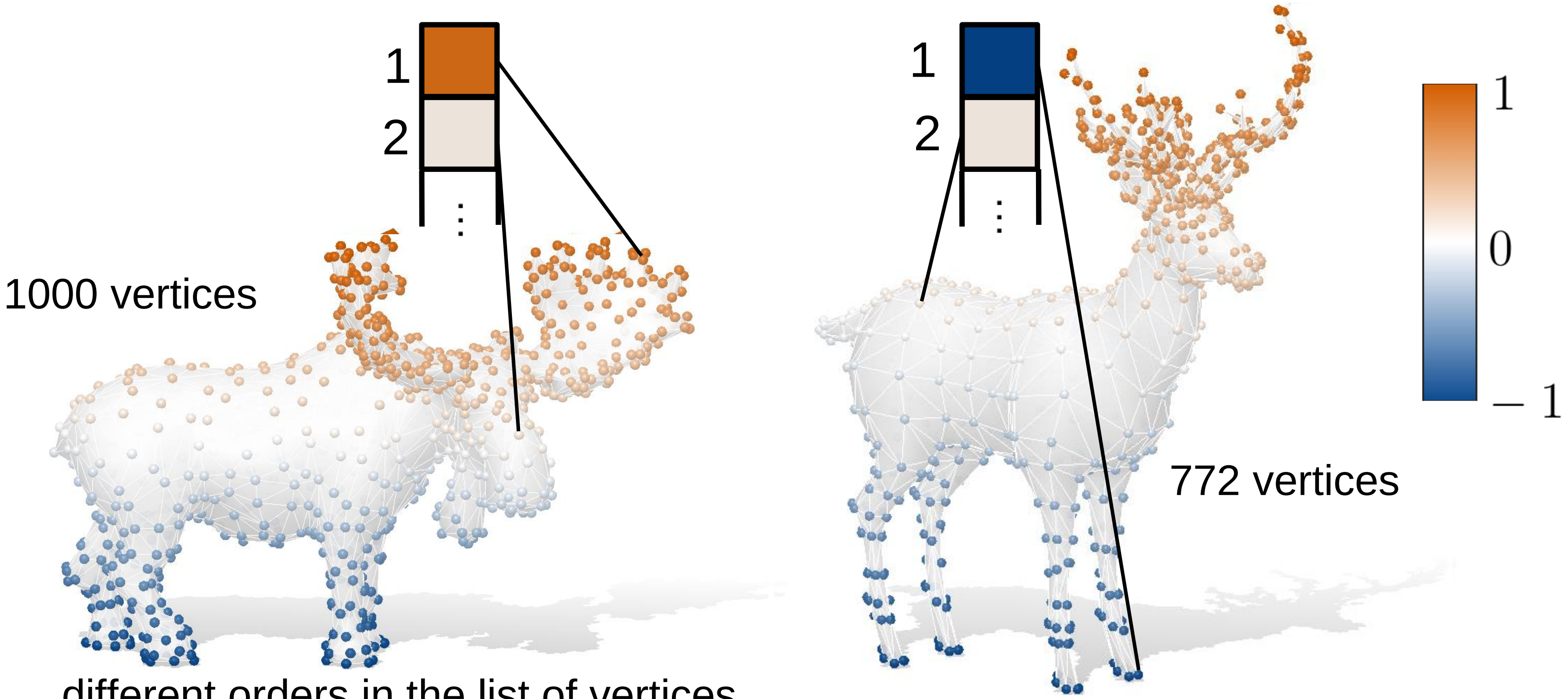
What is a function on a shape



Different meshes

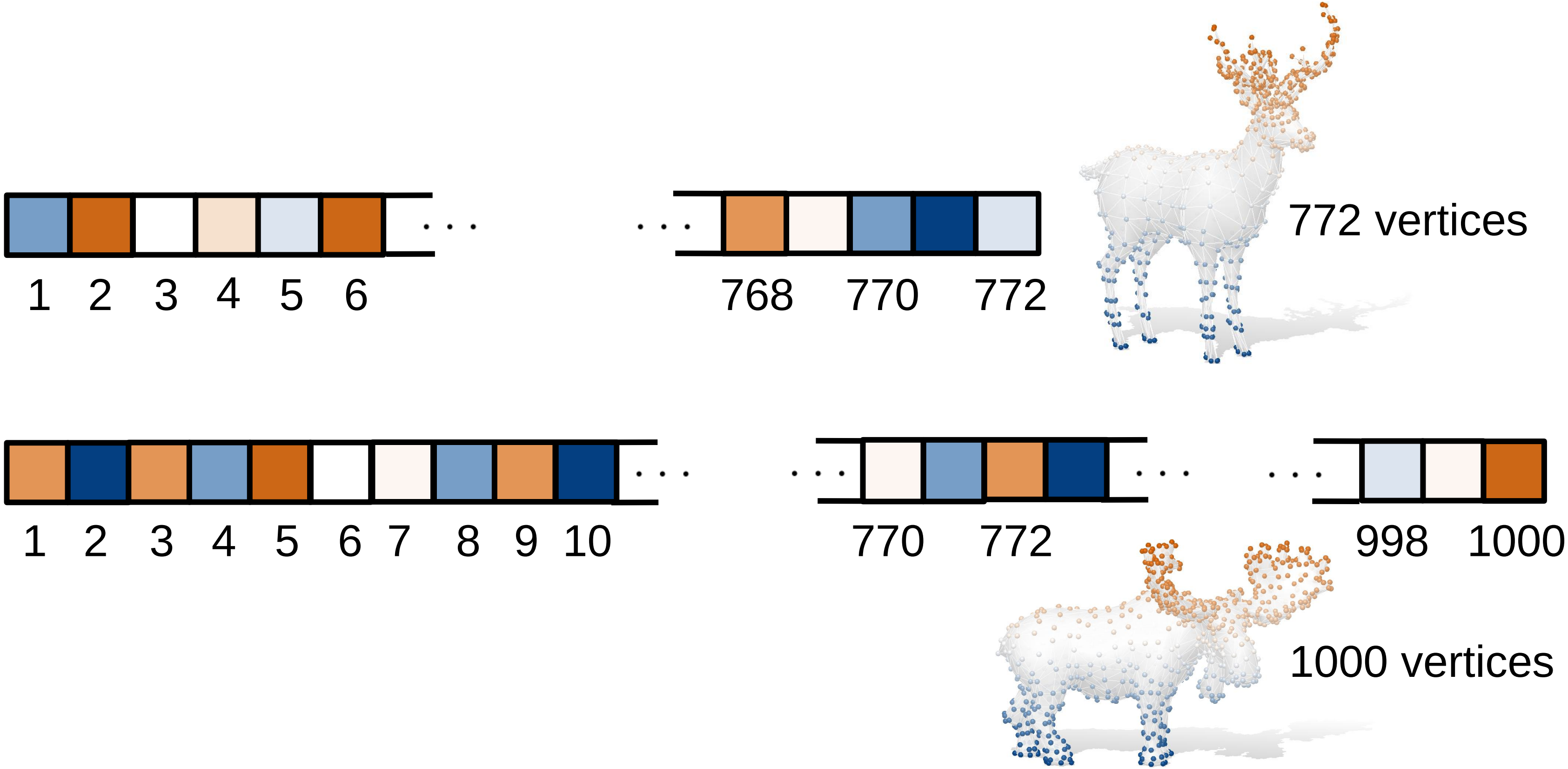


Different meshes



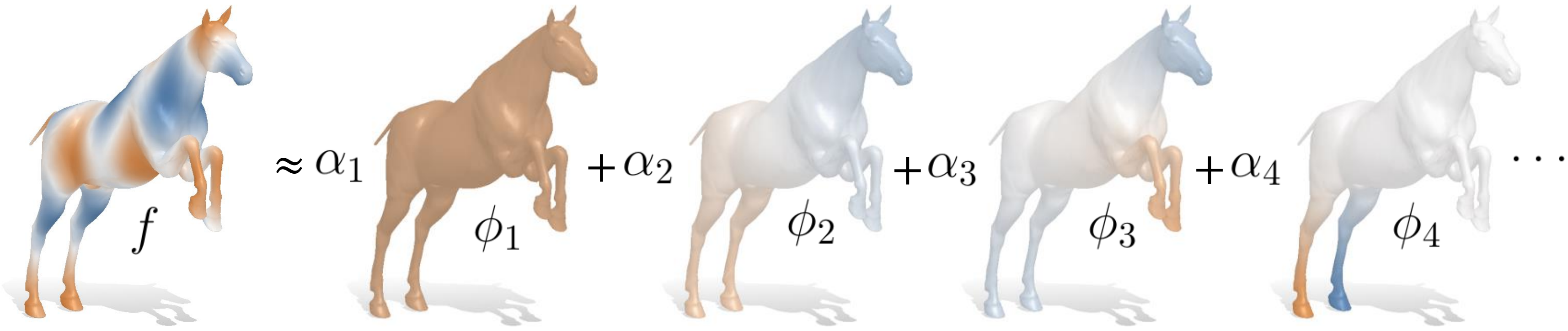
different orders in the list of vertices

Different vectors



These representations are not comparable!

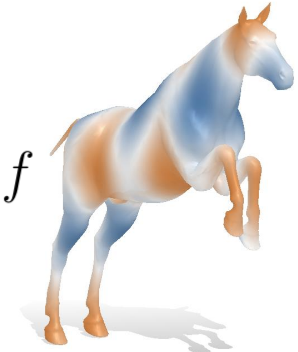
Fourier representation



Laplace-beltrami eigenfunctions towards an algorithm that understands geometry, *Levy*, 2006

Synthesis and analysis: dicrete setting

Given a signal: f



$$\langle f, g \rangle_{\mathcal{M}} = f^{\top} \mathbf{\Omega}_{\mathcal{M}} g$$

$$\mathbf{\Phi}_{\mathcal{M}} = [\phi_1, \ \phi_2, \ \ldots \ \phi_{k-1}, \ \phi_k]$$

$$\mathbf{\Phi}^{\dagger}_{\mathcal{M}} \ s.t. \ \mathbf{\Phi}^{\dagger}_{\mathcal{M}} \mathbf{\Phi}_{\mathcal{M}} = I$$

$$\mathbf{\Phi}_{\mathcal{M}} \ s.t. \ \mathbf{\Phi}^{\top}_{\mathcal{M}} \mathbf{\Omega}_{\mathcal{M}} \mathbf{\Phi}_{\mathcal{M}} = I$$

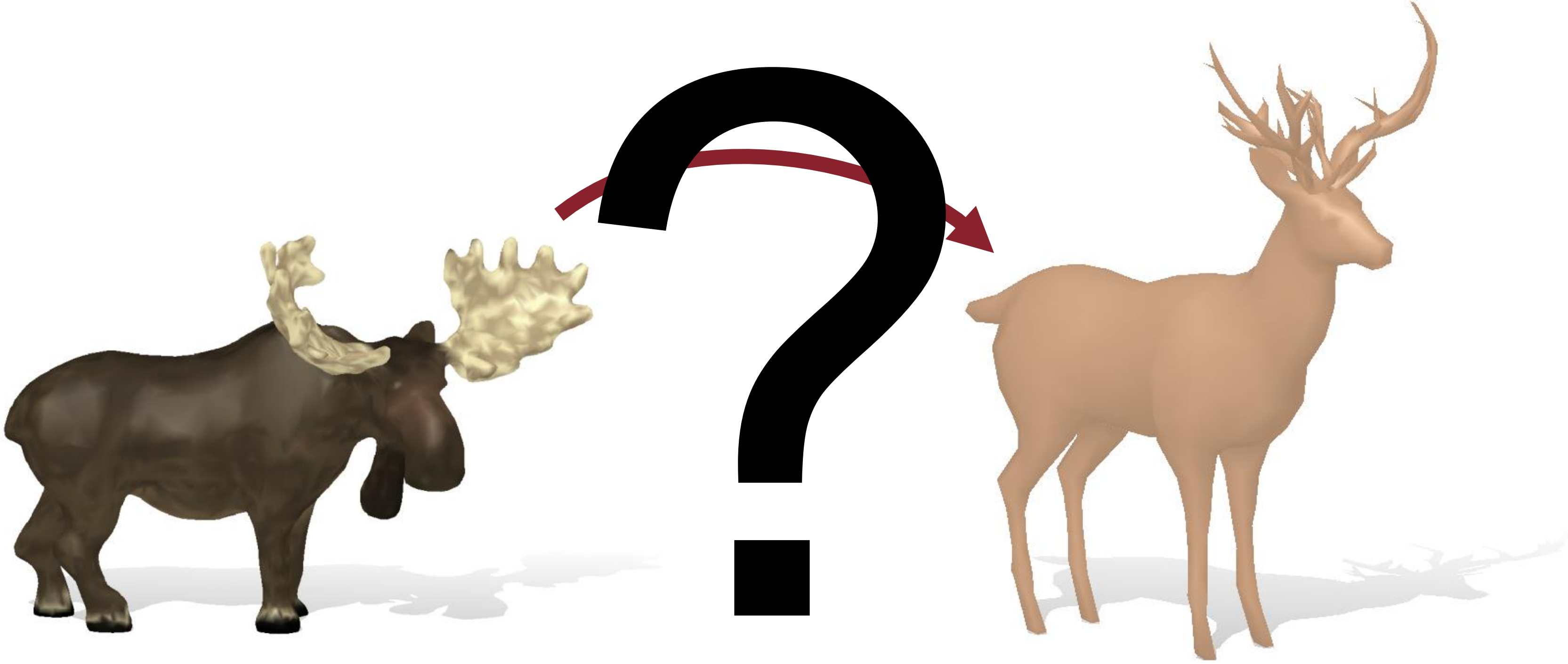
$$\mathbf{\Phi}^{\dagger}_{\mathcal{M}} = \mathbf{\Phi}^{\top}_{\mathcal{M}} \mathbf{\Omega}_{\mathcal{M}}$$

The analysis: $\boldsymbol{\alpha} = \mathbf{\Phi}^{\dagger}_{\mathcal{M}} f$

The synthesis: $f = \mathbf{\Phi}_{\mathcal{M}} \boldsymbol{\alpha} = \mathbf{\Phi}_{\mathcal{M}} \mathbf{\Phi}^{\dagger}_{\mathcal{M}} f$

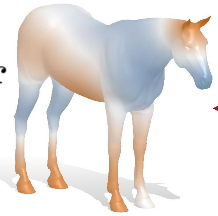
Laplace-beltrami eigenfunctions towards an algorithm that understands geometry, *Levy*, 2006

Question 1



Not directly possible! They are 2 different templates

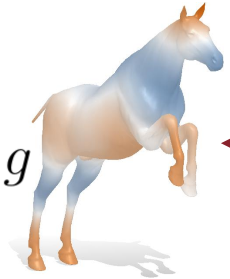
Question 2

f 

Fourier

$$[\alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4,\ \alpha_5,\ \dots\ \alpha_{29},\ \alpha_{30}]^\top = \boldsymbol{a}$$

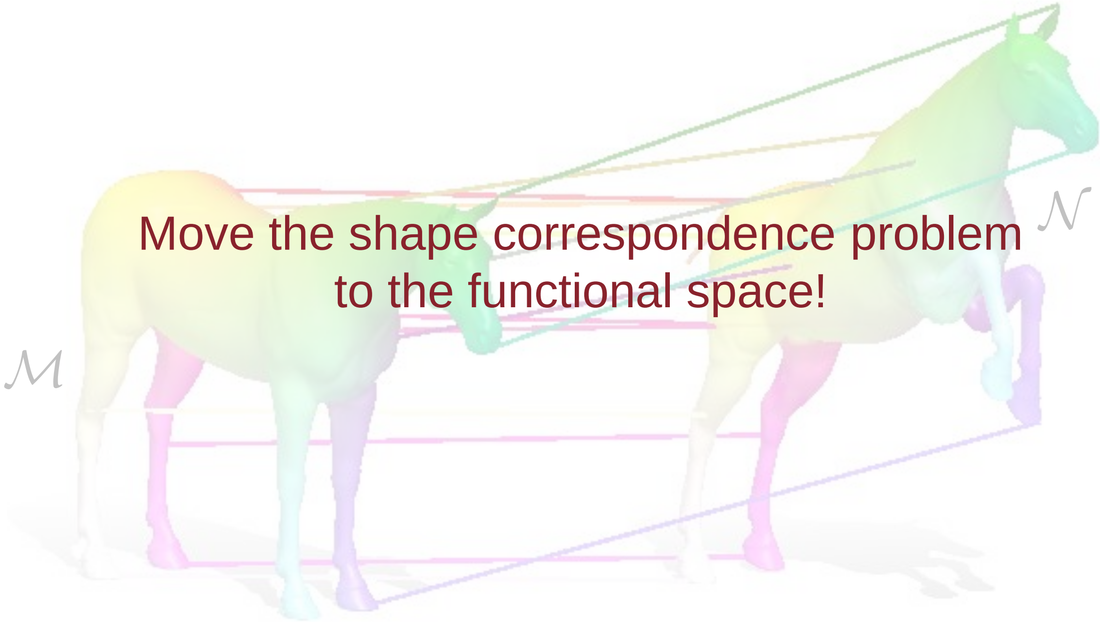
What is the relation between the two set of coefficients $\boldsymbol{a}$ and $\boldsymbol{b}$?

g 

Fourier

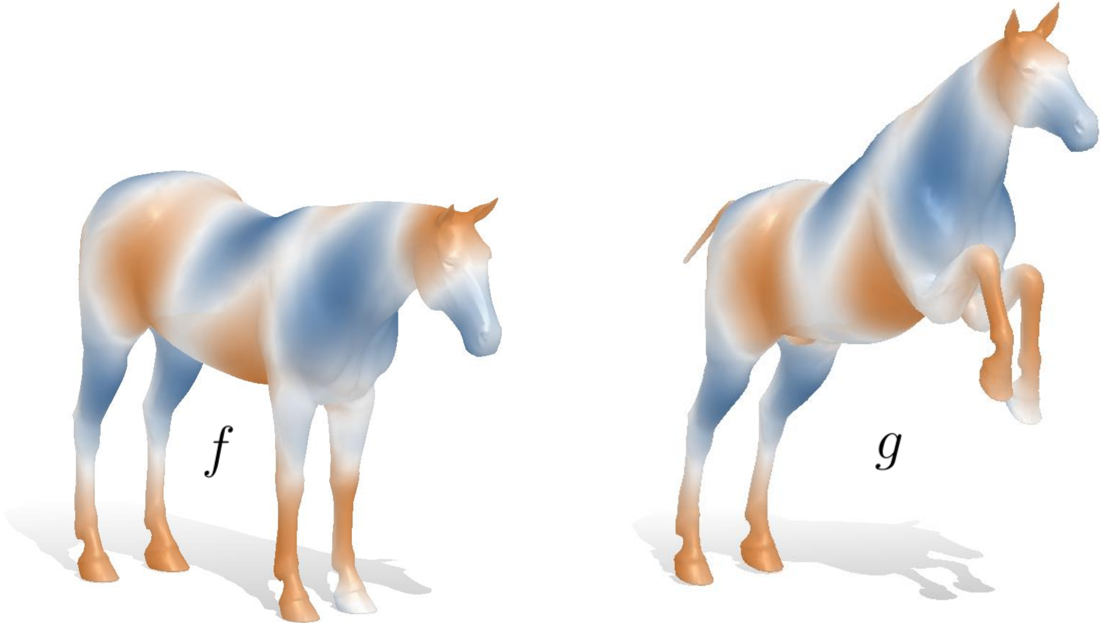
$$[\beta_1,\ \beta_2,\ \beta_3,\ \beta_4,\ \beta_5,\ \dots\ \beta_{29},\ \beta_{30}]^\top = \boldsymbol{b}$$

Answer: FUNCTIONAL MAPS



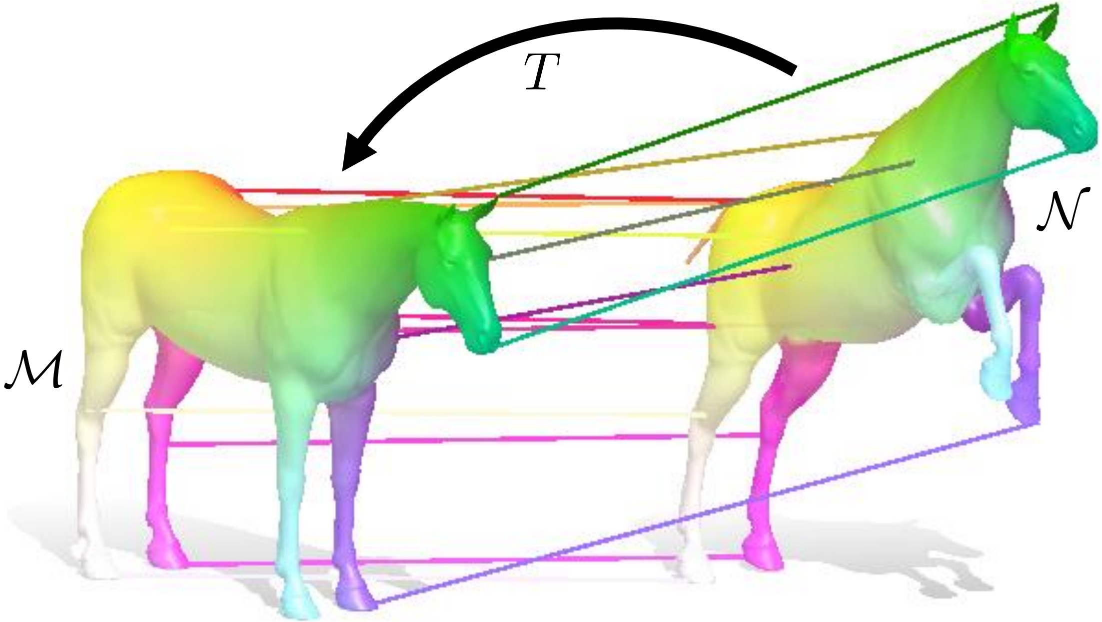
The problem to find a point-to-point map between $\mathcal{M}$ and $\mathcal{N}$

Corresponding functions



corresponding = arise from a point-to-point map

A point-to-point map



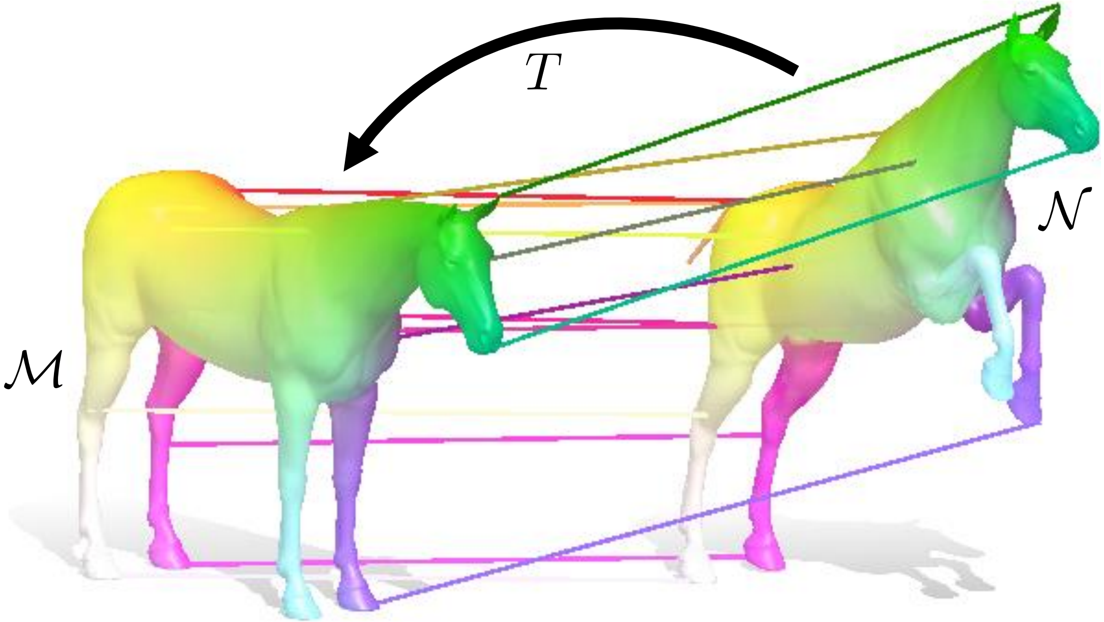
T is a point-to-point map

$$g(p) = f(T(p)) \; \forall p \in \mathcal{N}$$

Functional maps: a flexible representation of maps between shapes, *Ovsjanikov at al.*, 2012

A point-to-point map

$$T : \mathcal{N} \to \mathcal{M}$$



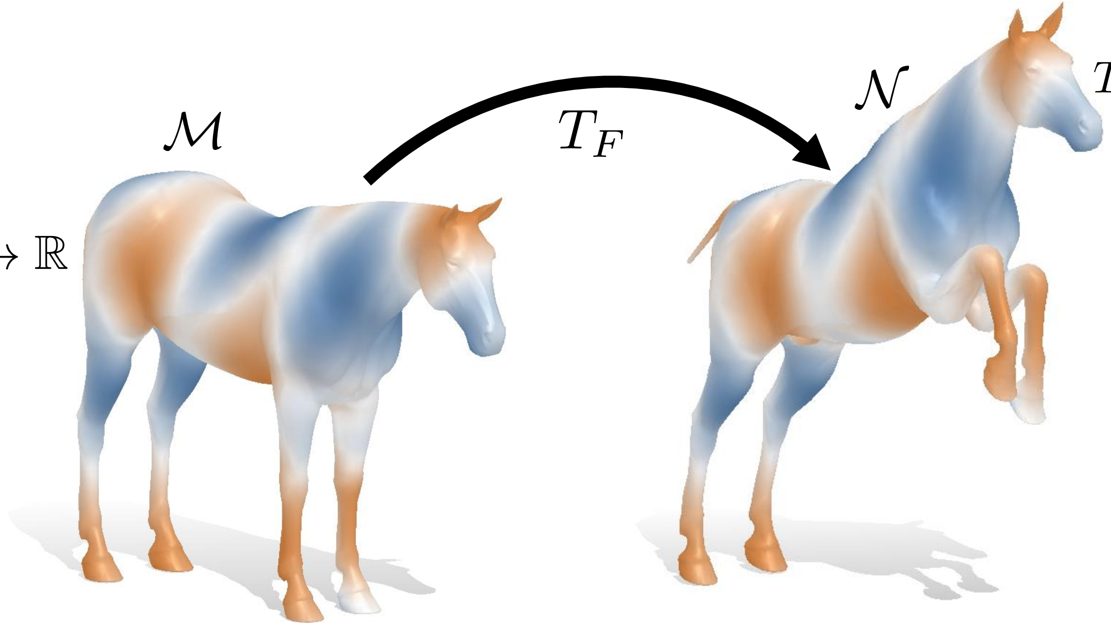
write it as a binary matrix

$$\Pi_{\mathcal{N}\mathcal{M}}(i,j) = 1 \Longleftrightarrow T(i) = j$$

Induces a functional map

$$T_F : F(\mathcal{M}) \longrightarrow F(\mathcal{N})$$

$f : \mathcal{M} \to \mathbb{R}$



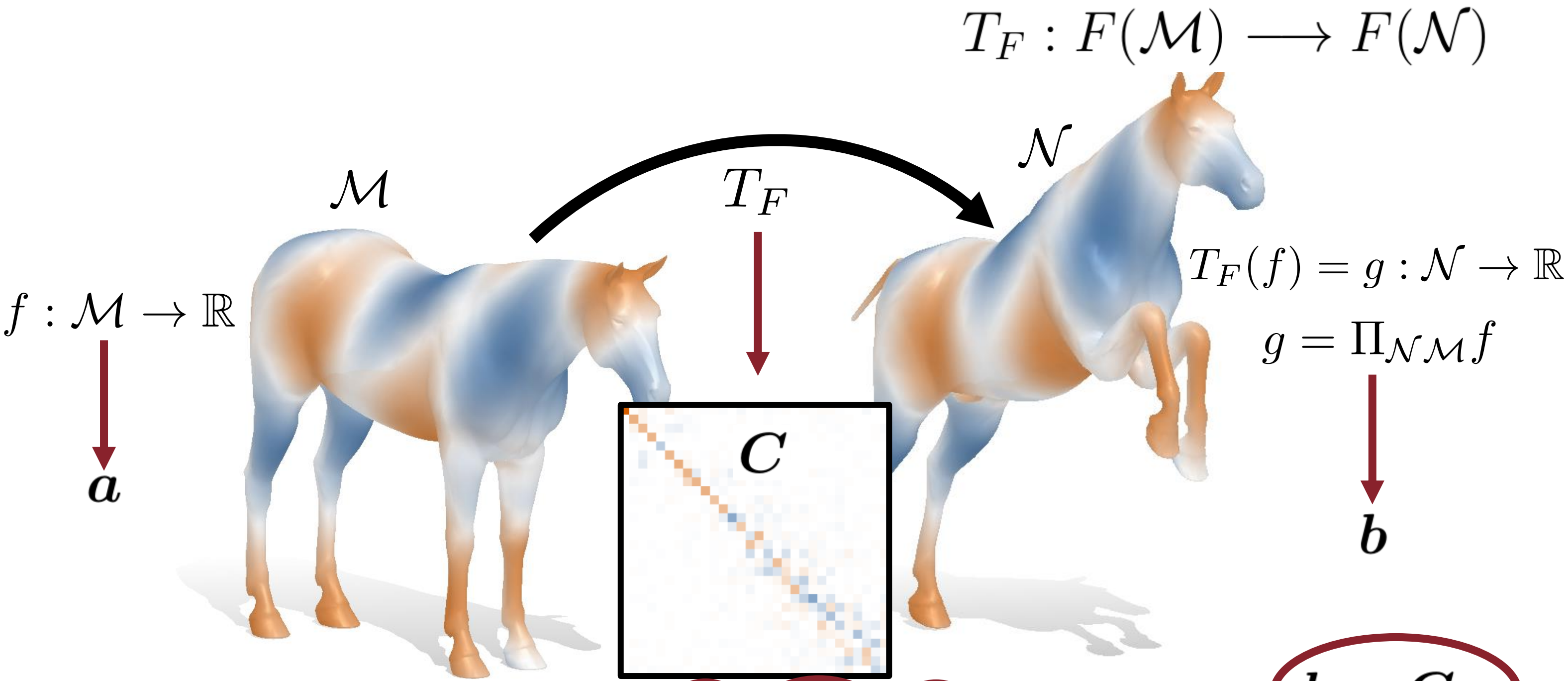
$T_F(f) = g : \mathcal{N} \to \mathbb{R}$

$g(y) = f(T(y))$

$$\mathcal{N} \xrightarrow{T} \mathcal{M}, \quad f : \mathcal{M} \to \mathbb{R}, \quad g : \mathcal{N} \to \mathbb{R}$$

The transfer is defined as: $g = \Pi_{\mathcal{N}\mathcal{M}} f$

Induces a functional map



$$\boldsymbol{C} = \boldsymbol{\Phi}_{\mathcal{N}}^{\dagger} \boldsymbol{\Pi}_{\mathcal{N}\mathcal{M}} \boldsymbol{\Phi}_{\mathcal{M}}$$

$$\boldsymbol{b} = \boldsymbol{C}\boldsymbol{a}$$

Exploiting Fourier

Functional maps: a flexible representation of maps between shapes, *Ovsjanikov et al.*, SIGGRAPH 2012

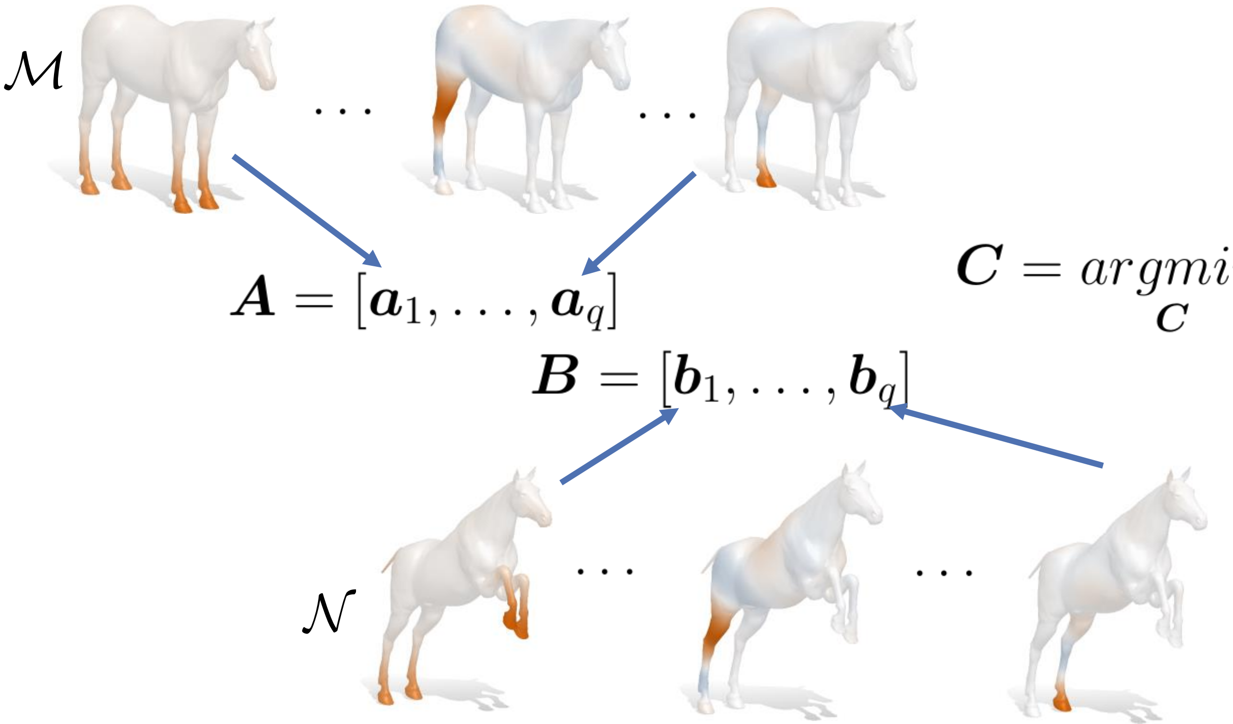
Summary so far:

- **Functions on meshes are vectors**
- **Corresponding functions** on different shapes depend on a **point-to-point correspondence**
- **A point-to-point map** induces a mapping between function (**a functional map**)
- **A point-to-point map** can be written as a binary matrix that operates on functions
- **Discrete Fourier basis:** Eigenvectors of the Laplacian
- We can represent the **functional map in the Fourier basis** which is:
    - **A small matrix** with the dimensions of the bases
    - **A linear operator** that maps **Fourier coefficients**
    - **Mainly diagonal** or close to it

Next Goal

Estimate the functional map associated to the correspondence between a pair of shapes

Functional maps estimation



$$\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_q]$$

$$\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_q]$$

$$\boldsymbol{C} = \underset{\boldsymbol{C}}{argmin} \|\boldsymbol{C}\boldsymbol{A} - \boldsymbol{B}\|^2 + \mathcal{R}(\boldsymbol{C})$$

Summary so far:

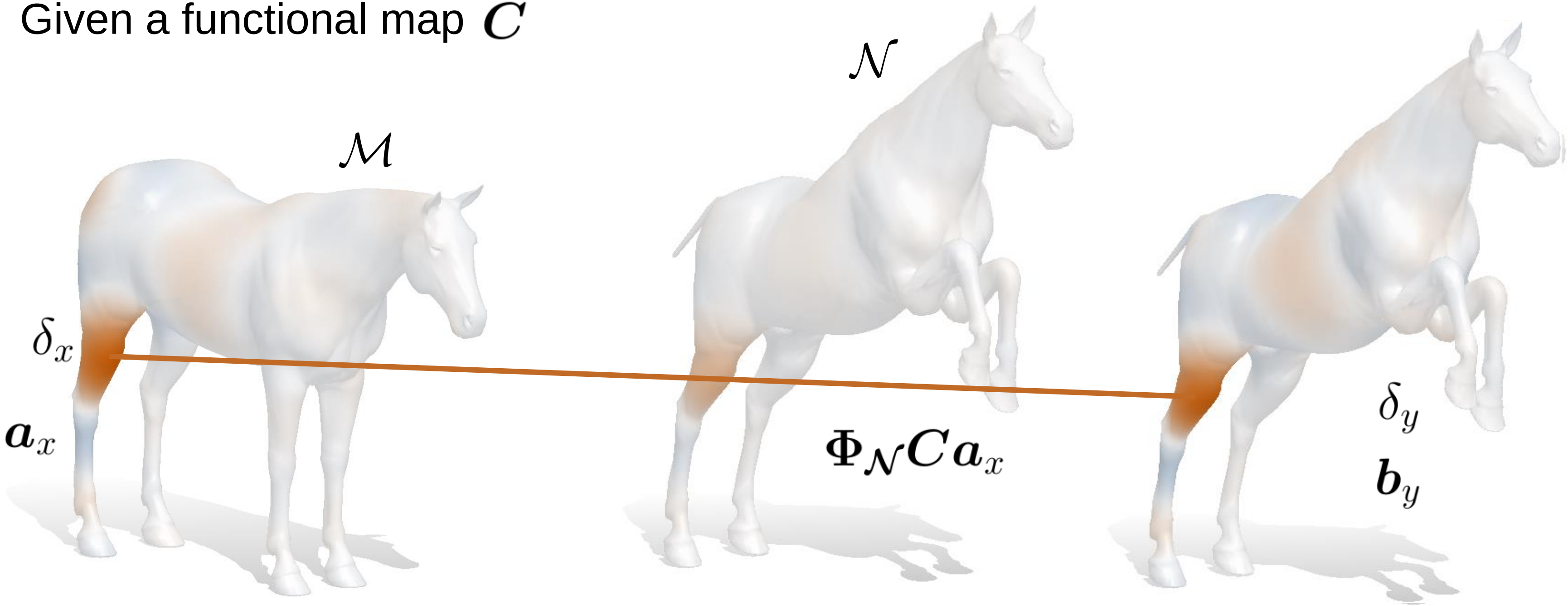
- **Functions on meshes are vectors**
- **Corresponding functions** on different shapes depend on a **point-to-point correspondence**
- **A point-to-point map** induces a mapping between function (**a functional map**)
- **A point-to-point map** can be written as a binary matrix that operates on functions
- **Discrete Fourier basis:** Eigenvectors of the Laplacian
- We can represent the **functional map in the Fourier basis** which is:
    - **A small matrix** with the dimensions of the bases
    - **A linear operator** that maps **Fourier coefficients**
    - **Mainly diagonal** or close to it
    - Can be estimated from a given set of **corresponding functions**

Next Goal

Obtain the correspondence from the functional map

Conversion to a point-to- point map

Given a functional map $\boldsymbol{C}$



$$T(x) = \underset{y}{argmin} \|\boldsymbol{b}_y - \boldsymbol{C}\boldsymbol{a}_x\|_2$$

Fmaps pipeline

1. Compute the first *k* (~30-100) eigenfunctions of the LBO.
   Store them in matrices: $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$

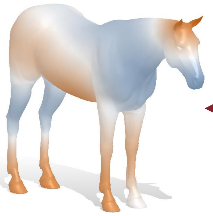
2. Compute probe functions (e.g., landmarks or descriptors) on $\mathcal{M}, \mathcal{N}$.
   Express them in $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$, as columns of $\boldsymbol{A}$ and $\boldsymbol{B}$

3. Solve $\underset{\boldsymbol{C}}{argmin} \|\boldsymbol{C}\boldsymbol{a} - \boldsymbol{b}\|_F^2 + \mathcal{R}(\boldsymbol{C})$

4. Convert the functional map to a point-to-point map *T*.

Question 2

f

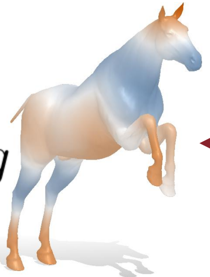


Fourier

$$[\alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4,\ \alpha_5,\ \ldots\ \alpha_{29},\ \alpha_{30}]^\top = \boldsymbol{a}$$

What is the relation between the two set of coefficients $\boldsymbol{a}$ and $\boldsymbol{b}$?

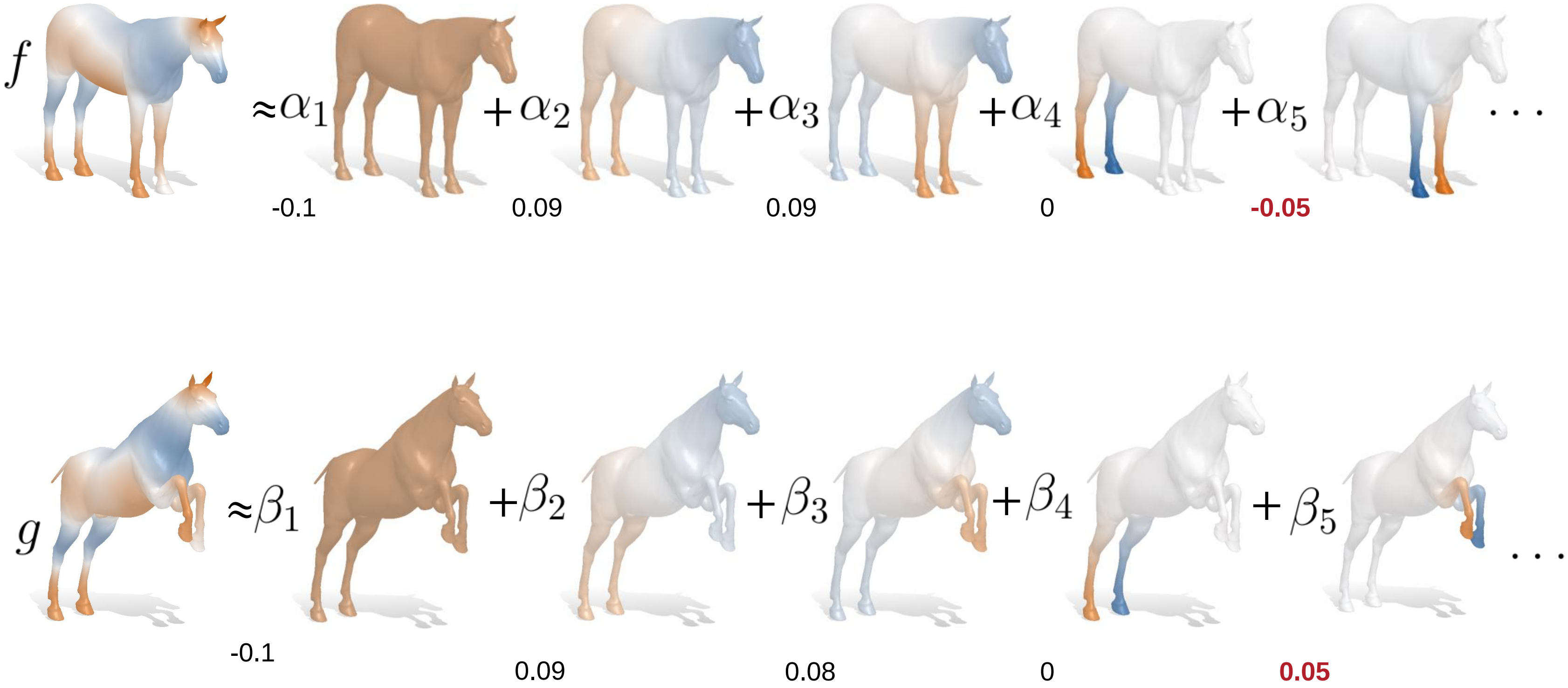
g



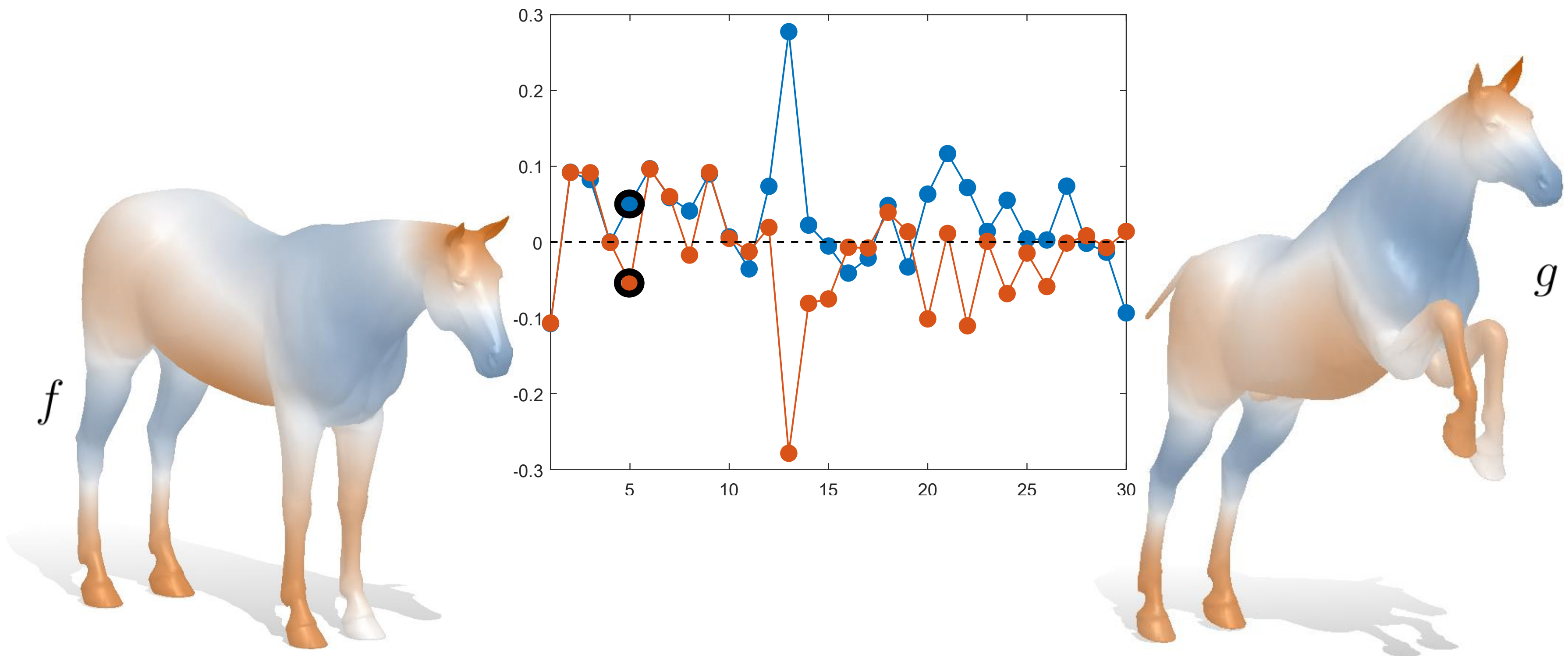
Fourier

$$[\beta_1,\ \beta_2,\ \beta_3,\ \beta_4,\ \beta_5,\ \ldots\ \beta_{29},\ \beta_{30}]^\top = \boldsymbol{b}$$

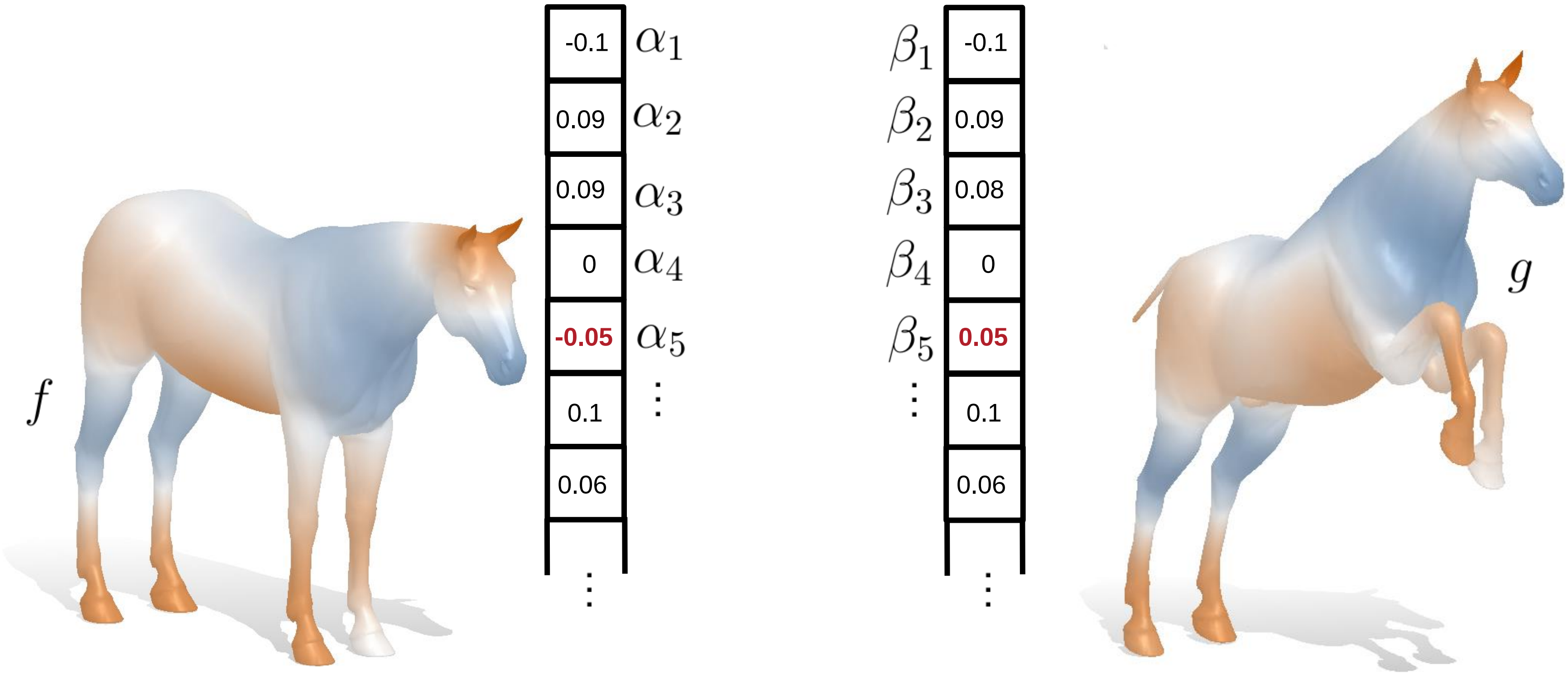
Fourier analysis and synthesis



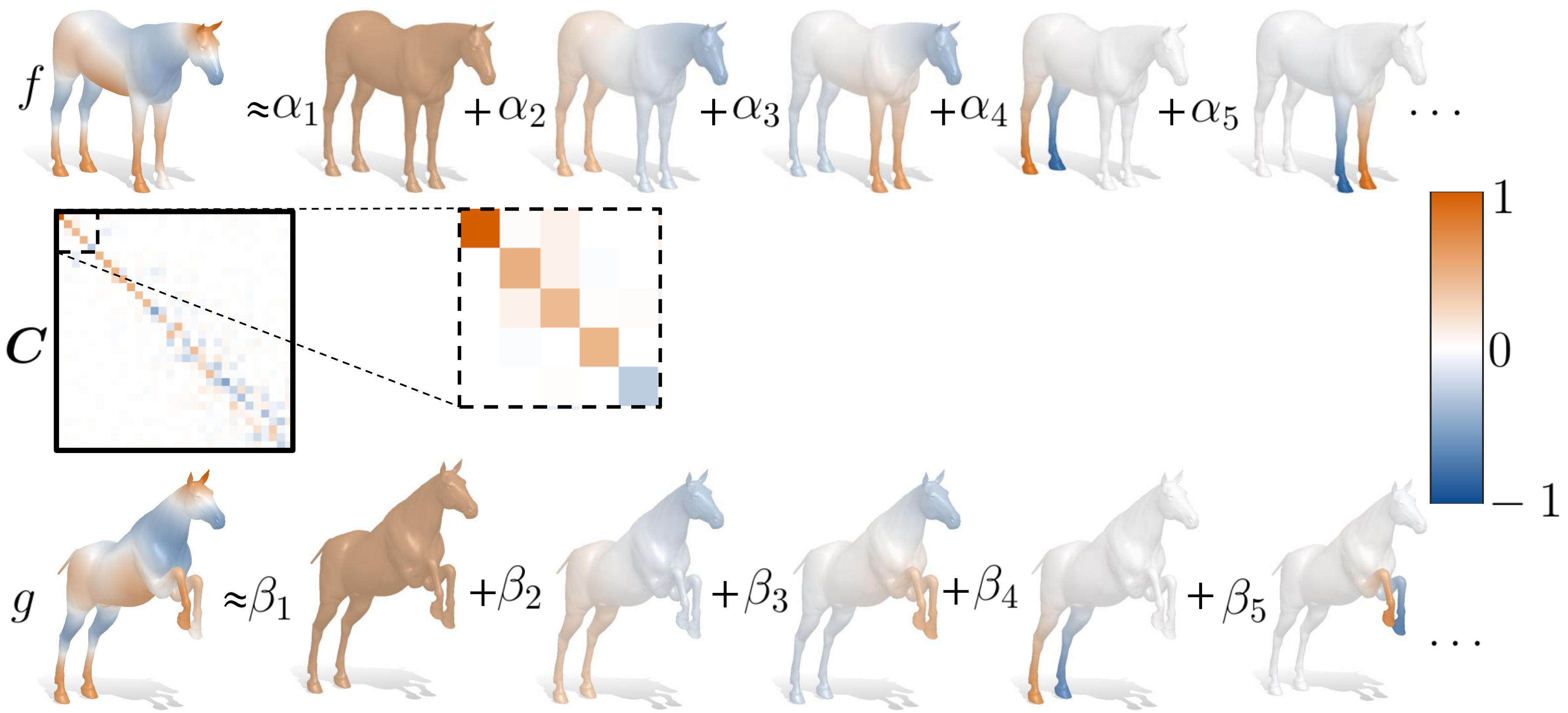
Compare the Fourier coefficients



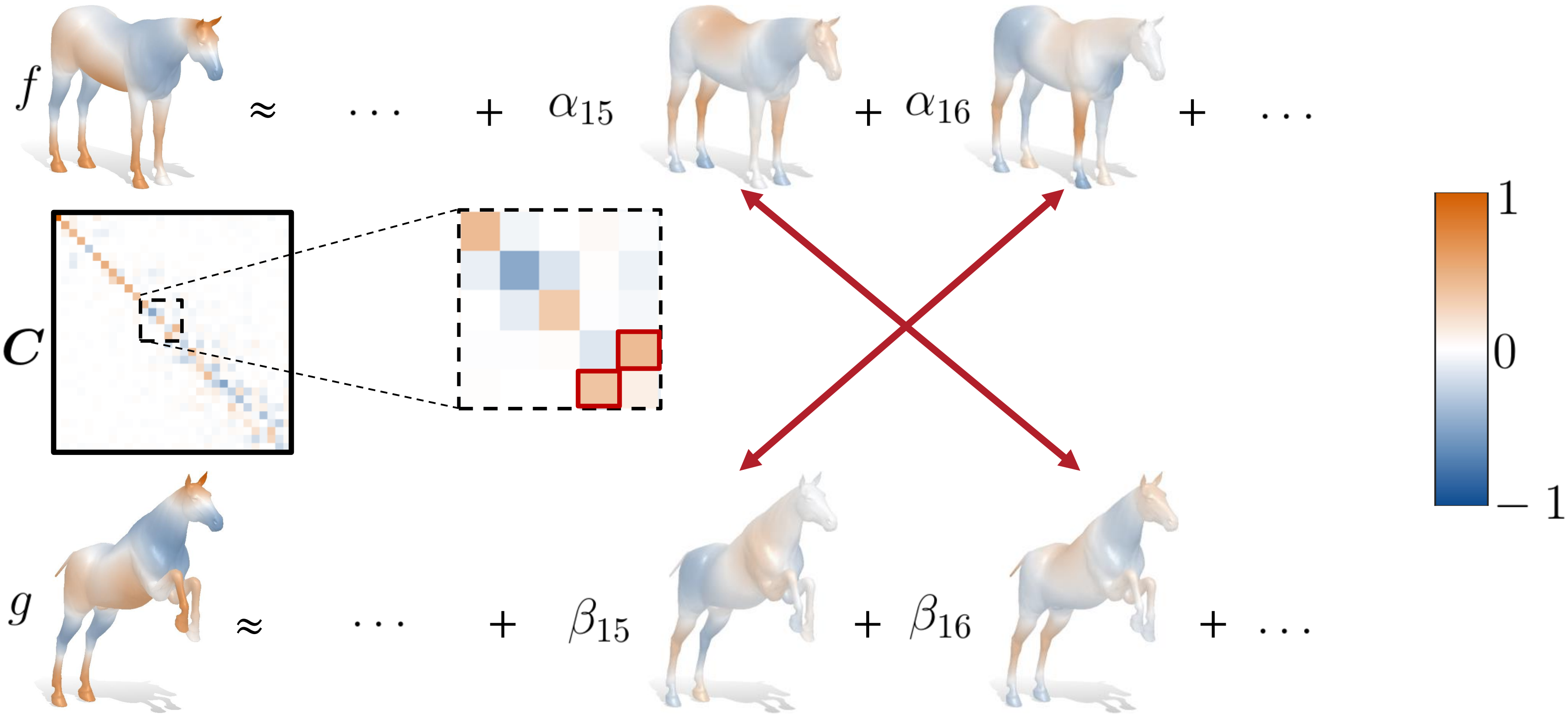
Compare the Fourier coefficients



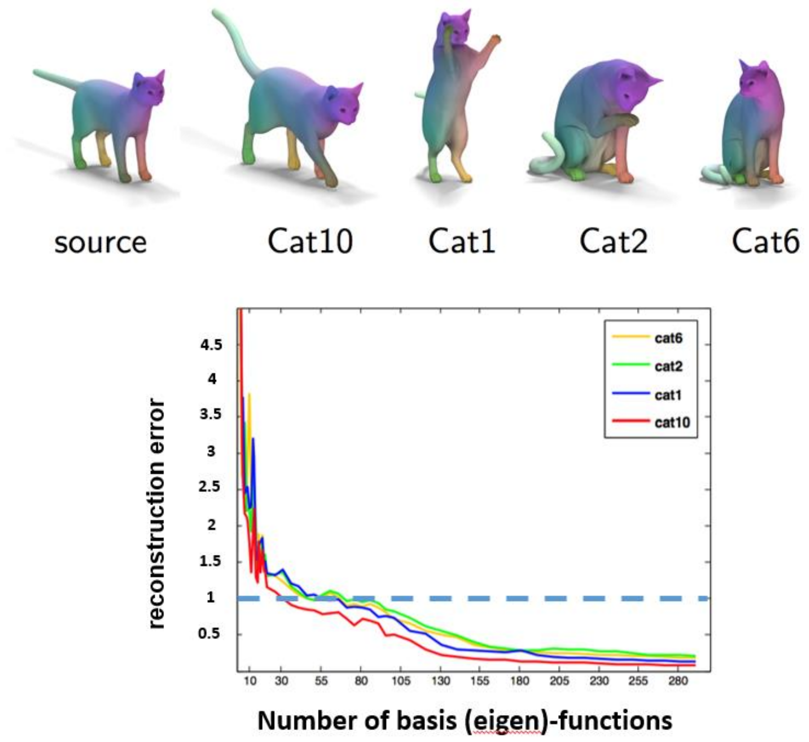
Functions on 2 different domains



Functions on 2 different domains



Functional map and the size of the basis



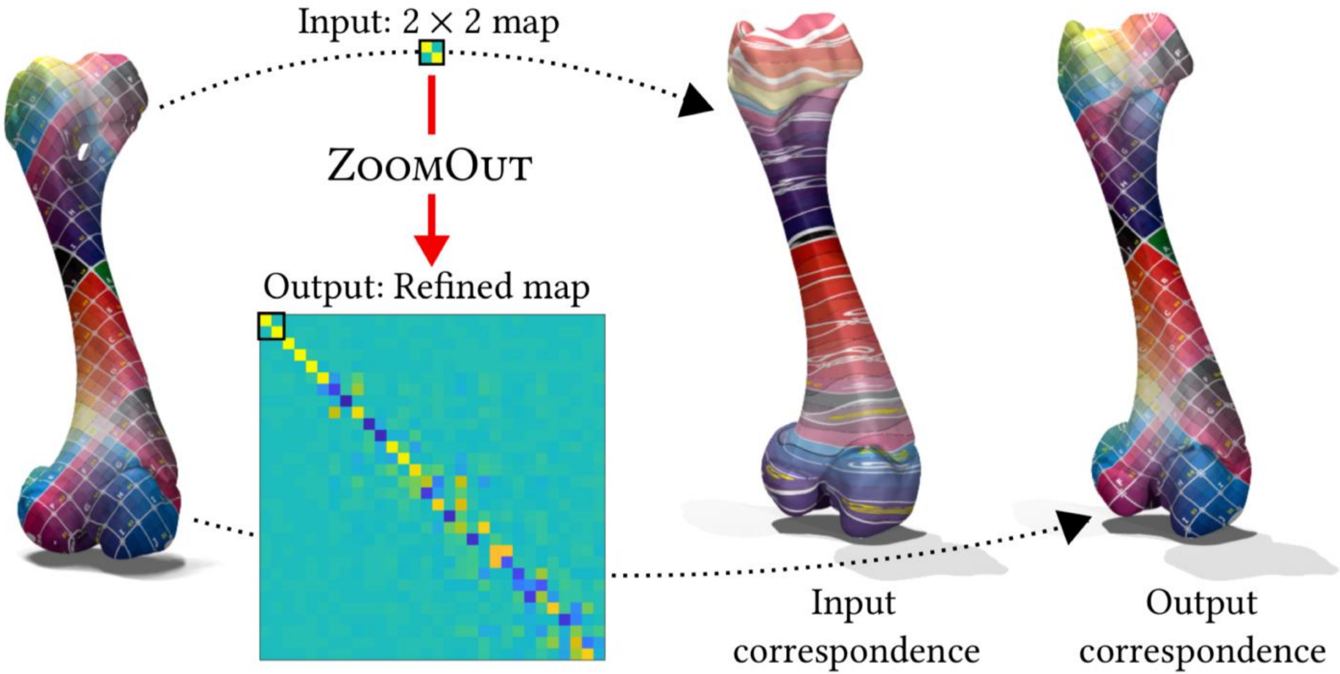
Slide credit M. *Ovsjanikov*

Functional maps: a flexible representation of maps between shapes, *Ovsjanikov et al.*, SIGGRAPH 2012

The Functional Maps trade-off

Low-pass	**Easy to optimize (fewer probe functions needed)**	Low-pass representation (poor functional transfer)
High-frequencies	**Better representation of details (good transfer)**	Hard to optimize (more probe functions needed)

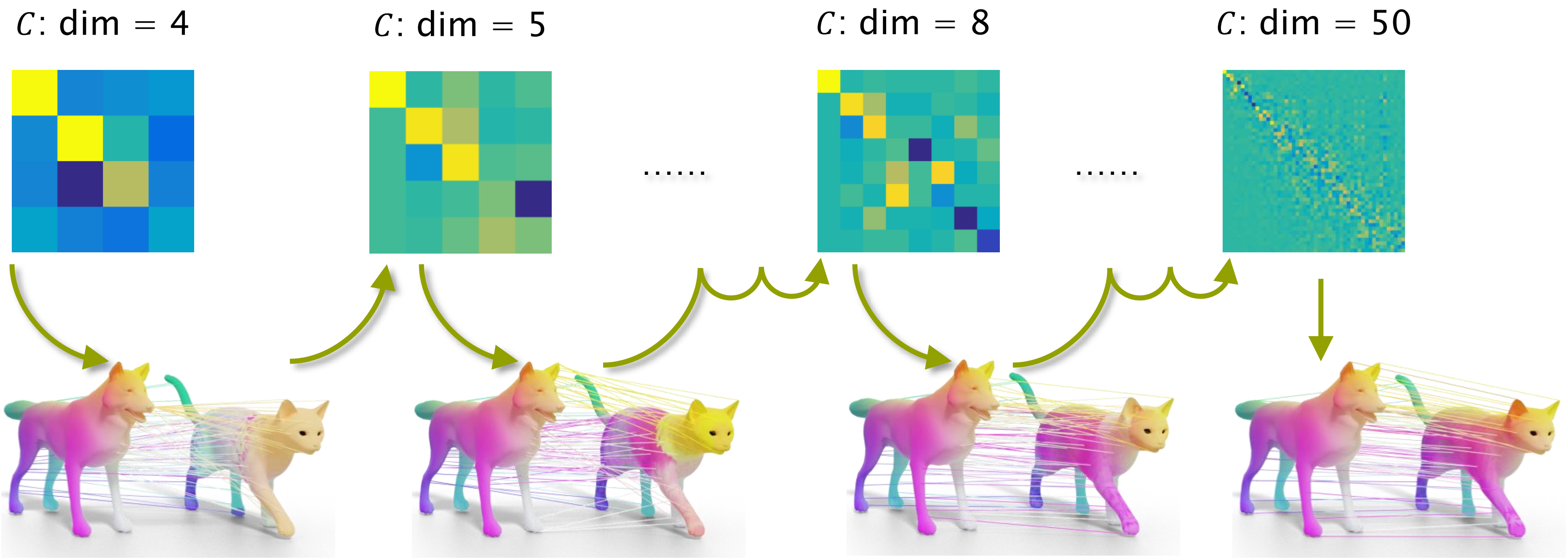
Exploiting the connection with point-to-point-map



[ZoomOut: Spectral Upsampling for Efficient Shape Correspondence, *Melzi* et al., 2019]

ZoomOut a visualization

Slide credit J. *Ren*



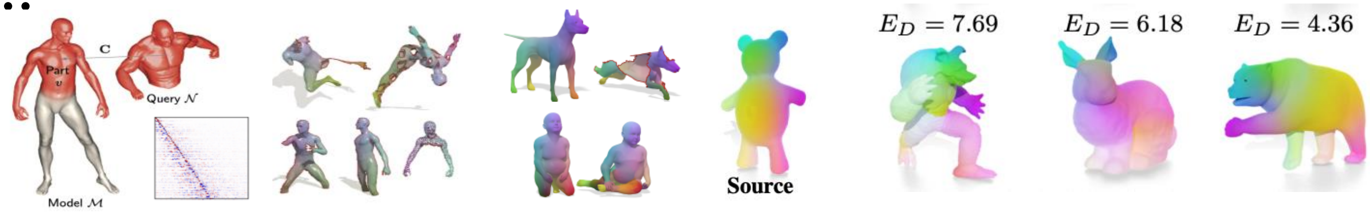
ZoomOut: Spectral Upsampling for Efficient Shape Correspondence, *Melzi* et al., 2019

Functional Map Improvements

Significant improvements to functional maps over the years:

1. Functional maps in shape and image *collections*.
2. Enabling *partial* shape matching.
3. Better understanding of *pointwise* map recovery.
4. Much better *symmetry* handling.
5. Techniques for promoting map continuity and smoothness.

…



A rich tool box for shape analysis and correspondence problems!

LBO and spectral stuff

https://colab.research.google.com/drive/1MhfzQBmPj2VMORPUERVzCgflpmLbvEVo#scrollTo=jpwcR2iGoYwn

LBO, WKS and Functional Maps DEMO

https://github.com/RobinMagnet/pyFM

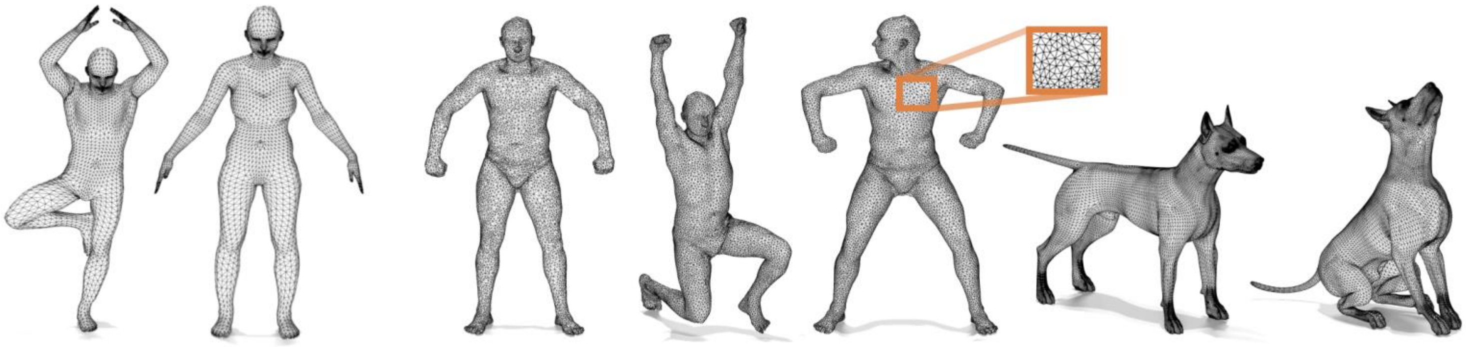
LEARNING ON GEOMETRIC DATA

1. Introduction: 3D Non-Rigid shapes and registration
2. Spectral representation
3. Axiomatic approaches
4. Functional maps
5. **Learning on geometric data**
6. Learning–based Functional maps
7. Other learning-based approaches
8. Transformers

Common datasets

Pros:
- Clean, manifold triangle meshes with ground truth maps

Cons:
- Most existing datasets are *synthetic*
- Shapes within a dataset are in 1-1 correspondence
- Scale is typically limited



FAUST/DFAUST/SURREAL

SCAPE

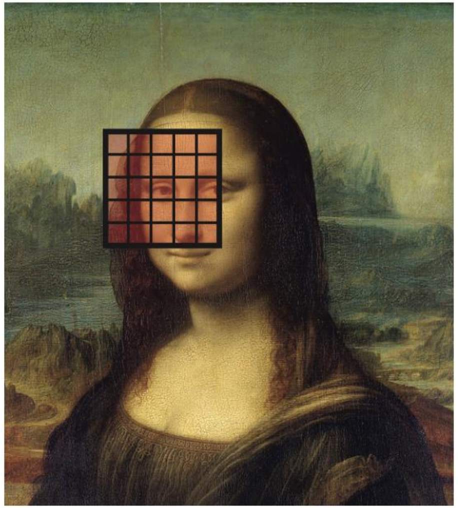
TOSCA

Bogo, Federica, et al., *FAUST: Dataset and evaluation for 3D mesh registration*. CVPR 2014
Anguelov, Dragomir, et al., *SCAPE: shape completion and animation of people*, SIGGRAPH 2005
Bronstein, Alexander et al., *Numerical geometry of non-rigid shapes*, Springer 2008

Non-Euclidean learning

image credit M.Bronstein

Idea: apply kernels directly on the surface!



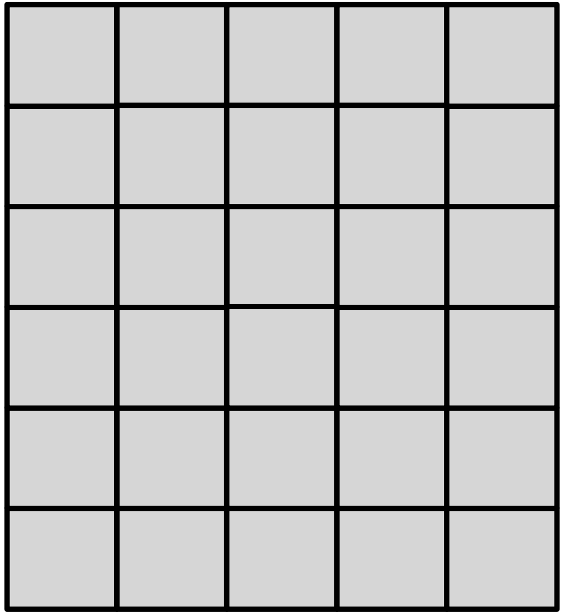
Euclidean

Non-Euclidean

Geometric Deep Learning: Going Beyond Euclidean Data Bronstein MM et al. 2017

A Comprehensive Survey on Geometric Deep Learning. Cao W, Yan Z, He Z, He Z. 2020

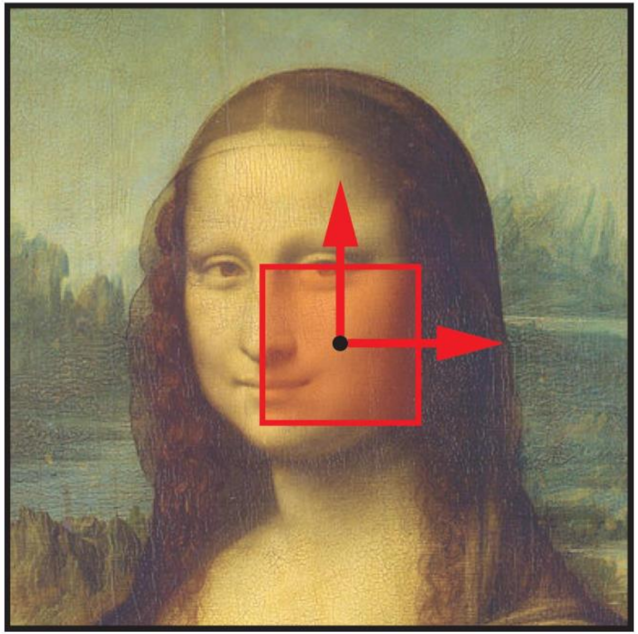
Image vs Geometry



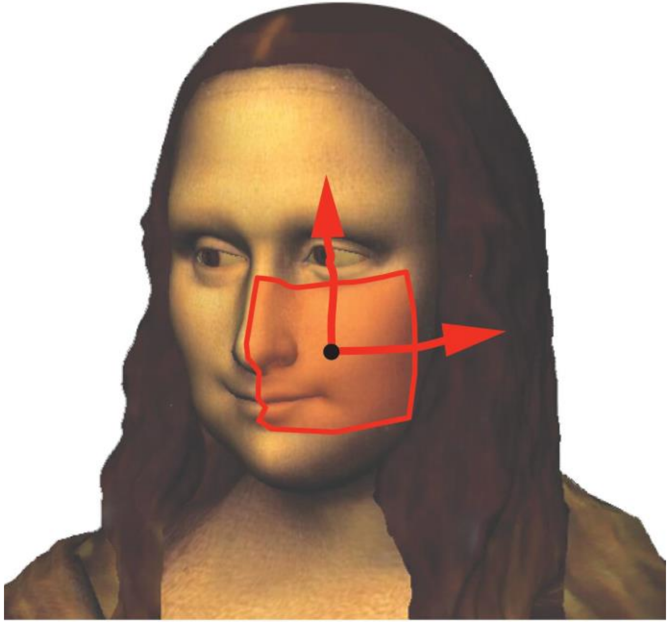
2D
only signal
fixed domain/template

3D
only domain
constant or no signal

Non-Euclidean convolution:



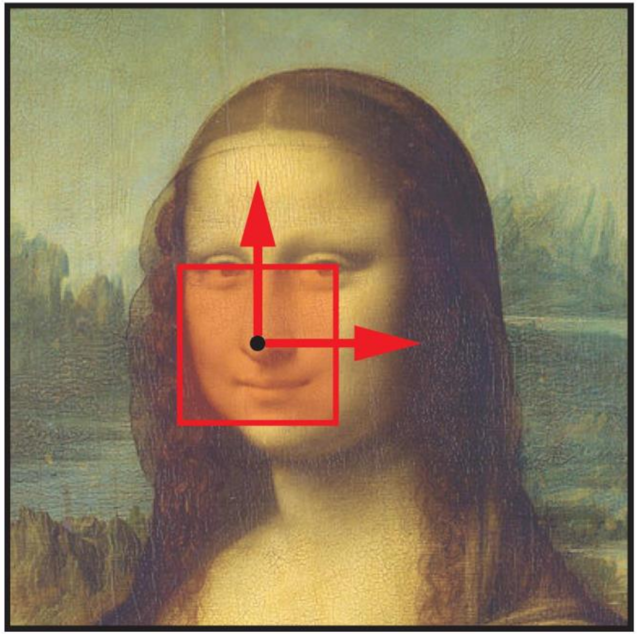
Euclidean



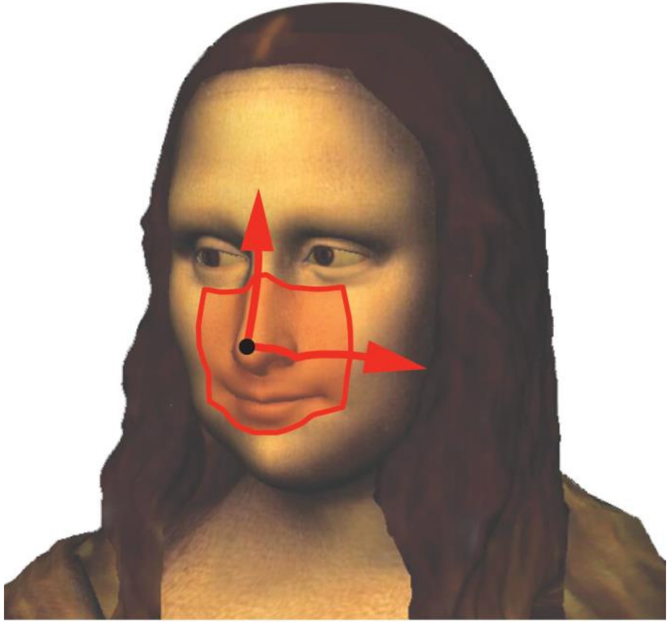
Non- Euclidean

Slide credit E. *Rodolà*

Non-Euclidean convolution:



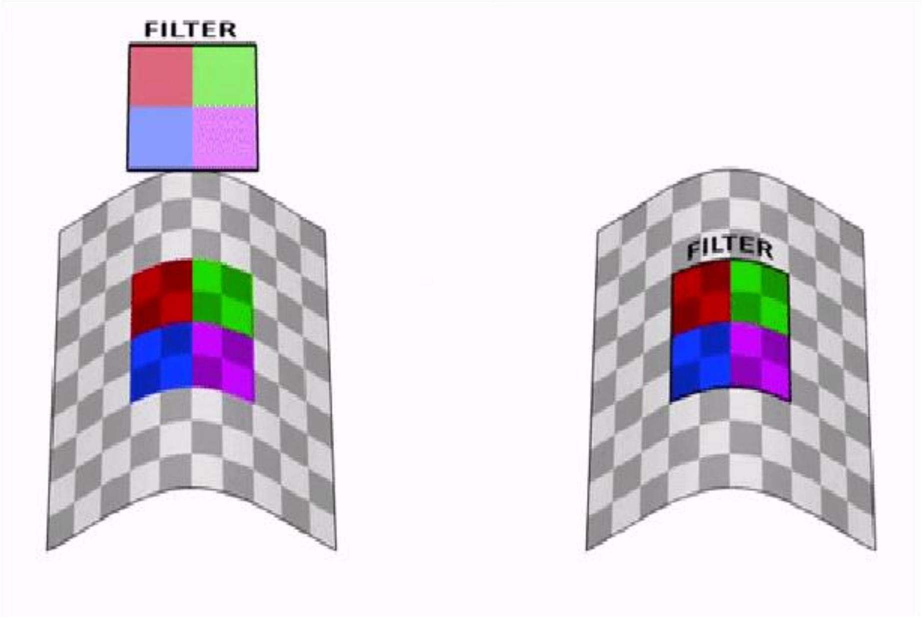
Euclidean



Non- Euclidean

Slide credit E. *Rodolà*

Intrinsic vs extrinsic:



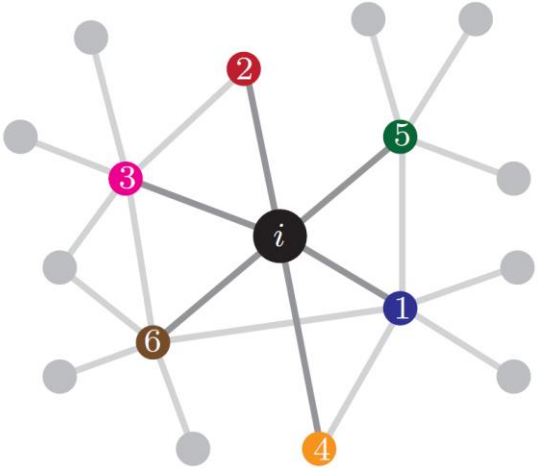
Extrinsic

Intrinsic

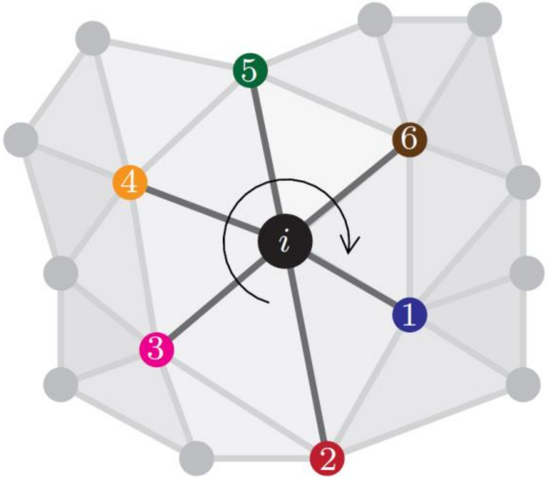
Slide credit E. *Rodolà*

Local ambiguity

Unlike images, there is not a canonical ordering of the points in the domain



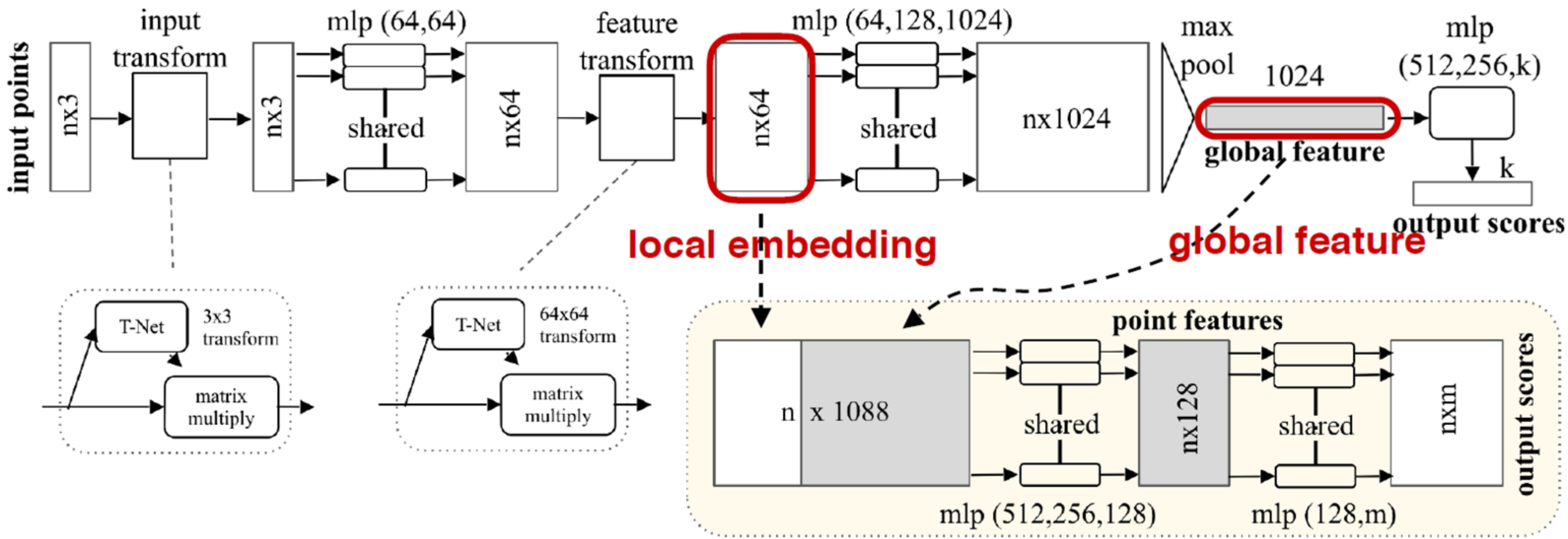
Graph rotation



Mesh rotation

Slide credit E. *Rodolà*

PointNET pointwise feature:



"PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*

Geodesic convolutional neural networks

Key idea: parameterize the shape *locally.*

Using local polar coordinates on the surface can multiply the signal *f* with a trainable kernel *g*

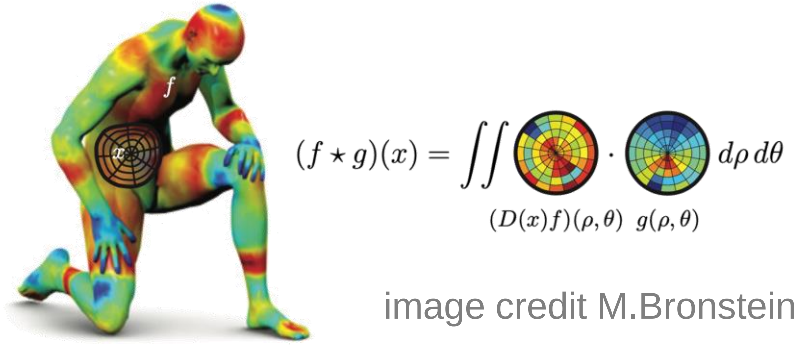


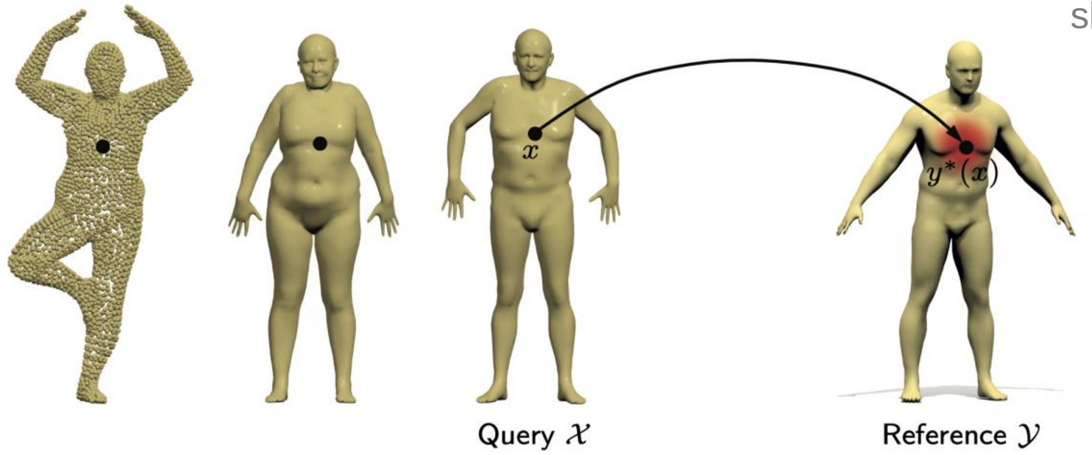
image credit M.Bronstein

Product is a scalar per point ⇒ a real-valued function on the surface. **Pose invariant!**

- Geodesic convolutional neural networks on Riemannian manifolds, Masci et al., 2015
- Learning shape correspondence with anisotropic convolutional neural networks, Boscaini et al., 2016
- Geometric deep learning on graphs and manifolds using mixture model CNNs, Monti et al., 2017
- …

Learning Correspondences with GCNN

slide credit E. Rodolà



- Correspondence = labeling problem
- GCNN output $\mathbf{f}_{\mathbf{\Theta}}(x)$ = probability distribution on reference $\mathcal{Y}$
- Minimize logistic regression cost w.r.t. GCNN parameters $\mathbf{\Theta}$

$$\ell(\mathbf{\Theta}) = -\sum_{(x,y^*(x))\in\mathcal{T}} \langle \delta_{y^*(x)}, \log \mathbf{f}_{\mathbf{\Theta}}(x)\rangle_{L^2(\mathcal{Y})}$$

Geodesic convolutional neural networks on Riemannian manifolds, Masci et al. 2015

Correspondence learning via ASCNN

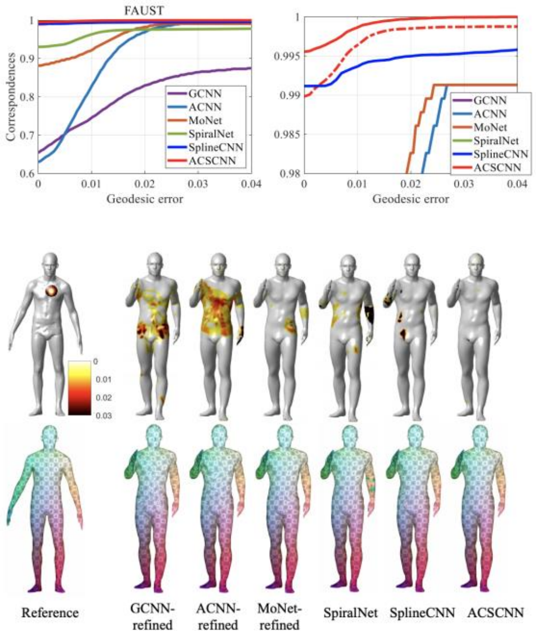


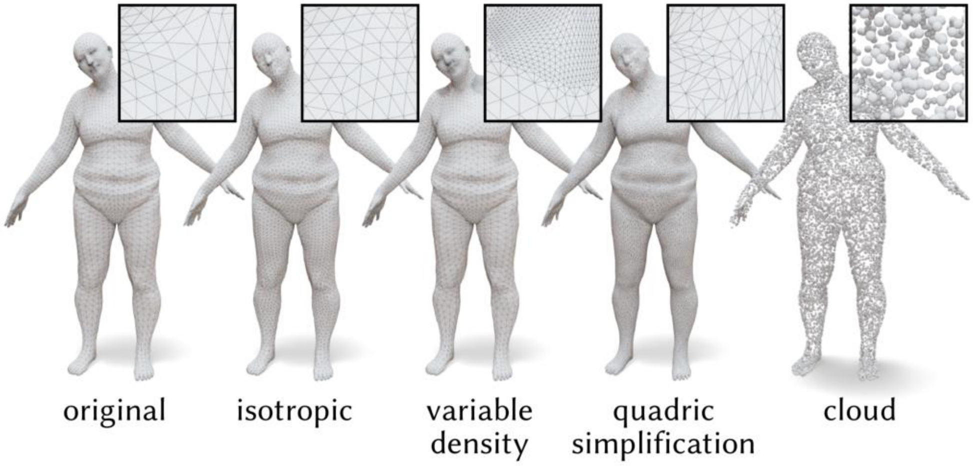
Table 1. Performance comparisons on FAUST dataset.

Method	Refinement	Input	Accuracy (r=0)	Accuracy (r=0.01)
GCNN [29]		SHOT	66.61 %	74.98 %
ACNN [7]	FM[36]	SHOT	62.40 %	83.31 %
MoNet [34]	PMF[49]	SHOT	88.20 %	92.35 %
SpiralNet [27]		SHOT	93.06 %	96.32 %
ACSCNN		SHOT	98.06 %	99.26 %
SplineCNN [19]		1	99.12 %	99.37 %
ACSCNN		1	98.98 %	99.64 %
ACSCNN	PMF[49]	1	99.56 %	99.87 %

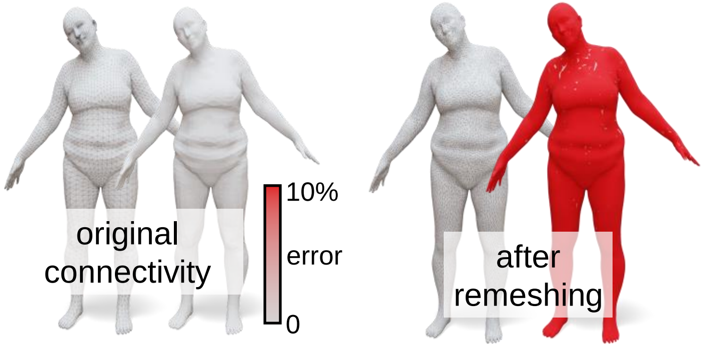
Correspondence is solved!

Li, Qinsong, et al. *Shape correspondence using anisotropic Chebyshev spectral CNNs*, CVPR 2020.

What happens under remeshing?



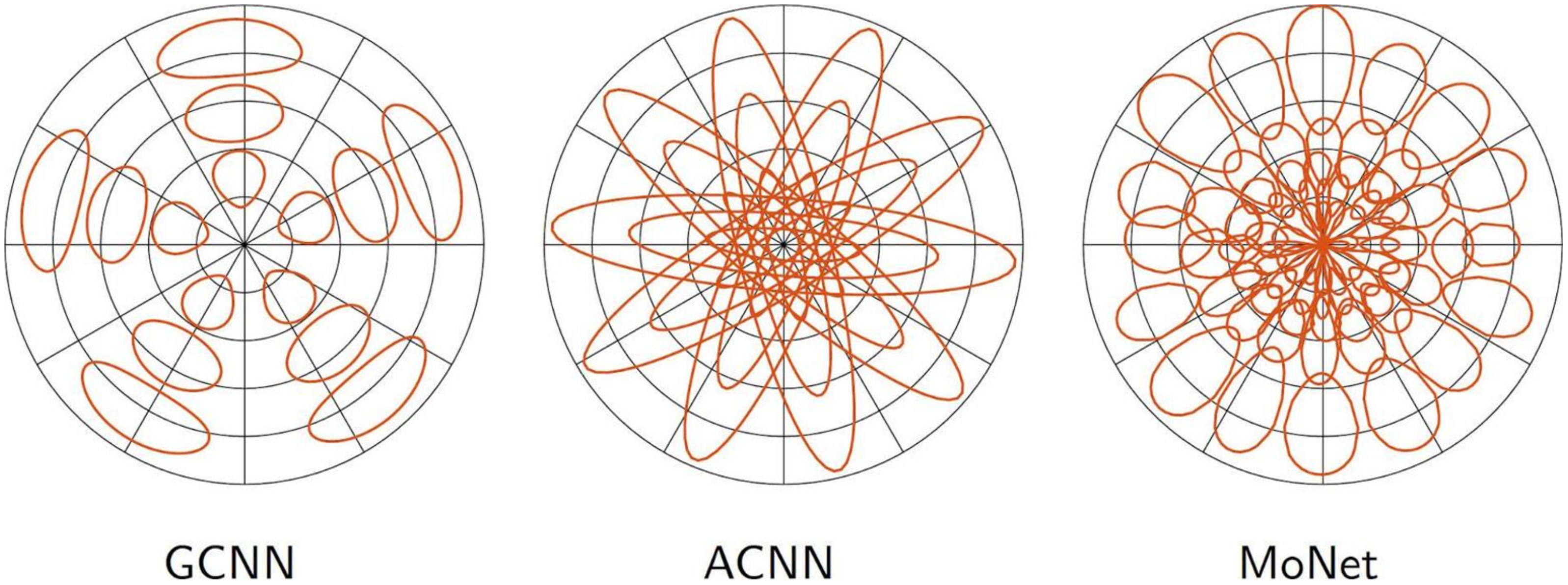
		remeshed/sampled variants			
Method	**orig**	**iso**	**dense**	**qes**	**cloud**
ACSCNN	**0.05**	35.29	19.09	41.15	-
SplineCNN	3.51	31.09	27.95	40.43	-
HSN	9.57	20.01	24.84	25.40	-



ASCNN correspondence error

Spatial convolution filters

Slide credit E. *Rodolà*



- Loss is *independent* of the geometry
- Requires a template, difficult to generalize to new classes
- Difficult to obtain discretization independent results
- Requires dense ground truth maps

"Geometric deep learning on graphs and manifolds using mixture model CNNs", Monti et al., 2016

Theorem of convolution

Convolution on the Euclidean domain $[-\pi, \pi]$ of two functions $f, g\colon [-\pi, \pi] \to \mathbb{R}$ is defined as:

$$(f \star g)(x) = \int_{-\pi}^{\pi} f(x')g(x - x')dx'$$

Convolution theorem: Fourier transform diagonalizes the convolution operator.

More explicitly: convolution can be computed in the spectral (Fourier) domain:

$$\left(\widehat{f \star g}\right) = \hat{f} \cdot \hat{g}$$

coefficients of the convoluted signal = the frequency-wise product of the coefficients

Spectral representation of the convolution operator

We can define a truncated fourier basis $\Phi = [\phi_1, \dots, \phi_K]$

We can encode the analysis and synthesis operators respectively in this basis

$$\mathcal{G}\underline{f} = \underline{f} \star \underline{g} = \Phi\left(\widehat{\underline{f} \star \underline{g}}\right) = \Phi\left(\underline{\hat{f}} \cdot \underline{\hat{g}}\right) = \Phi\hat{\mathcal{G}}\begin{bmatrix}\underline{\hat{f}_1} \\ \vdots \\ \underline{\hat{f}_K}\end{bmatrix} = \Phi\hat{\mathcal{G}}\Phi^\dagger \underline{f}$$

$$\begin{bmatrix}\underline{\hat{f}_1} \\ \vdots \\ \underline{\hat{f}_K}\end{bmatrix} = \Phi^\dagger \underline{f}$$

convolution theorem

$$\hat{\mathcal{G}} = diag\left(\underline{g}\right) = \begin{bmatrix}\underline{\hat{g}_1} & 0 & \dots & 0 \\ 0 & \underline{\hat{g}_2} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \underline{\hat{g}_K}\end{bmatrix}$$

$$\mathcal{G}\underline{f} = \Phi\hat{\mathcal{G}}\Phi^\dagger \underline{f} \longrightarrow \mathcal{G} = \Phi\hat{\mathcal{G}}\Phi^\dagger$$

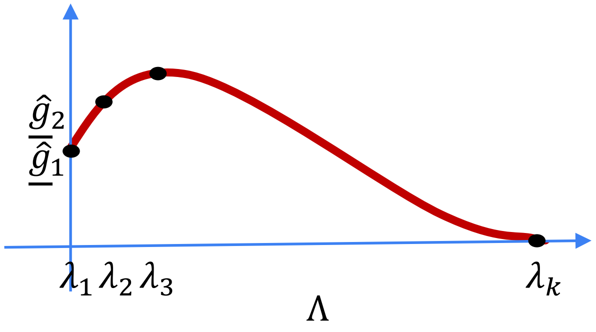
Spectral convolution

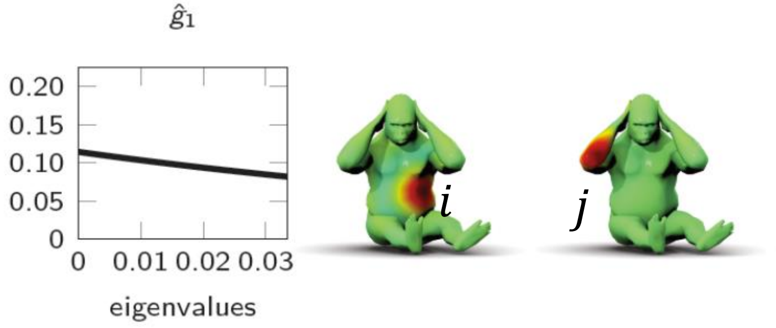
Spectral convolutional layer (Given Δ, Φ, Λ)

$$\underline{\hat{f}} = \Phi^{\dagger}\underline{f}$$

$$\underline{f'} = \Phi \mathrm{T}(\Lambda)\underline{\hat{f}}$$

$$T(\Lambda) = diag(\tau(\Lambda))$$



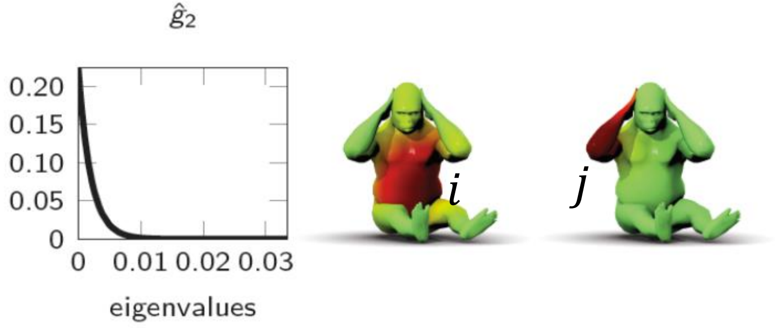


Define one window for each node is inefficient

Spectral Translation given $\underline{\delta_i}$ indicator of i

$$T_i\underline{g} = \underline{g} \star \underline{\delta_i} \qquad \widehat{T_i\underline{g}} = \widehat{\underline{g} \star \underline{\delta_i}} = \underline{\hat{g}} \odot \underline{\hat{\delta}_i}$$

$$\underline{\hat{\delta}_i} = \Phi(i) = [\phi_1(i), \ldots, \phi_K(i)]$$



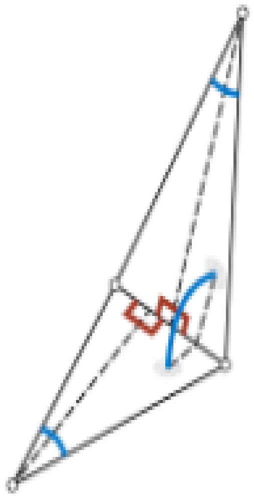
“Spectral Networks and Locally Connected Networks on Graphs", Bruna et al., 2014

“Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks", Boscaini et al., 2016

MeshCNN

Idea: edges instead of vertices are analogous to pixels

Input: 5 dimensional vector for each edge



- the dihedral angle
- two inner angles
- two edge-length ratios for each face

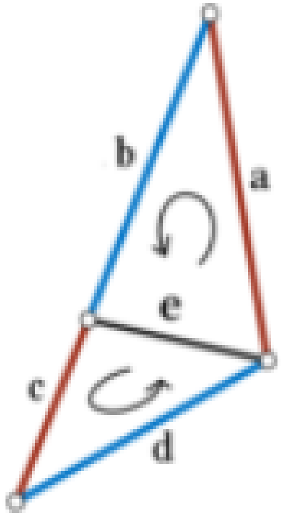
“MeshCNN: A Network with an Edge" Hanocka et al., 2019

MeshCNN

Idea: edges instead of vertices are analogous to pixels

Input: 5 dimensional vector for each edge

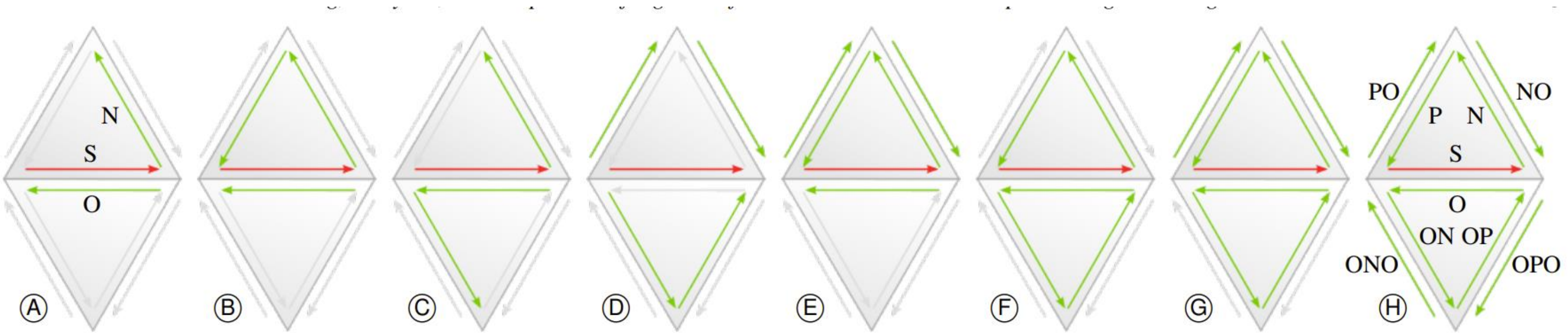
Convolution: Well defined for each edge and its Neighbour given by the 4 connected edges.

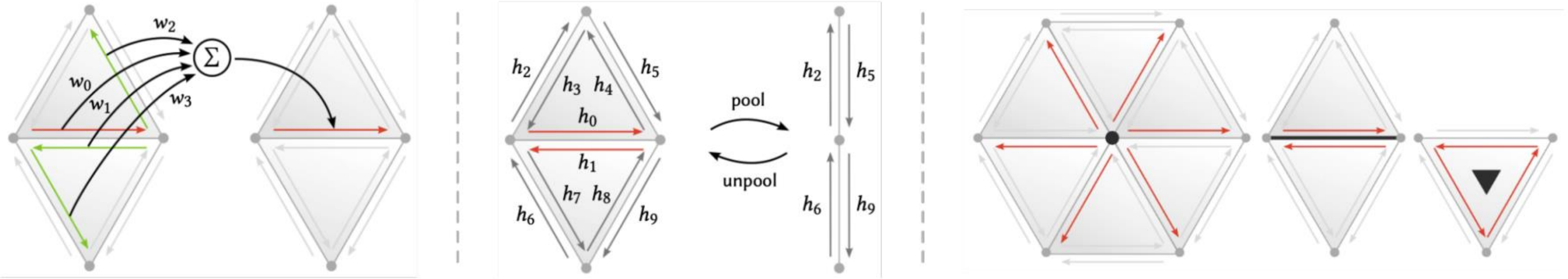


Mesh Convolution

"MeshCNN: A Network with an Edge" Hanocka et al., 2019

Half Edge CNN

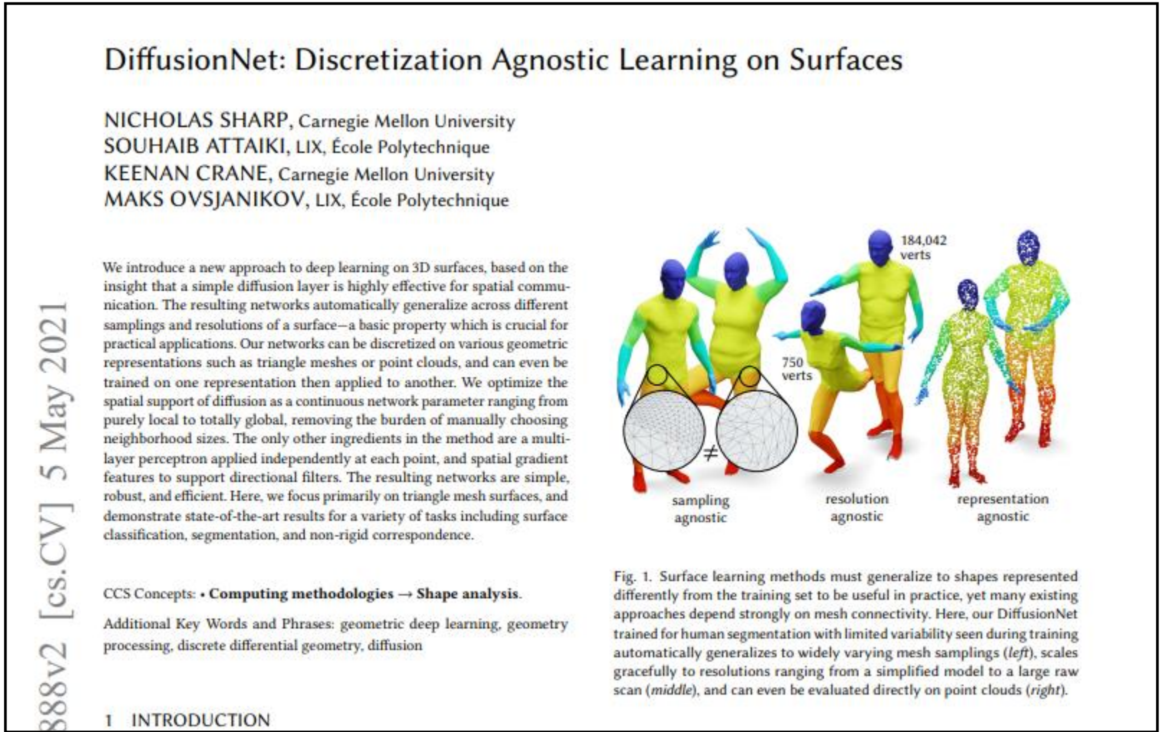




"HalfedgeCNN for Native and Flexible Deep Learning onTriangle Meshes" Ludwig et al., 2023

DiffusionNet

"DiffusionNet: Discetization agnostic learning on surfaces" Sharp et al., 2021



DiffusionNet: Discretization Agnostic Learning on Surfaces

NICHOLAS SHARP, Carnegie Mellon University
SOUHAIB ATTAIKI, LIX, École Polytechnique
KEENAN CRANE, Carnegie Mellon University
MAKS OVSJANIKOV, LIX, École Polytechnique

We introduce a new approach to deep learning on 3D surfaces, based on the insight that a simple diffusion layer is highly effective for spatial communication. The resulting networks automatically generalize across different samplings and resolutions of a surface—a basic property which is crucial for practical applications. Our networks can be discretized on various geometric representations such as triangle meshes or point clouds, and can even be trained on one representation then applied to another. We optimize the spatial support of diffusion as a continuous network parameter ranging from purely local to totally global, removing the burden of manually choosing neighborhood sizes. The only other ingredients in the method are a multi-layer perceptron applied independently at each point, and spatial gradient features to support directional filters. The resulting networks are simple, robust, and efficient. Here, we focus primarily on triangle mesh surfaces, and demonstrate state-of-the-art results for a variety of tasks including surface classification, segmentation, and non-rigid correspondence.

CCS Concepts: • **Computing methodologies → Shape analysis**.

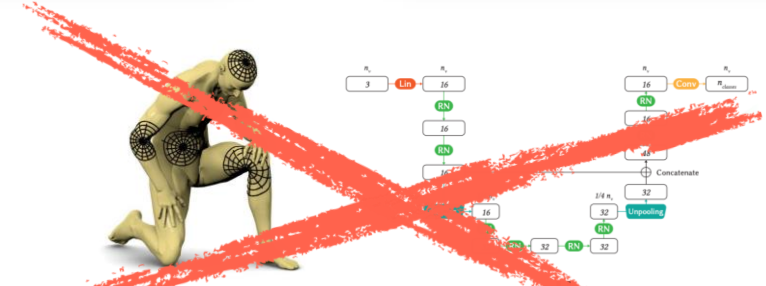
Additional Key Words and Phrases: geometric deep learning, geometry processing, discrete differential geometry, diffusion

Fig. 1. Surface learning methods must generalize to shapes represented differently from the training set to be useful in practice, yet many existing approaches depend strongly on mesh connectivity. Here, our DiffusionNet trained for human segmentation with limited variability seen during training automatically generalizes to widely varying mesh samplings (*left*), scales gracefully to resolutions ranging from a simplified model to a large raw scan (*middle*), and can even be evaluated directly on point clouds (*right*).

1 INTRODUCTION

Diffusion Based Networks

pointwise MLP + **learned diffusion** + gradient features



geodesic convolutions

pooling hierarchies

difficult on surfaces

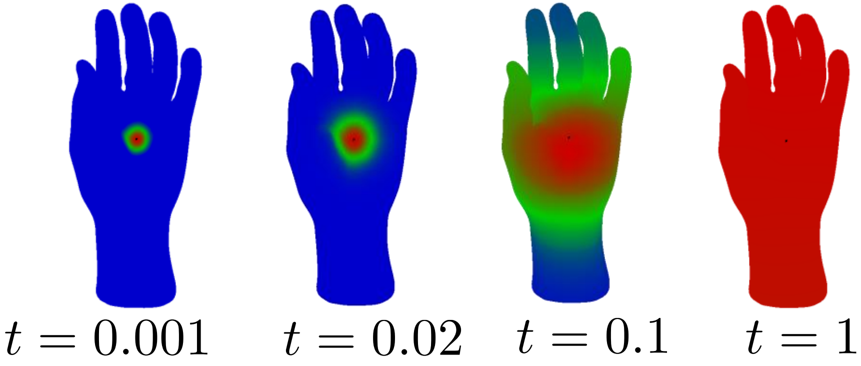
source of non-robustness

use diffusion instead!

"DiffusionNet: Discetization agnostic learning on surfaces" Sharp et al., 2021

Recall: Laplacian and Diffusion

$$\Delta_{\mathcal{X}} u(x,t) = -\frac{\partial u(x,t)}{\partial t}$$



$$k_t(x,y) = \sum_{l=0}^{\infty} e^{-t\lambda_l} \phi_l(x)\phi_l(y)$$

“DiffusionNet: Discetization agnostic learning on surfaces" Sharp et al., 2021

Learned Diffusion

~~convolution~~

~~pooling~~

✓ diffusion!

Key idea: the *diffusion time* is a learned parameter

↳ variable per-channel spatial support

↳ ranges from purely local to totally global

↳ automatically optimized during training

learned diffusion layer $h_t : \mathbb{R}^{\Omega \times k} \to \mathbb{R}^{\Omega \times k}$ parameterized by $t \in \mathbb{R}^k_{\geq 0}$

> **Lemma**: diffusion + pointwise MLPs can represent all (radially symmetric) convolutions.

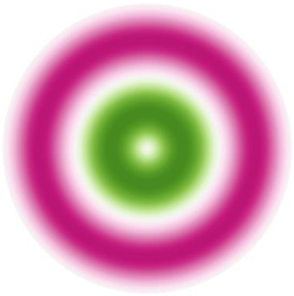
“DiffusionNet: Discetization agnostic learning on surfaces" Sharp et al., 2021

Spatial gradient features

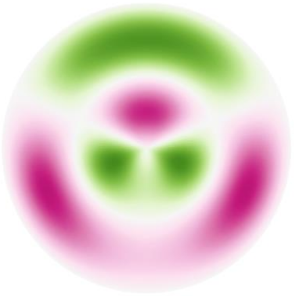
Challenge: we want to go beyond radially-symmetric filters

Solution: append extra features, dot products of spatial gradient

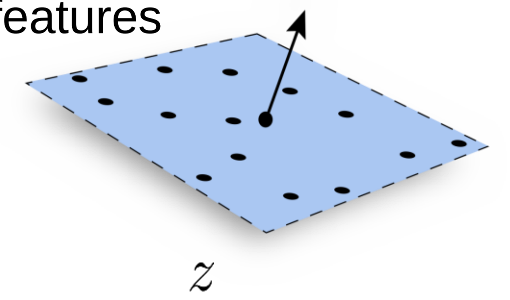
✗ radially symmetric filters only



beyond radially symmetric filters ✓



spatial gradient of scalar features

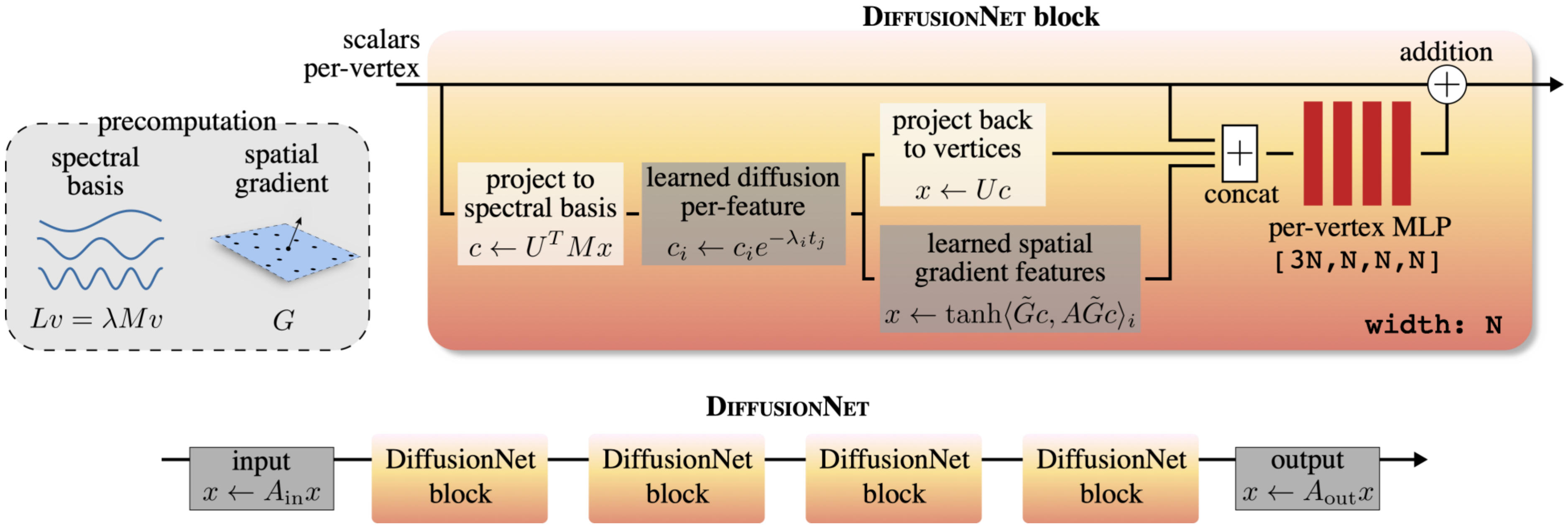


$filter(i) = g(z, A)$, where A is a learned rotation

Important detail: invariant to choice of tangent space

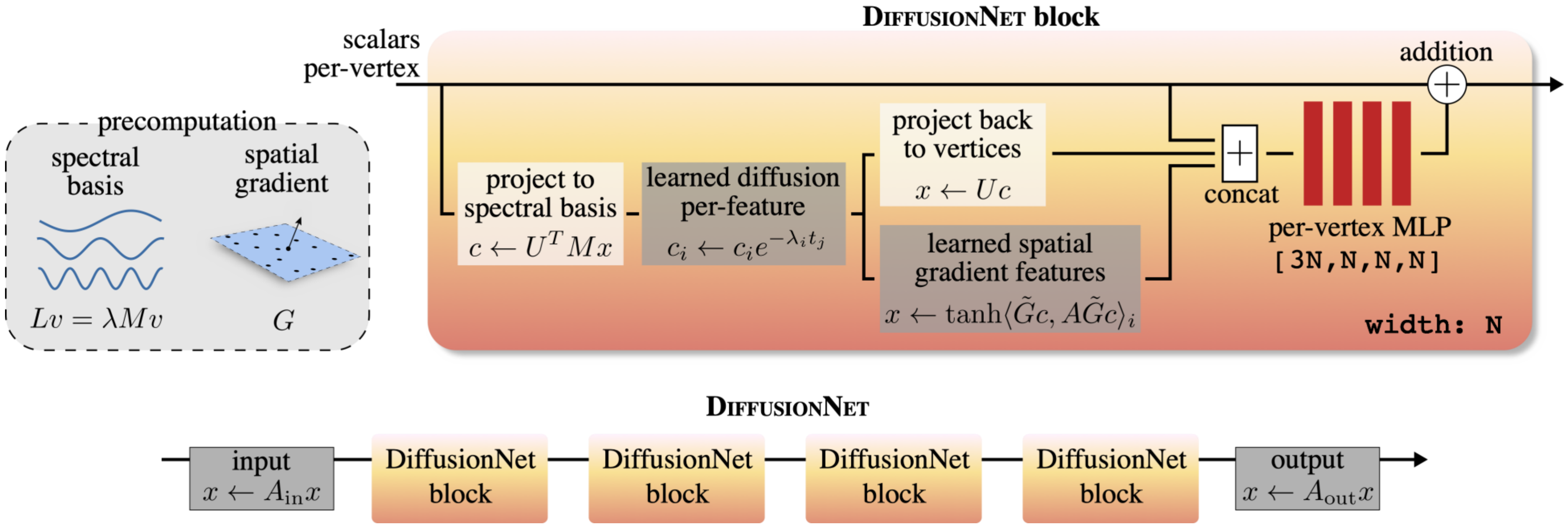
"DiffusionNet: Discetization agnostic learning on surfaces" Sharp et al., 2021

DiffusionNet Architecture



DiffusionNet: Discretization Agnostic Learning on Surfaces, N. Sharp, S. Attaiki, K. Crane, M.O., https://arxiv.org/abs/2012.00888

DiffusionNet Architecture

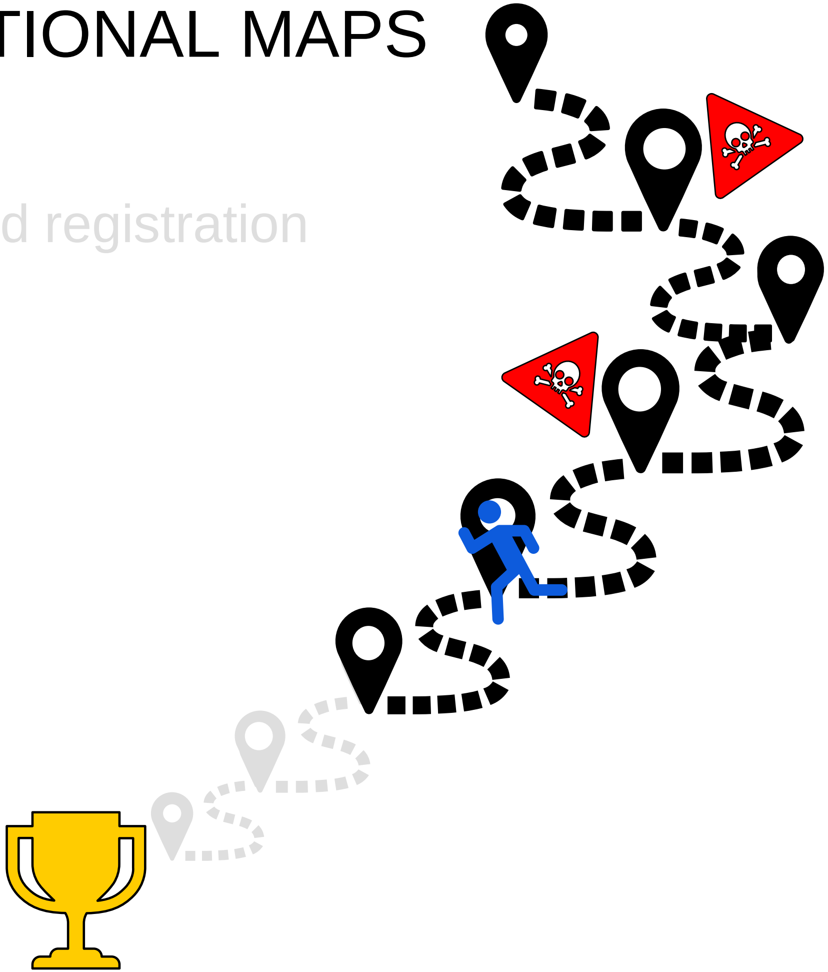


"DiffusionNet: Discetization agnostic learning on surfaces" Sharp et al., 2021

DiffusionNet DEMO

https://github.com/nmwsharp/diffusion-net

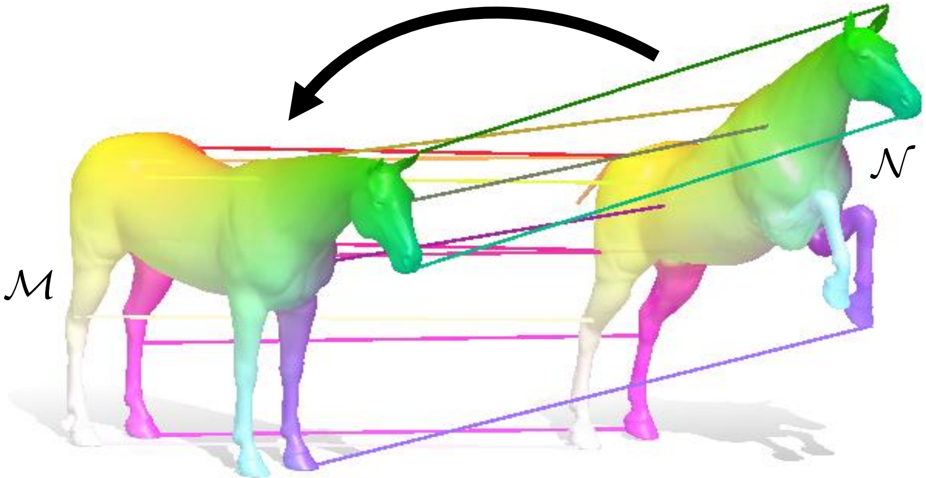
LEARNING-BASED FUNCTIONAL MAPS

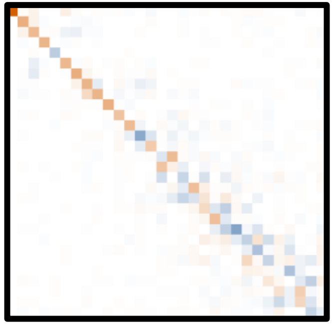


1. Introduction: 3D Non-Rigid shapes and registration
2. Spectral representation
3. Axiomatic approaches
4. Functional maps
5. Learning on geometric data
6. **Learning–based Functional maps**
7. Other learning-based approaches
8. Transformers

Functional Map Representation

$$\Pi_{\mathcal{M}\mathcal{N}} : \mathcal{N} \rightarrow \mathcal{M}$$





$$\boldsymbol{C} = \boldsymbol{\Phi}_{\mathcal{N}}^{\dagger} \boldsymbol{\Pi}_{\mathcal{N}\mathcal{M}} \boldsymbol{\Phi}_{\mathcal{M}}$$

Main Advantages:

- Functional map matrix **C** is *much* smaller than $\Pi_{\mathcal{M}\mathcal{N}}$
- Natural constraints on the map are easy to express.

Functional maps: a flexible representation of maps between shapes, *Ovsjanikov et al.* SIGGRAPH 2012

Fmaps pipeline

Given a pair of shapes $\mathcal{M}, \mathcal{N}$:

1. Compute the first *k* (~30-100) eigenfunctions of the LBO.
   Store them in matrices: $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$

2. Compute probe functions (e.g., landmarks or descriptors) on $\mathcal{M}, \mathcal{N}$.
   Express them in $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$, as columns of $\boldsymbol{A}$ and $\boldsymbol{B}$

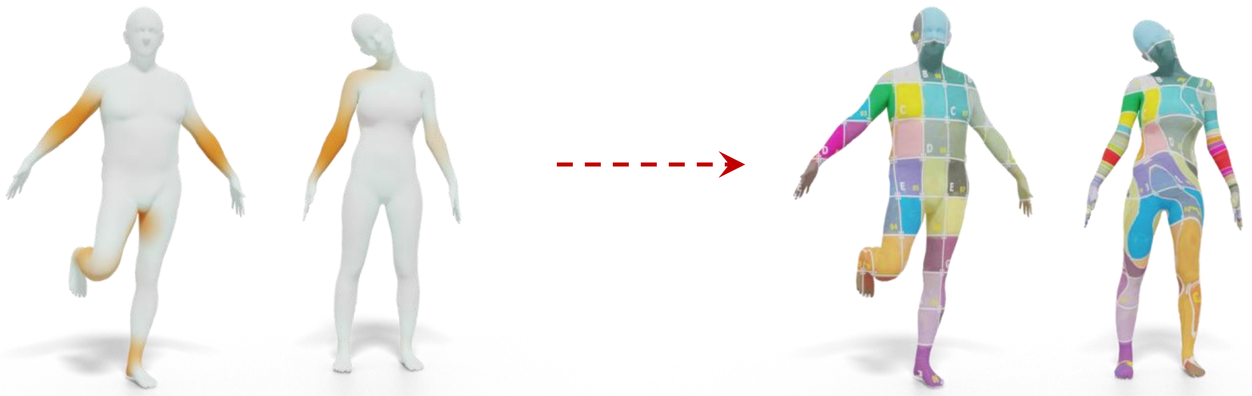
3. Solve $\underset{\boldsymbol{C}}{argmin} \|\boldsymbol{C}\boldsymbol{a} - \boldsymbol{b}\|_F^2 + \mathcal{R}(\boldsymbol{C})$

4. Convert the functional map to a point-to-point map *T*.

Computing and Processing Correspondences with Functional Maps., Ovsjanikov et al., SIGGRAPH courses 2017

Main Question

What happens if the input descriptors are bad?

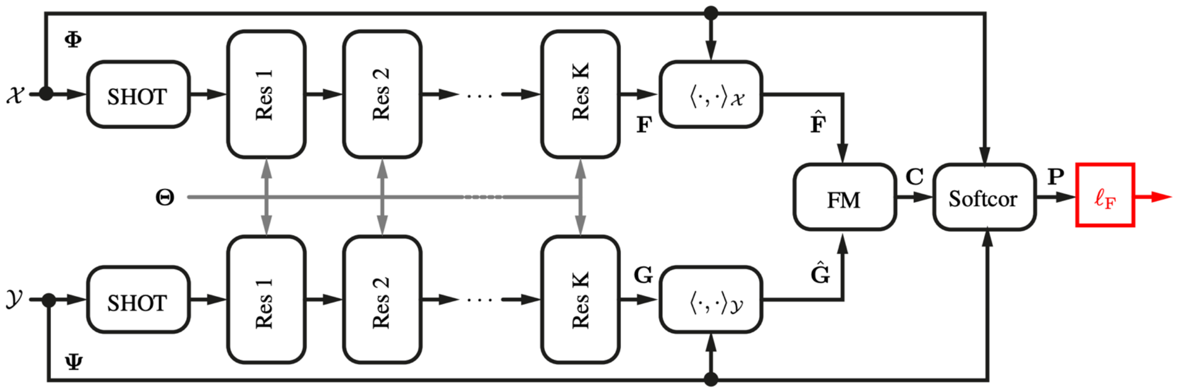


Input descriptors not in alignment

Results in poor texture transfer

FMNet

Learning approach to computing mappings.



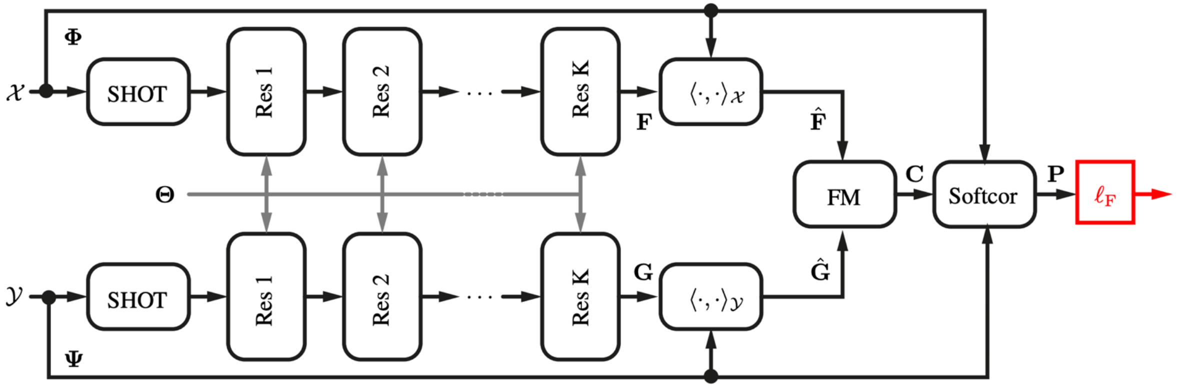
FM layer: $\mathbf{C} = \arg\min_{\mathbf{C}} \|\mathbf{C}\hat{\mathbf{F}} - \hat{\mathbf{G}}\|$

Solution given by a linear system of equations.
Can back-propagate via derivatives of linear systems

Deep functional maps: Structured prediction for dense shape correspondence. Litany et al., ICCV 2017

FMNet

Learning approach to computing mappings.



FM layer: $\mathbf{C} = \arg\min_{\mathbf{C}} \|\mathbf{C}\hat{\mathbf{F}} - \hat{\mathbf{G}}\|$

Training loss: $\ell_F = \sum_{(x,y)\in(\mathcal{X},\mathcal{Y})} P(x,y) d_{\mathcal{Y}}(y, \pi^*(x))$ $\quad \mathbf{P} = |\mathbf{\Psi}\mathbf{C}\mathbf{\Phi}^\top \mathbf{A}|^\wedge$

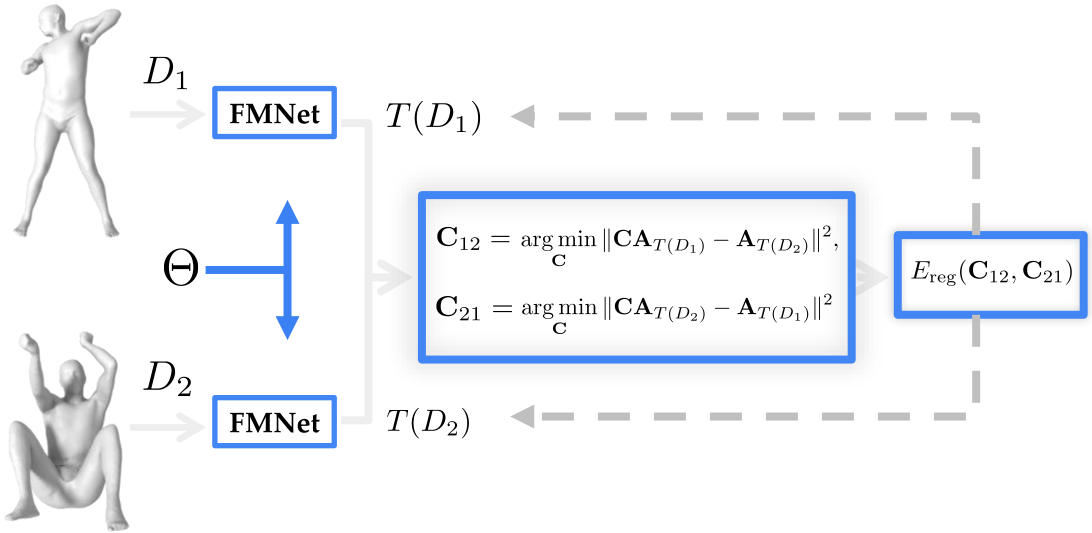
$P(x,y)$:soft map corresponding to the fmap C

Key advantage: evaluates the *entire map*. State-of-the art in 2017

Deep functional maps: Structured prediction for dense shape correspondence. Litany et al., ICCV 2017

SURFMNet

Main idea: make the loss fully unsupervised.



Rouffosse et al., "Unsupervised Deep Learning for Structured Shape Matching," ICCV 2019
Halimi et al., "Unsupervised Learning of Dense Shape Correspondence," CVPR 2019

SURFMNet

Replace supervised loss with unsupervised one

$$\mathrm{loss}_{unsupervised} = \sum_{i \in \mathrm{penalties}} w_i E_i(C_{12}, C_{21})$$

$$\begin{cases} E_1(C_{12}, C_{21}) = \|C_{12}C_{21} - Id\|^2 \\ E_1(C_{12}, C_{21}) = \|C_{21}C_{12} - Id\|^2 \end{cases}$$ Bijectivity

$$E_2(C) = \|C^T C - Id\|^2$$ Area-preservation

$$E_3(C) = \|\Lambda_2 C - C\Lambda_1\|^2$$ Near-isometry

$$E_4(C) = \sum_i \|CX_{f_i} - Y_{g_i}C\|^2$$ Functional map close to pointwise one

All penalties are in the reduced basis. **50x faster** than FMNet

Rouffosse et al., “Unsupervised Deep Learning for Structured Shape Matching,” ICCV 2019

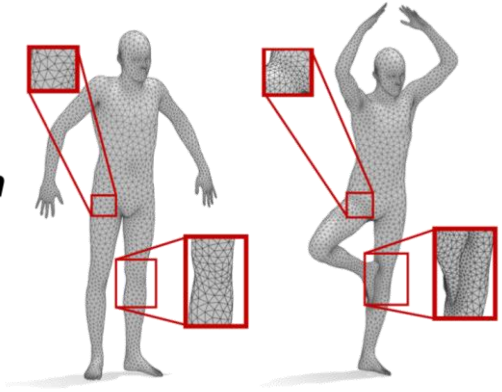
Datasets

FAUST :

- Subset: train on 80 and test on 20
- Whole set : train on 100 shapes, ***without ground truth***

SCAPE :

- Subset: train on 50 and test on 10
- Whole set : train on 60 shapes, ***without ground truth***

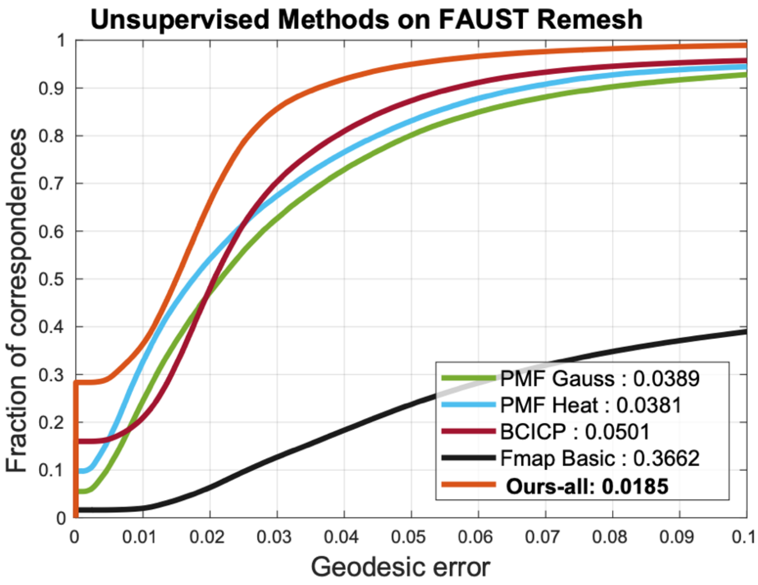


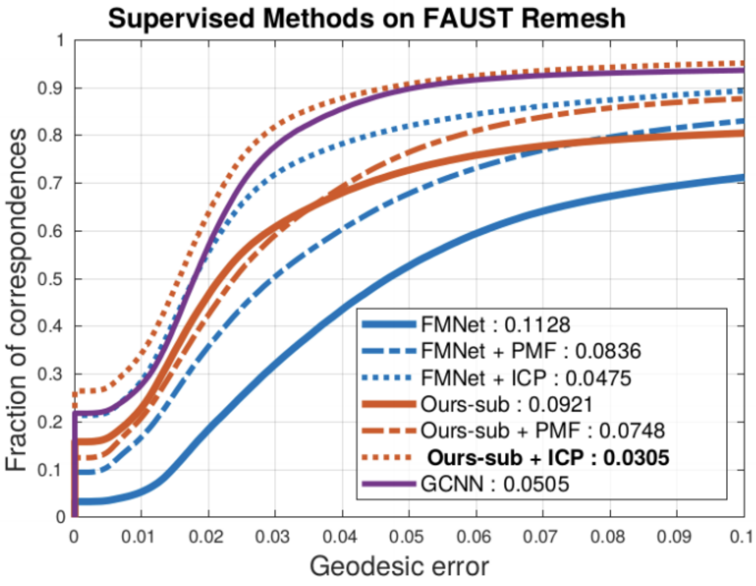
Remeshed FAUST - 5000 vertices*

*datasets released as part of: *Continuous and Orientation-preserving Correspondences via Functional Maps*, J. Ren, A. Poulenard, P. Wonka, M. O, SIGGRAPH Asia 2018

SURFMNet results

Comparison to unsupervised methods



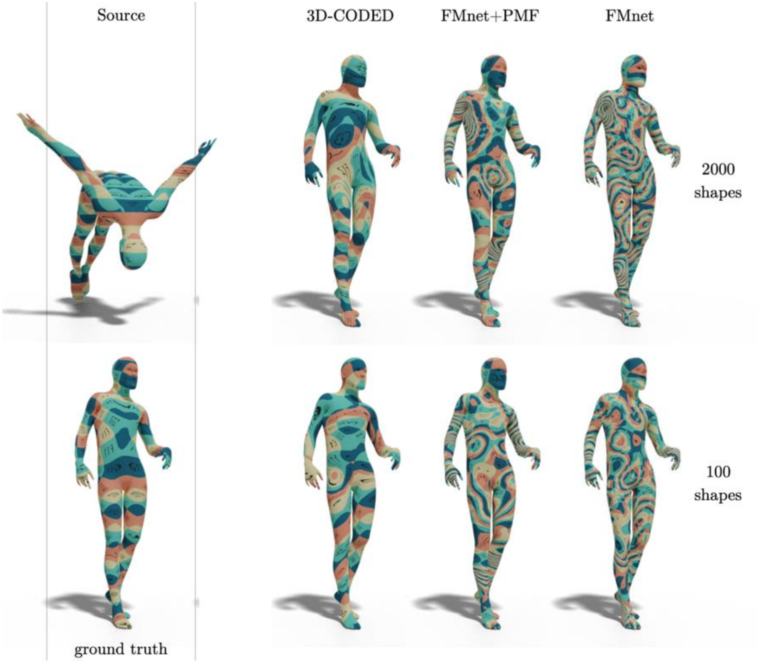


Remeshing makes the problem a lot harder

Rouffosse et al., “Unsupervised Deep Learning for Structured Shape Matching,” ICCV 2019

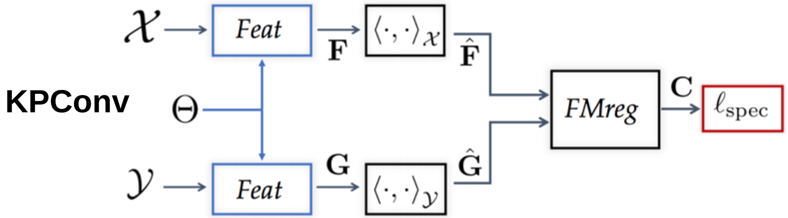
Questions for improvement

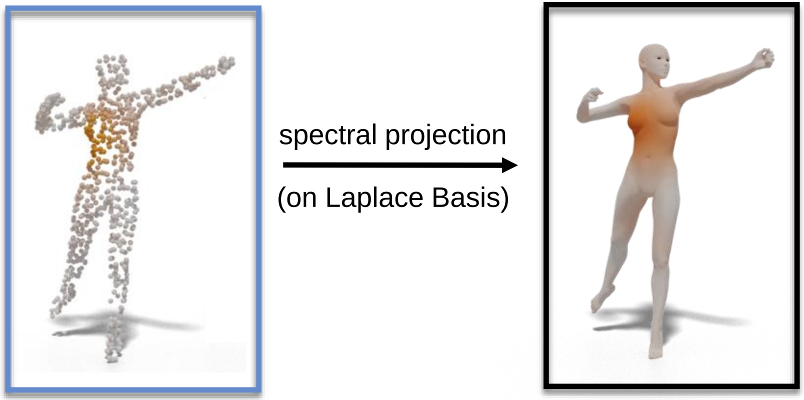
1. Use *raw geometry* (XYZ) instead of SHOT features as input?
2. How well do the methods generalize across different datasets?



Geometric Deep Functional Maps

Extract descriptor functions from the raw geometry!

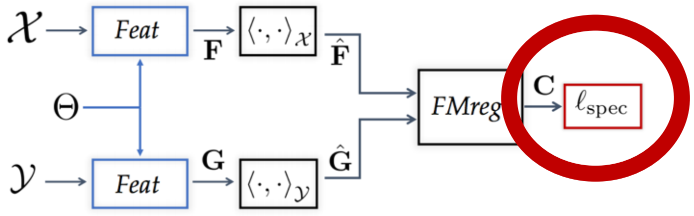




Deep Geometric Functional Maps: Robust Feature Learning for Shape Correspondence, Donati et al., CVPR 2020

Geometric Deep Functional Maps

Extract descriptor functions from the raw geometry!



Training loss in the spectral domain:

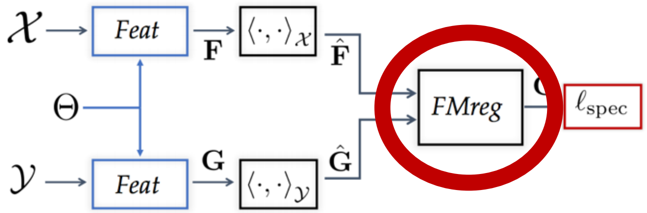
$$\ell_{spec}(C) = \|C - C_{gt}\|_F^2, \qquad C_{gt} = \Phi_2^{+}\Pi_{21}^{gt}\Phi_1$$

- Penalizes the map as a whole
- Does not require a template
- Does not require geodesic distance matrices

Deep Geometric Functional Maps: Robust Feature Learning for Shape Correspondence, Donati et al., CVPR 2020

Geometric Deep Functional Maps

Extract descriptor functions from the raw geometry!



Additional constraint **inside the network**: commutativity with Laplacian

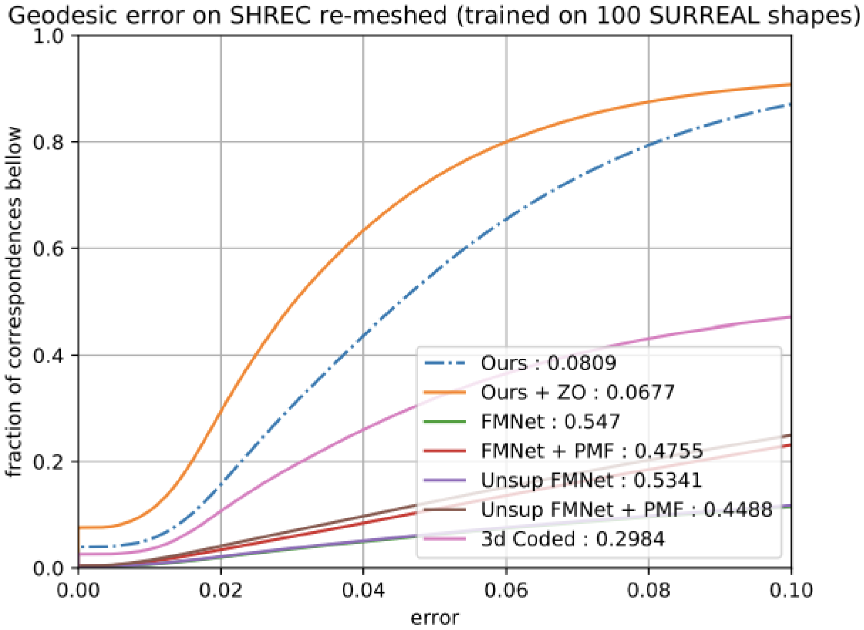
$$\min_{\mathbf{C}} \|\mathbf{CA} - \mathbf{B}\|^2 + \lambda \|\mathbf{C}\Delta_{\mathcal{M}} - \Delta_{\mathcal{N}}\mathbf{C}\|^2$$

Linear system for **every row** in **C** !

- Fully differentiable
- gives better maps

Deep Geometric Functional Maps: Robust Feature Learning for Shape Correspondence, Donati et al., CVPR 2020

Generalization Across Datasets



Deep Geometric Functional Maps: Robust Feature Learning for Shape Correspondence, Donati et al., CVPR 2020

Issues with Deep GeomFmaps

Problem:

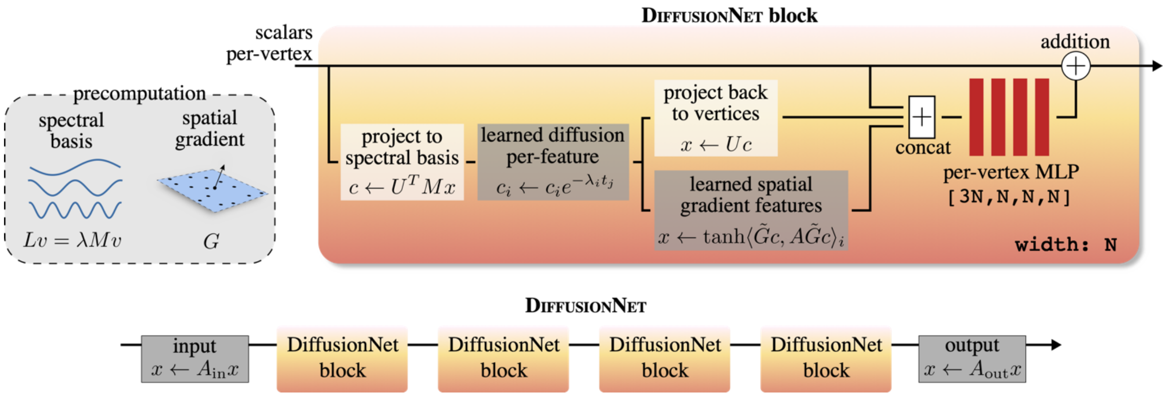
Still use *extrinsic* feature extractor (KPConv)

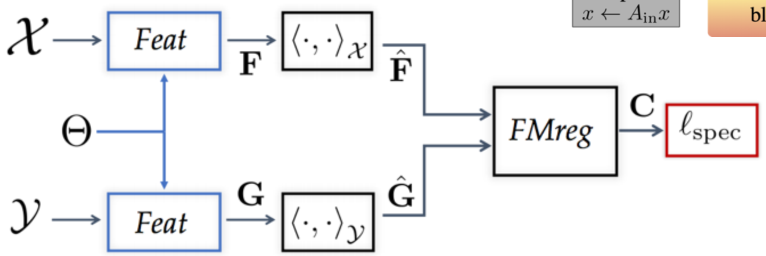
Main Questions:

How to enable **robust** and **efficient** intrinsic learning on surfaces (*choosing the architecture*)?

DiffusionNet for Geometric Deep Functional Maps

Immediate improvements with more robust feature learning methods:



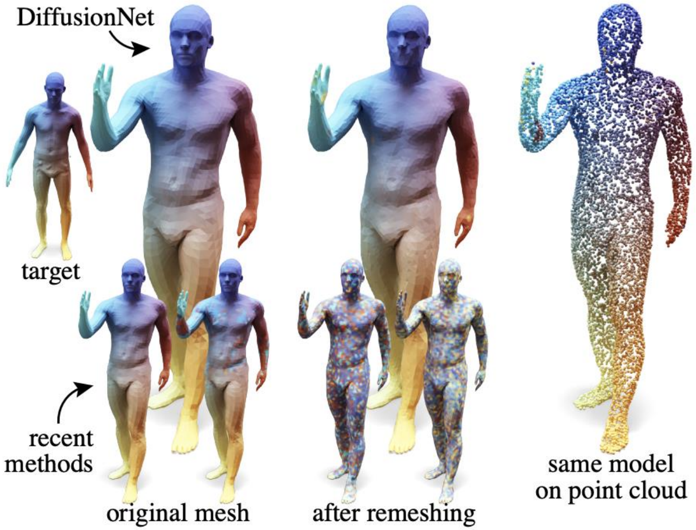


"DiffusionNet: Discetization agnostic learning on surfaces" Sharp et al., 2021

DiffusionNet for Geometric Deep Functional Maps

Method / Dataset	FAUST	SCAPE	F on S	S on F
KPConv [18]	3.1	4.4	11.0	6.0
KPConv - hks [18]	2.90	3.28	10.65	5.55
HSN [86]	3.29	3.53	25.41	16.66
ACSCNN [41]	2.75	3.22	8.44	6.08
DiffusionNet - hks	**2.53**	**2.97**	**5.61**	**3.00**

Table 4. Our approach yields state-of-the-art correspondence results when used as a feature extractor for deep functional maps [18]. X on Y means train on X and test on Y. Reported error values are mean geodesic error ×100 on shapes normalized to have unit area.



Key property in practice:

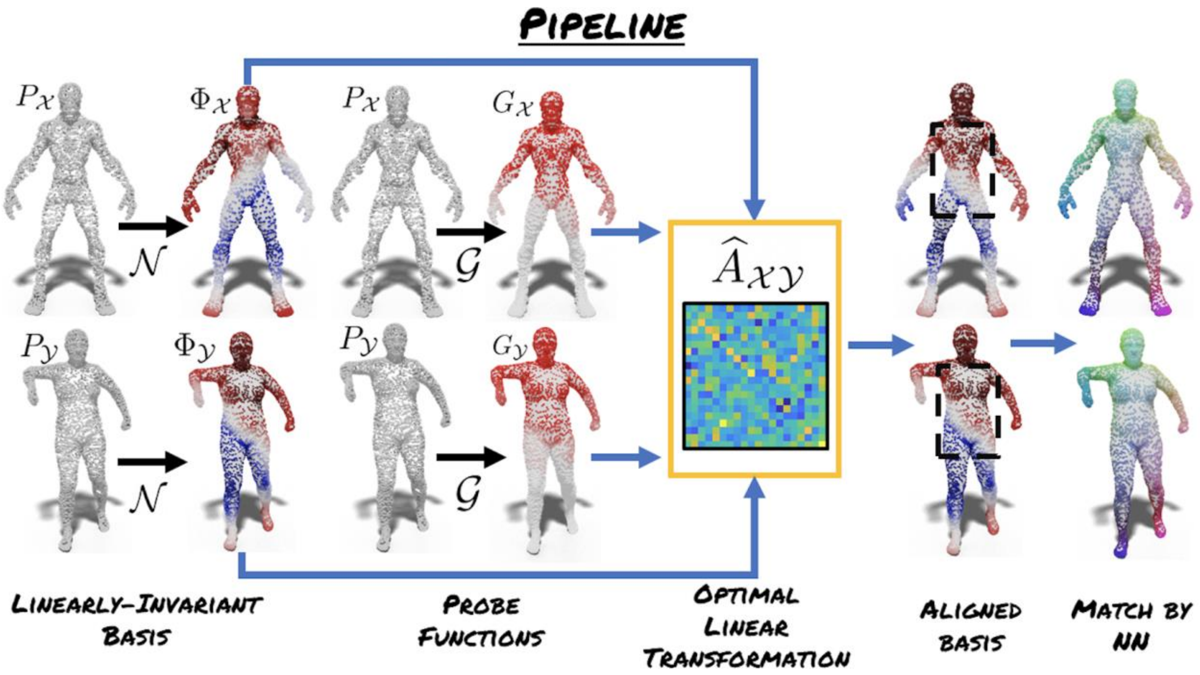
runs easily on full-size meshes/clouds! (no remeshing/downsampling)

“DiffusionNet: Discetization agnostic learning on surfaces" Sharp et al., 2021

DiffusionNet-based Functional Maps DEMO

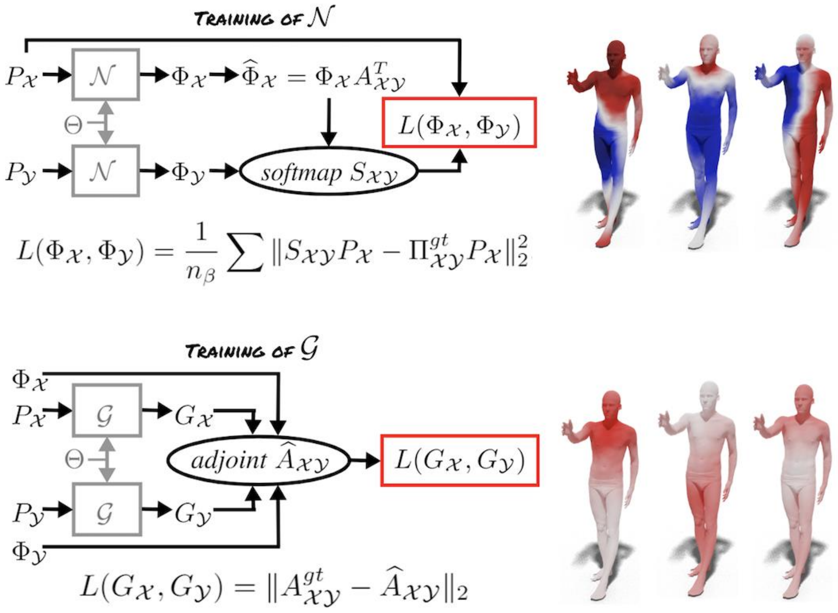
https://github.com/nmwsharp/diffusion-net/tree/master/experiments/functional_correspondence

Fully Differentiable Functional Maps



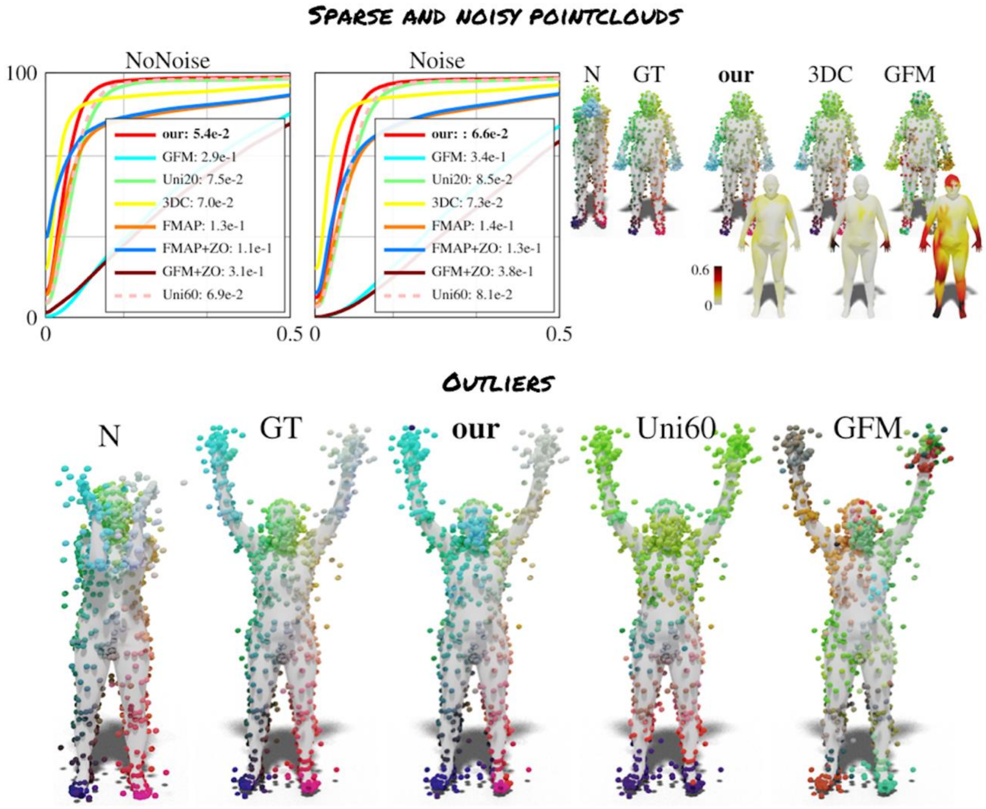
Correspondence Learning via Linearly-invariant Embedding, R. Marin et al., NeurIPS 2020

Fully Differentiable Functional Maps



Correspondence Learning via Linearly-invariant Embedding, R. Marin et al., NeurIPS 2020

Results



Correspondence Learning via Linearly-invariant Embedding, R. Marin et al., NeurIPS 2020

Results

Correspondence Learning via Linearly-invariant Embedding, R. Marin et al., NeurIPS 2020

LIE DEMO

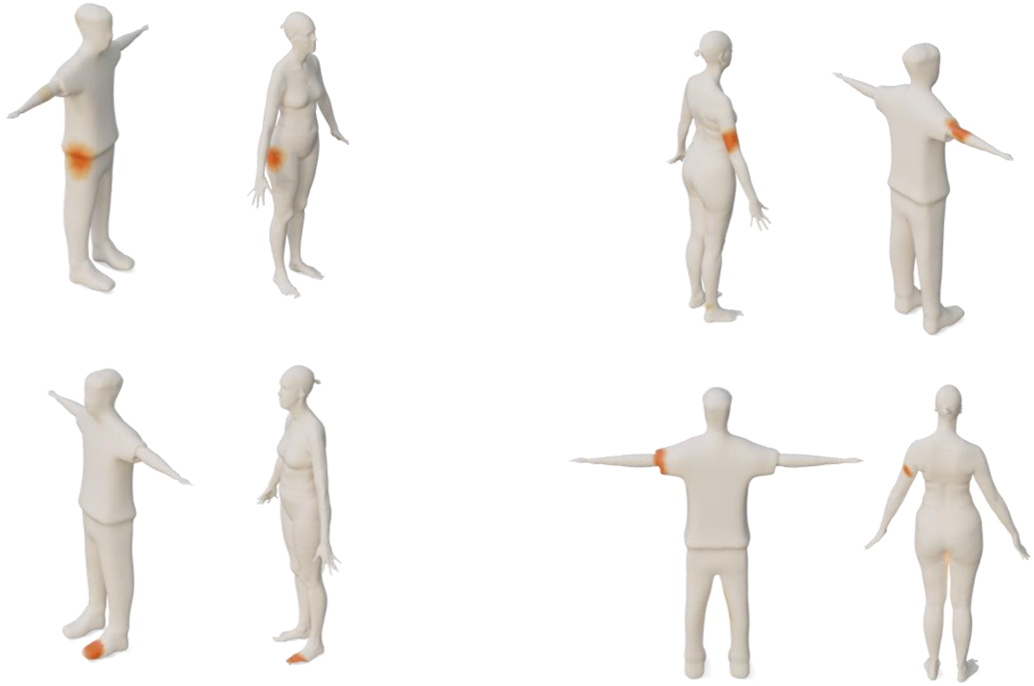
https://github.com/riccardomarin/Diff-FMaps

Current and Future Directions

1. Link between *shape matching* and *contrastive learning*.
2. Need for more datasets and tasks.
3. Better functional bases (beyond Laplacian).
4. Exploiting unsupervised feature pre-training in other tasks.
5. Working on other representations: implicit, point clouds, images, graphs, etc.

Visualize learned features for unsupervised Deep FMaps

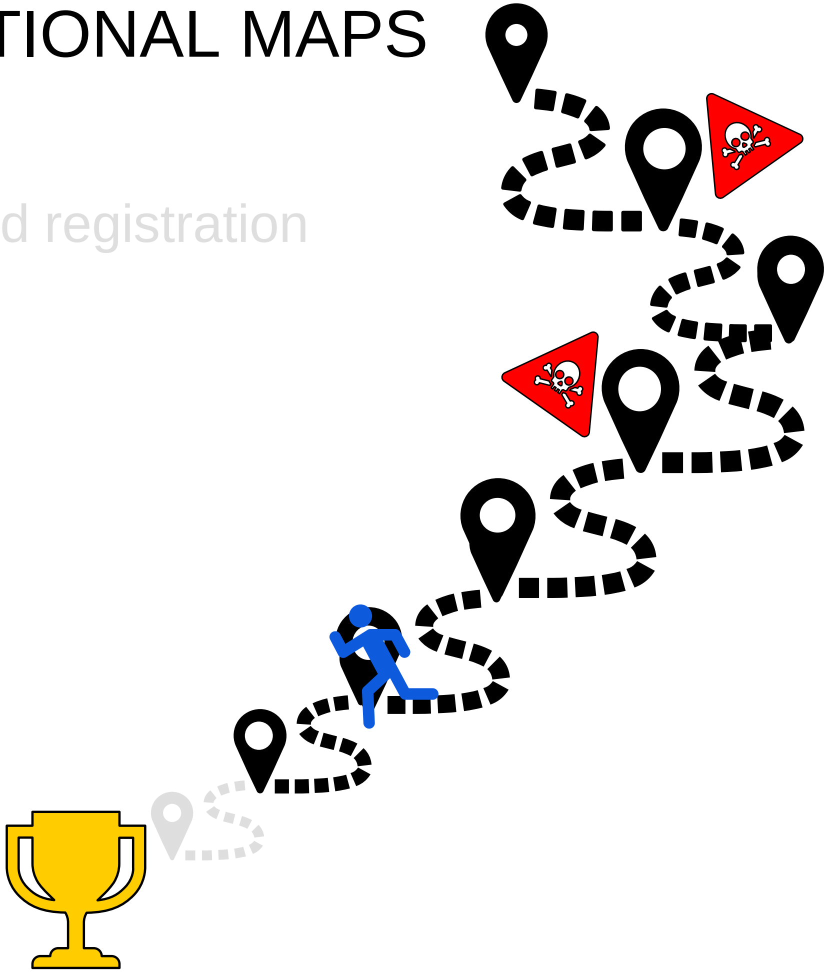
Learned features tend to be consistent and well-localized even on quite non-isometric shapes.



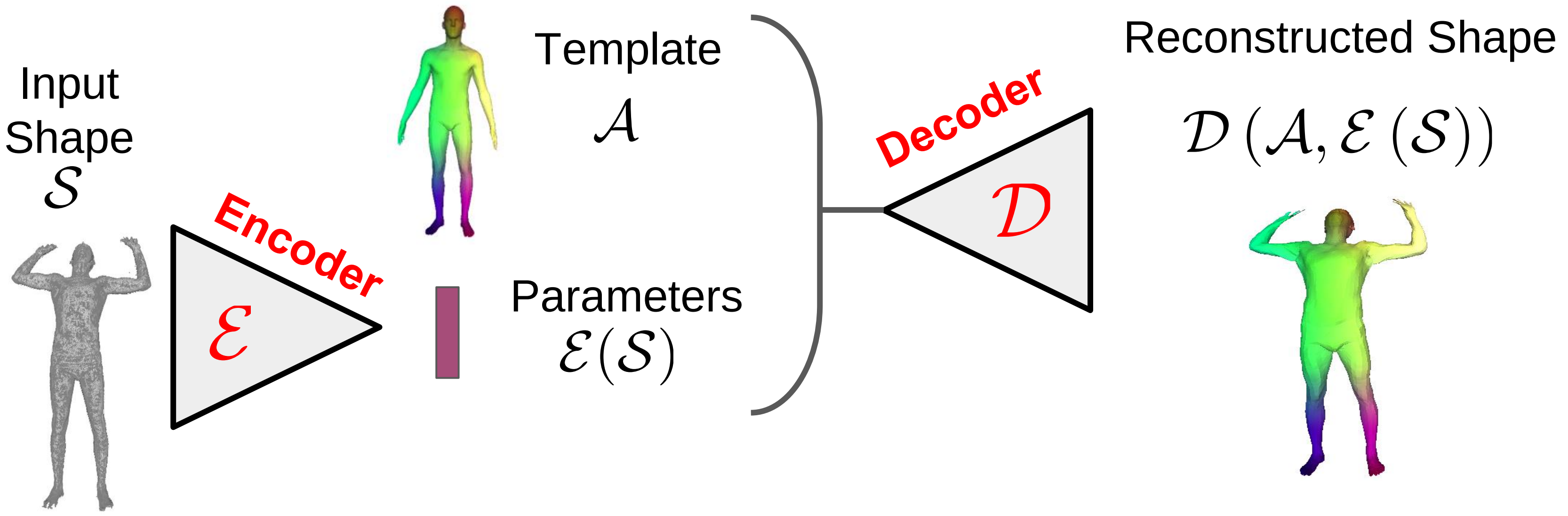
Understanding and Improving Features Learned in Deep Functional Maps, Attaiki, et al. CVPR 2023

LEARNING-BASED FUNCTIONAL MAPS

1. Introduction: 3D Non-Rigid shapes and registration
2. Spectral representation
3. Axiomatic approaches
4. Functional maps
5. Learning on geometric data
6. Learning–based Functional maps
7. **Other learning-based approaches**
8. Transformers

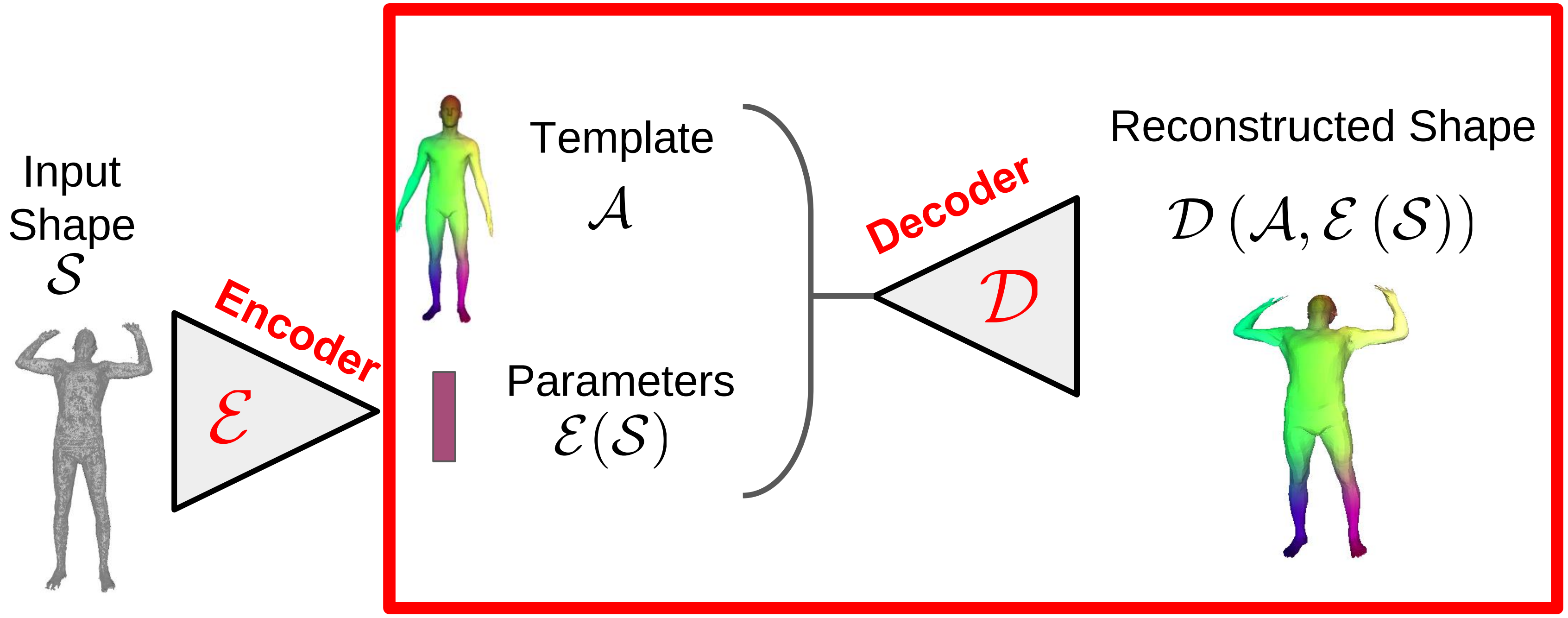


Auto-Encoder-based approach

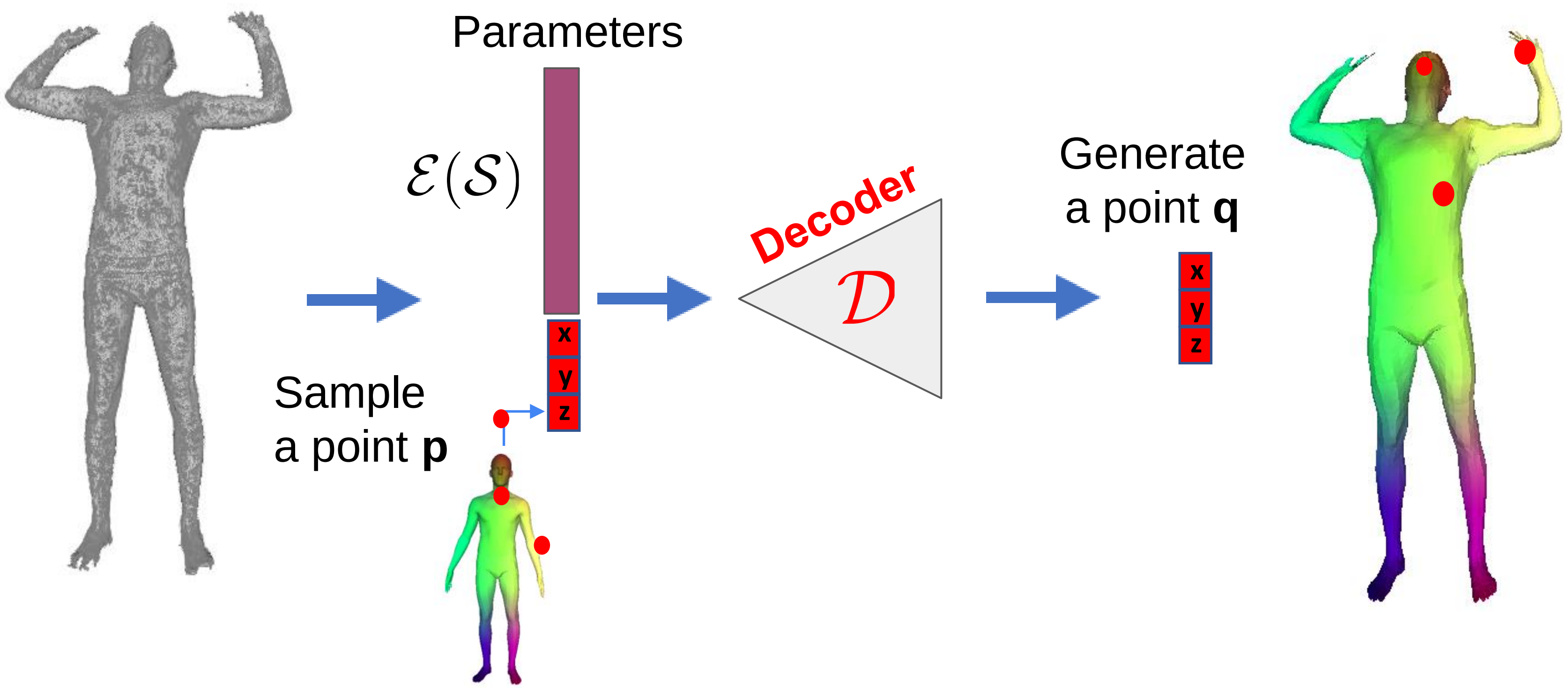


3D-CODED : 3D Correspondences by Deep Deformation, Groueix et al. ECCV 2018

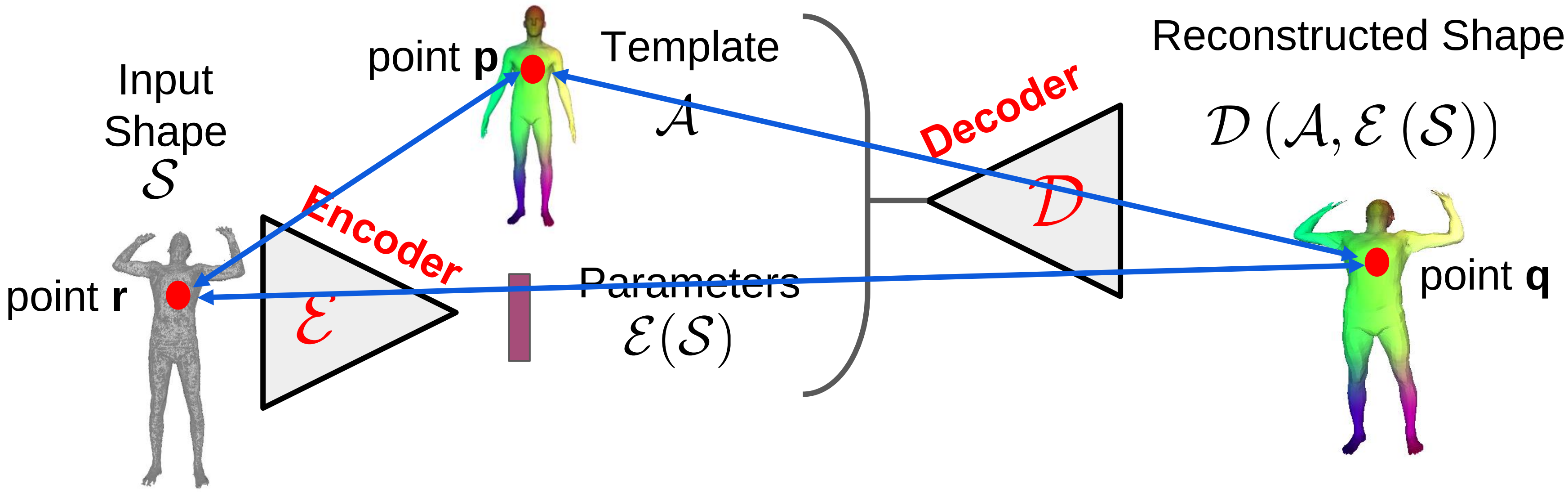
Decoder


Input
Shape
$\mathcal{S}$
Encoder
$\mathcal{E}$
Template
$\mathcal{A}$
Parameters
$\mathcal{E}(\mathcal{S})$
Decoder
$\mathcal{D}$
Reconstructed Shape
$\mathcal{D}(\mathcal{A}, \mathcal{E}(\mathcal{S}))$

The decoder which deforms the template

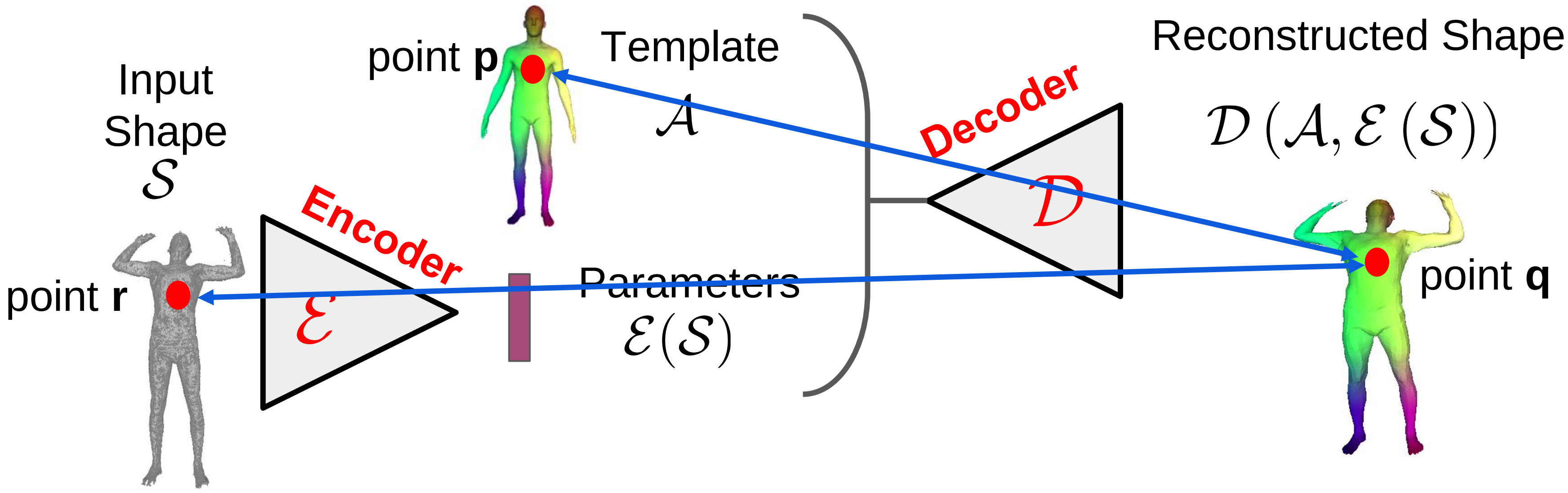


Supervised Loss



$$\mathcal{L}^{\text{sup}}(\mathcal{E}, \mathcal{D}) = \sum_{j=1}^{\#points} |\mathbf{r}_j - \mathbf{q}_j|^2$$

Unsupervised Loss



$$\mathcal{L}^{\mathrm{CD}}(\mathcal{E};\mathcal{D}) = \sum_{\mathbf{q}\in\mathcal{D}(\mathcal{A})} \min_{\mathbf{r}\in\mathcal{S}} |\mathbf{r}-\mathbf{q}|^2 + \sum_{\mathbf{r}\in\mathcal{S}} \min_{\mathbf{q}\in\mathcal{D}(\mathcal{A})} |\mathbf{r}-\mathbf{q}|^2 .$$

Unsupervised Loss + regularizations

$$\mathcal{L}^{\text{unsup}} = \mathcal{L}^{\text{CD}} + \lambda_{Lap}\mathcal{L}^{\text{Lap}} + \lambda_{edges}\mathcal{L}^{\text{edges}}$$

$\mathcal{L}^{\text{CD}}$: **Chamfer distance** (nearest neighbors based reconstruction loss) between deformed template and target shape.

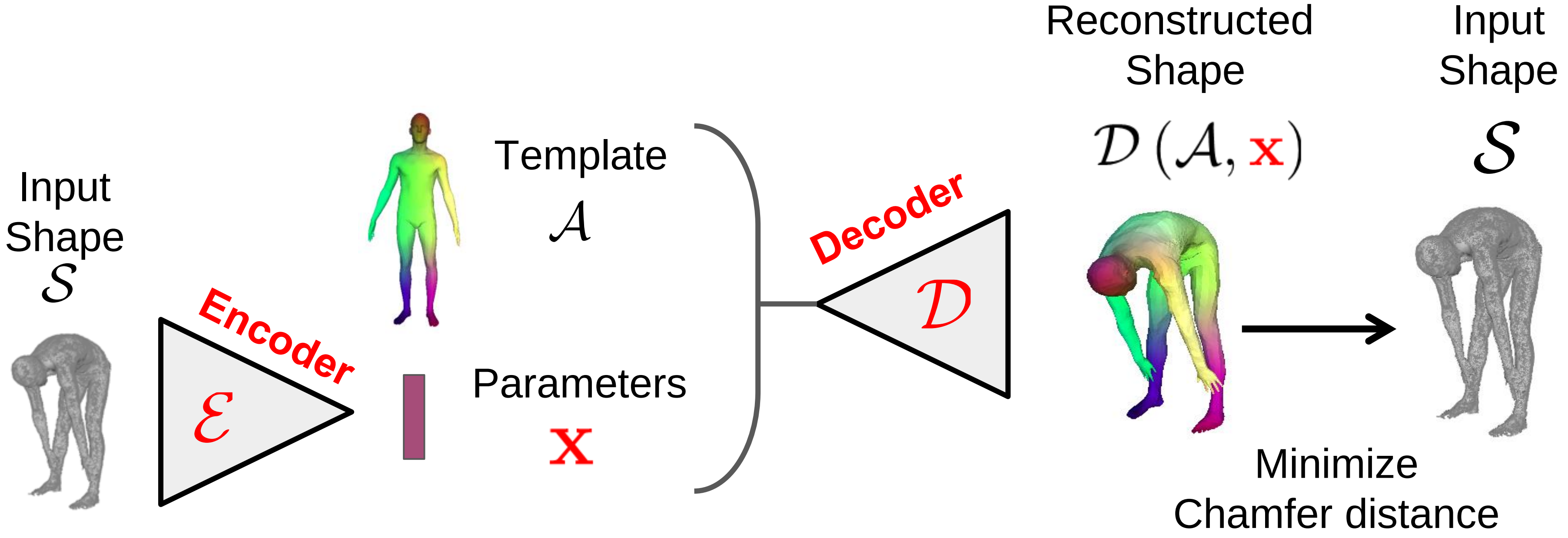
$$\mathcal{L}^{\text{CD}}(\mathcal{E};\mathcal{D}) = \sum_{\mathbf{q}\in\mathcal{D}(\mathcal{A})} \min_{\mathbf{r}\in\mathcal{S}} |\mathbf{r}-\mathbf{q}|^2 + \sum_{\mathbf{r}\in\mathcal{S}} \min_{\mathbf{q}\in\mathcal{D}(\mathcal{A})} |\mathbf{r}-\mathbf{q}|^2 .$$

$\mathcal{L}^{\text{edges}}$: **Edge ratio loss** (regularization). Preserve local neighbourhood of the template by encouraging each edge in the deformed template to keep the same length.

$$\mathcal{L}^{\text{edges}} = \frac{1}{\#E} \cdot \sum_{(i,j)\in E} \left| \frac{\|q_i - q_j\|}{\|p_i - p_j\|} - 1 \right|$$

$\mathcal{L}^{\text{Lap}}$: **Laplacian loss** (regularization). Preserve local neighbourhood of the template by encouraging the laplacian of the deformed template to remain constant.

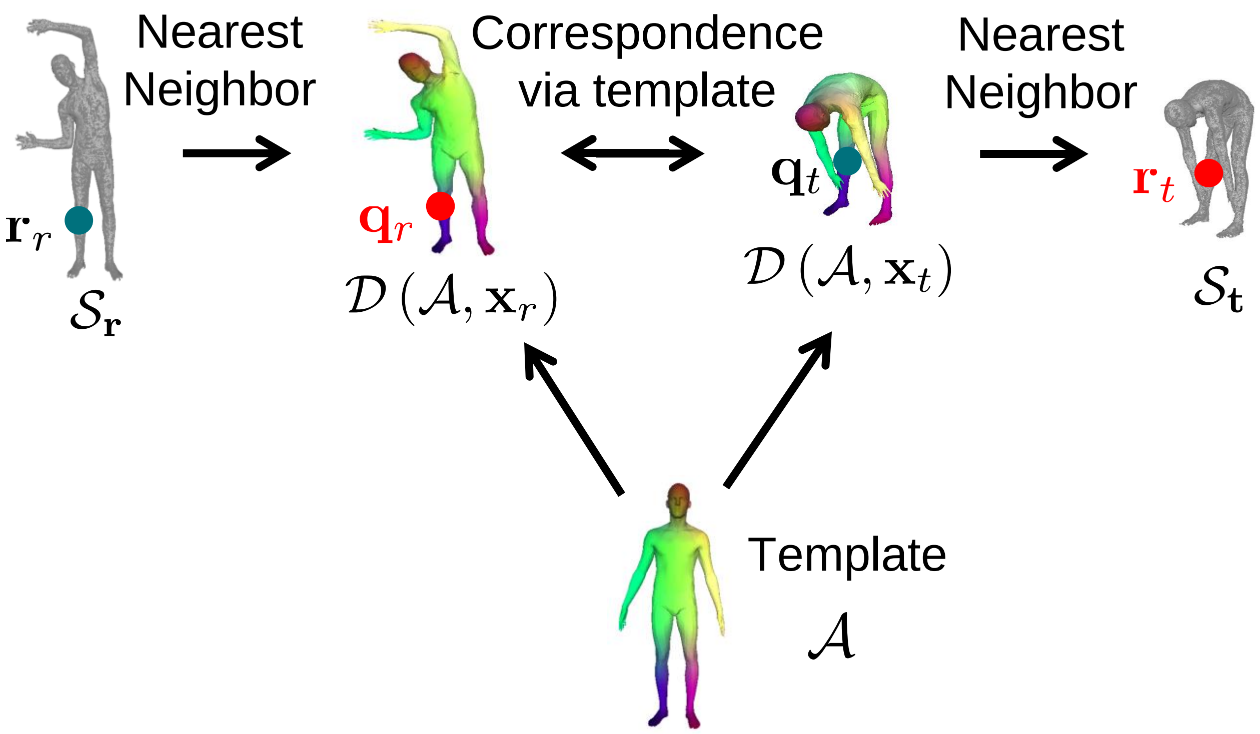
Refinement as parameters optimization



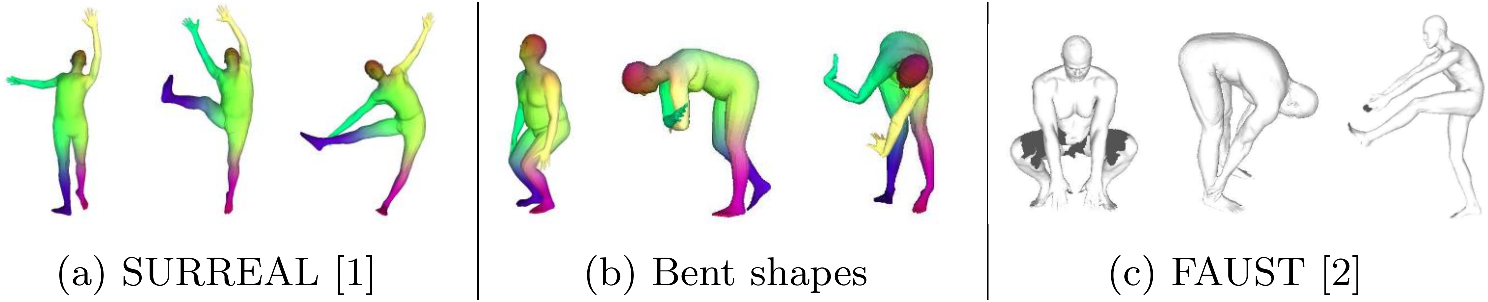
$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{\mathbf{p}\in\mathcal{A}} \min_{\mathbf{r}\in\mathcal{S}} |\mathcal{D}(\mathbf{p};\mathbf{x}) - \mathbf{r}|^2 + \sum_{\mathbf{q}\in\mathcal{S}} \min_{\mathbf{r}\in\mathcal{A}} |\mathcal{D}(\mathbf{p};\mathbf{x}) - \mathbf{r}|^2 .$$

3D-CODED : 3D Correspondences by Deep Deformation, Groueix et al. ECCV 2018

Evaluation: Finding 3D shape correspondences



Datasets: 230 000 synthetic human shapes

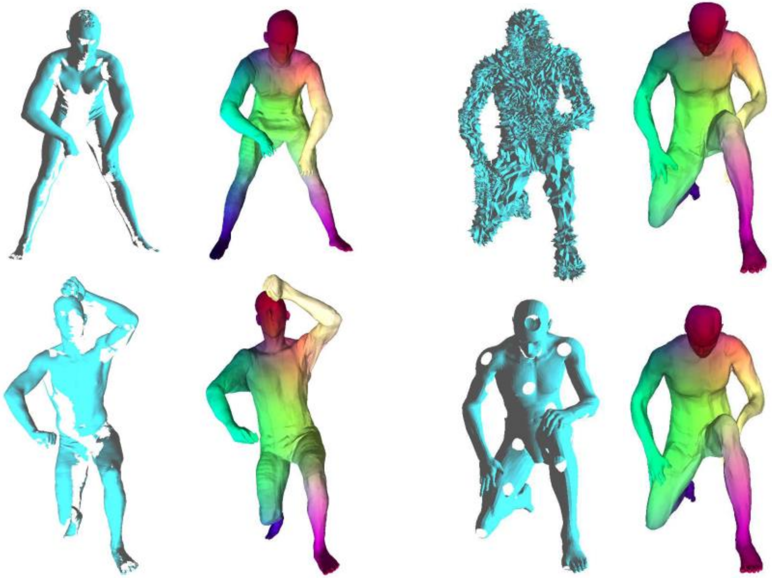


synthetic training data (a, b), real testing data (c).

Learning from synthetic humans, Varol et al. CVPR (2017)
[2] FAUST: Dataset and evaluation for 3D mesh registration, Bogo et al. CVPR (2014)

Robustness to perturbations

noise, holes, sampling, topology, scaling



SCAPE **TOSCA**

Scape: shape completion and animation of people, Anguelov et al. TOG (2005)
Numerical geometry of non-rigid shapes, Bronstein et al. Springer Science & Business Media (2008)

3D-CODED DEMO

https://github.com/ThibaultGROUEIX/3D-CODED

LEARNING-BASED FUNCTIONAL MAPS

1. Introduction: 3D Non-Rigid shapes and registration
2. Spectral representation
3. Axiomatic approaches
4. Functional maps
5. Learning on geometric data
6. Learning–based Functional maps
7. Other learning-based approaches
8. **Transformers**

Transformers

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention

Transformers Everywhere

NLP

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Published as a conference paper at ICLR 2021

Biology

BERTOLOGY MEETS BIOLOGY: INTERPRETING ATTENTION IN PROTEIN LANGUAGE MODELS

Jesse Vig[1] Ali Madani[1] Lav R. Varshney[1,2] Caiming Xiong[1]
Richard Socher[1] Nazneen Fatema Rajani[1]

Computer Graphics

Laplacian Mesh Transformer: Dual Attention and Topology Aware Network for 3D Mesh Classification and Segmentation

Xiao-Juan Li[1,2], Jie Yang (✉)[1,2], and Fang-Lue Zhang[3]
[1] Institute of Computing Technology, Chinese Academy of Sciences

Published as a conference paper at ICLR 2021

Computer Vision

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy[∗,†], Lucas Beyer[∗], Alexander Kolesnikov[∗], Dirk Weissenborn[∗], Xiaohua Zhai[∗], Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby[∗,†]
[∗]equal technical contribution, [†]equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com

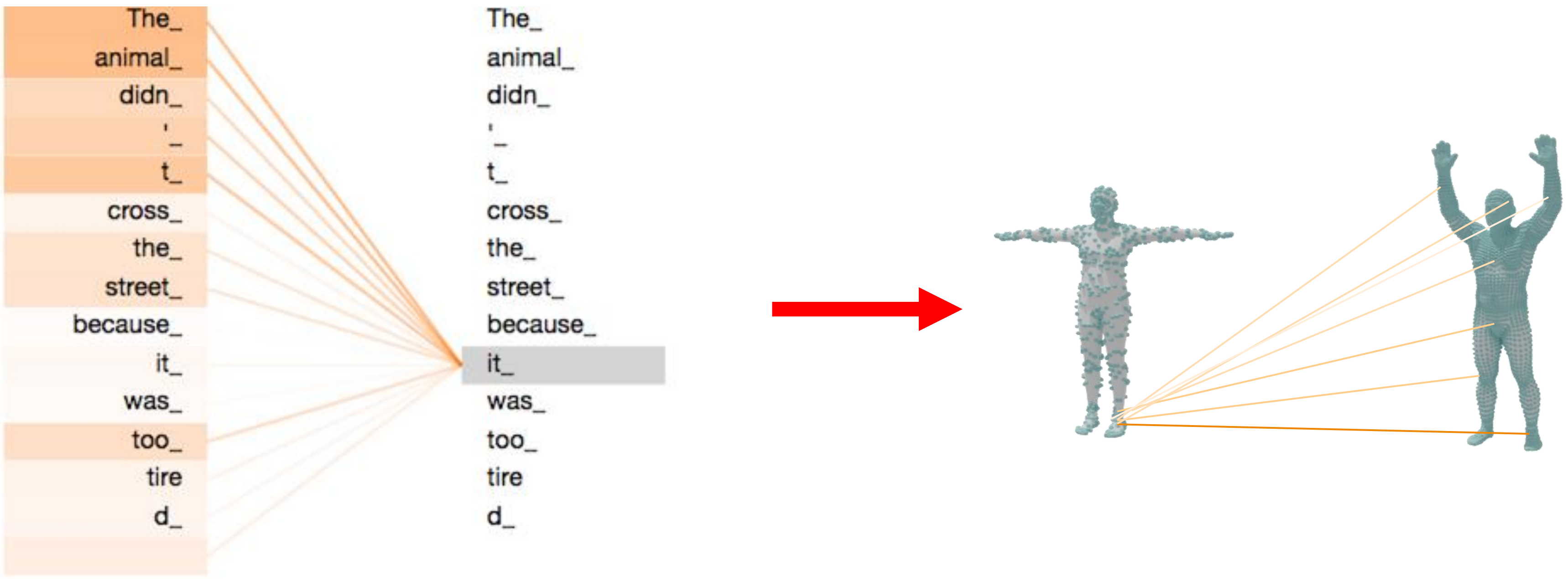
INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Audio

AST: Audio Spectrogram Transformer

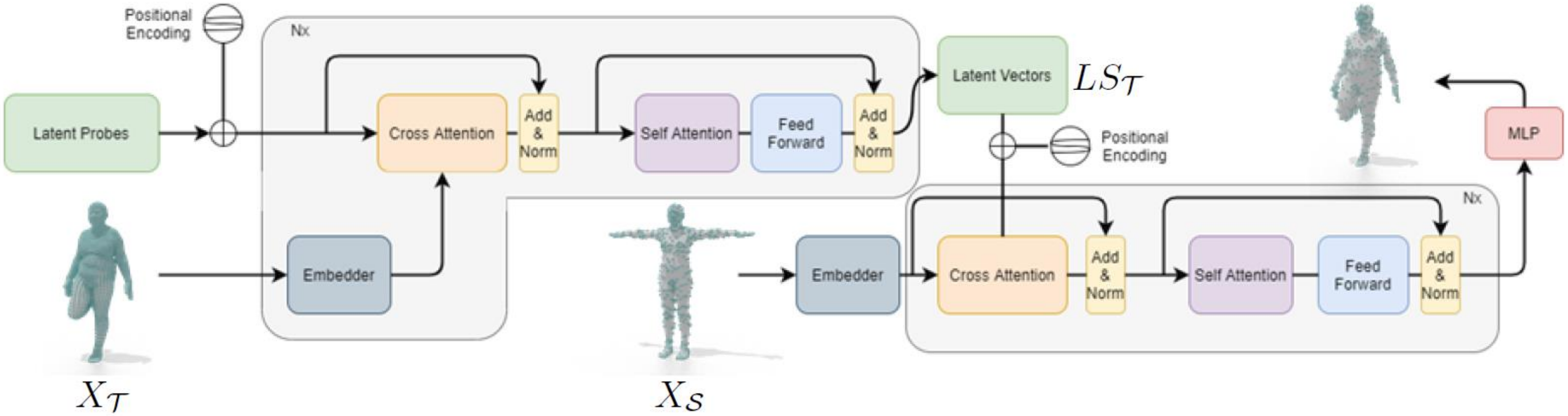
Yuan Gong, Yu-An Chung, James Glass
MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
{yuangong, andyyuan, glass}@mit.edu

Idea


The_
animal_
didn_
'_
t_
cross_
the_
street_
because_
it_
was_
too_
tire
d_
The_
animal_
didn_
'_
t_
cross_
the_
street_
because_
it_
was_
too_
tire
d_

Trappolini et al. NEURIPS 2021 (*SRTT*)

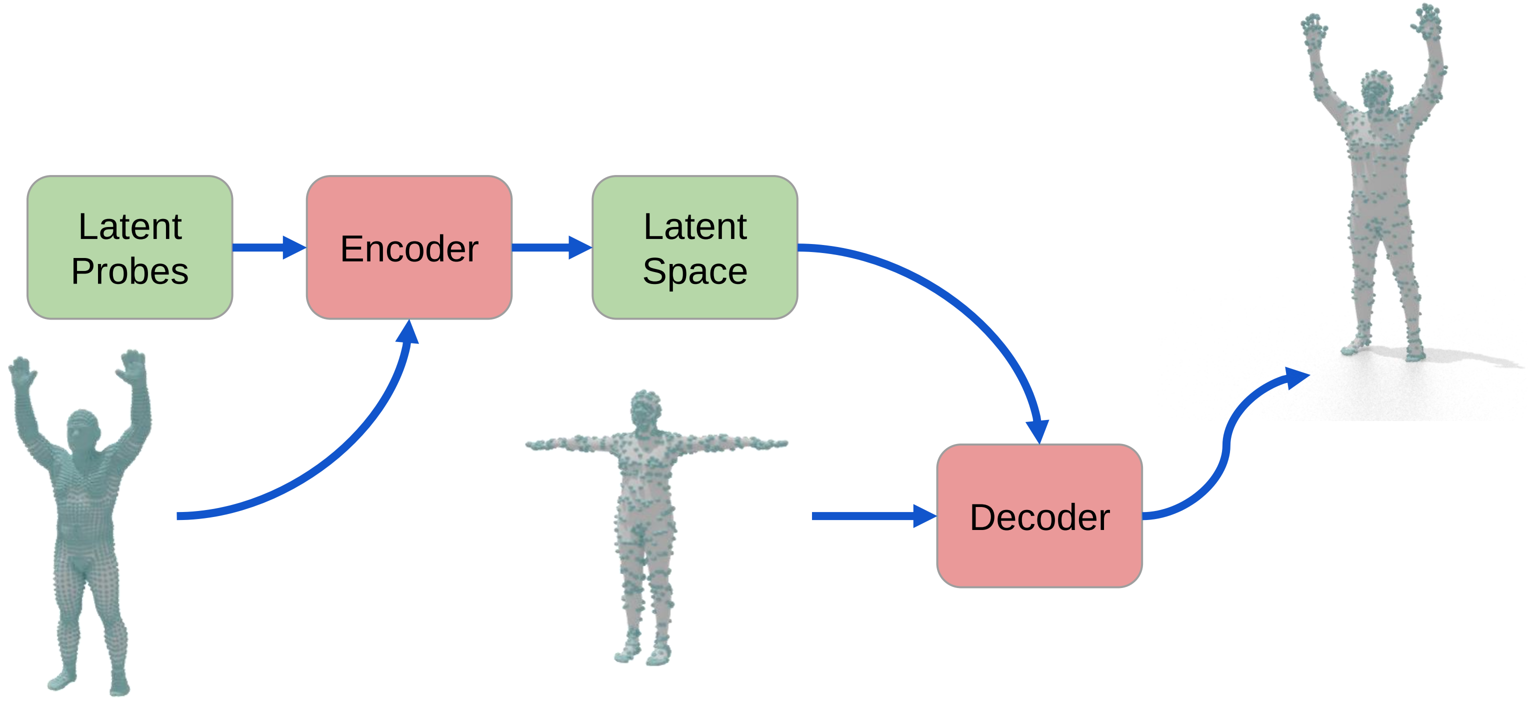
Is the first method for shape registration that exploits the transformers



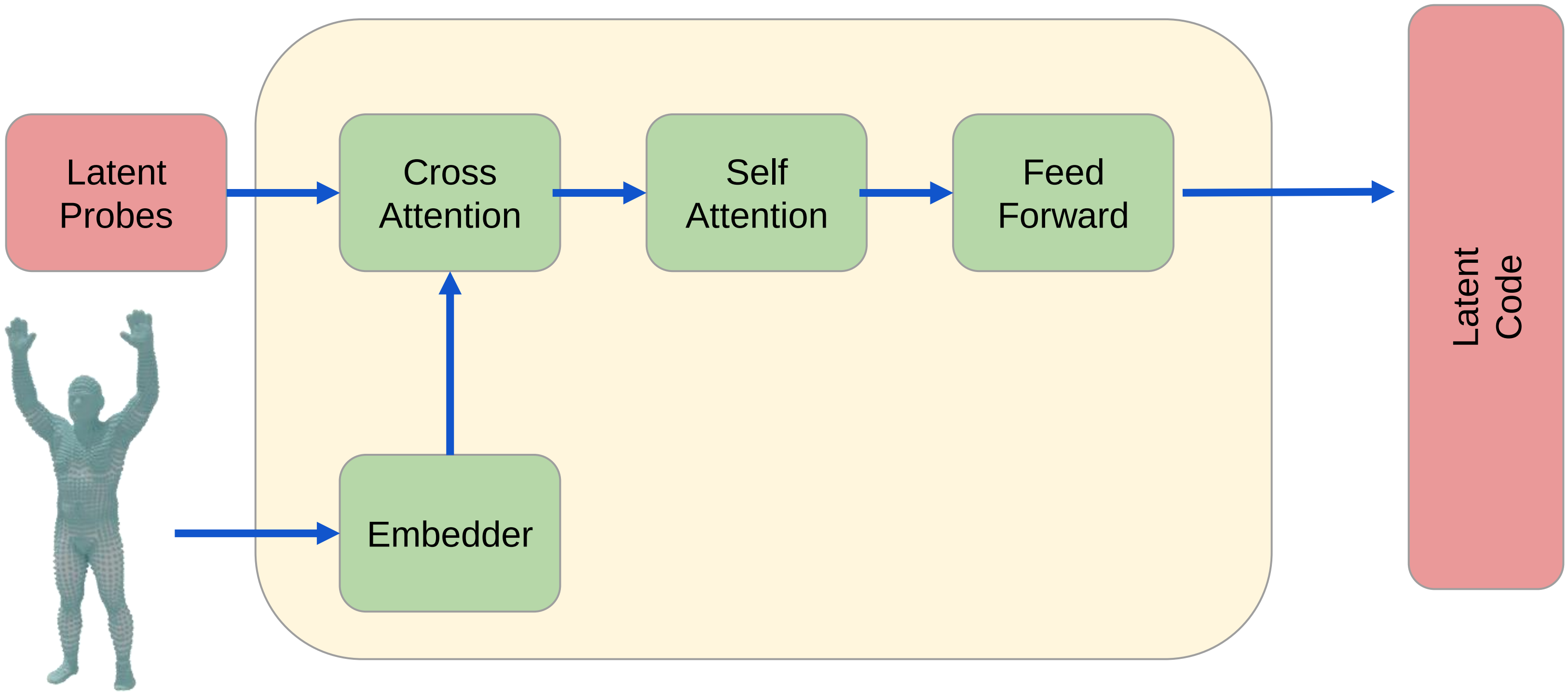
1. Adopts the Perceiver
2. Proposes the surface attention
3. Select one direction

Trappolini et al., "*Shape registration in the time of transformers*", NEURIPS 2021

Encoder Decoder Transformers

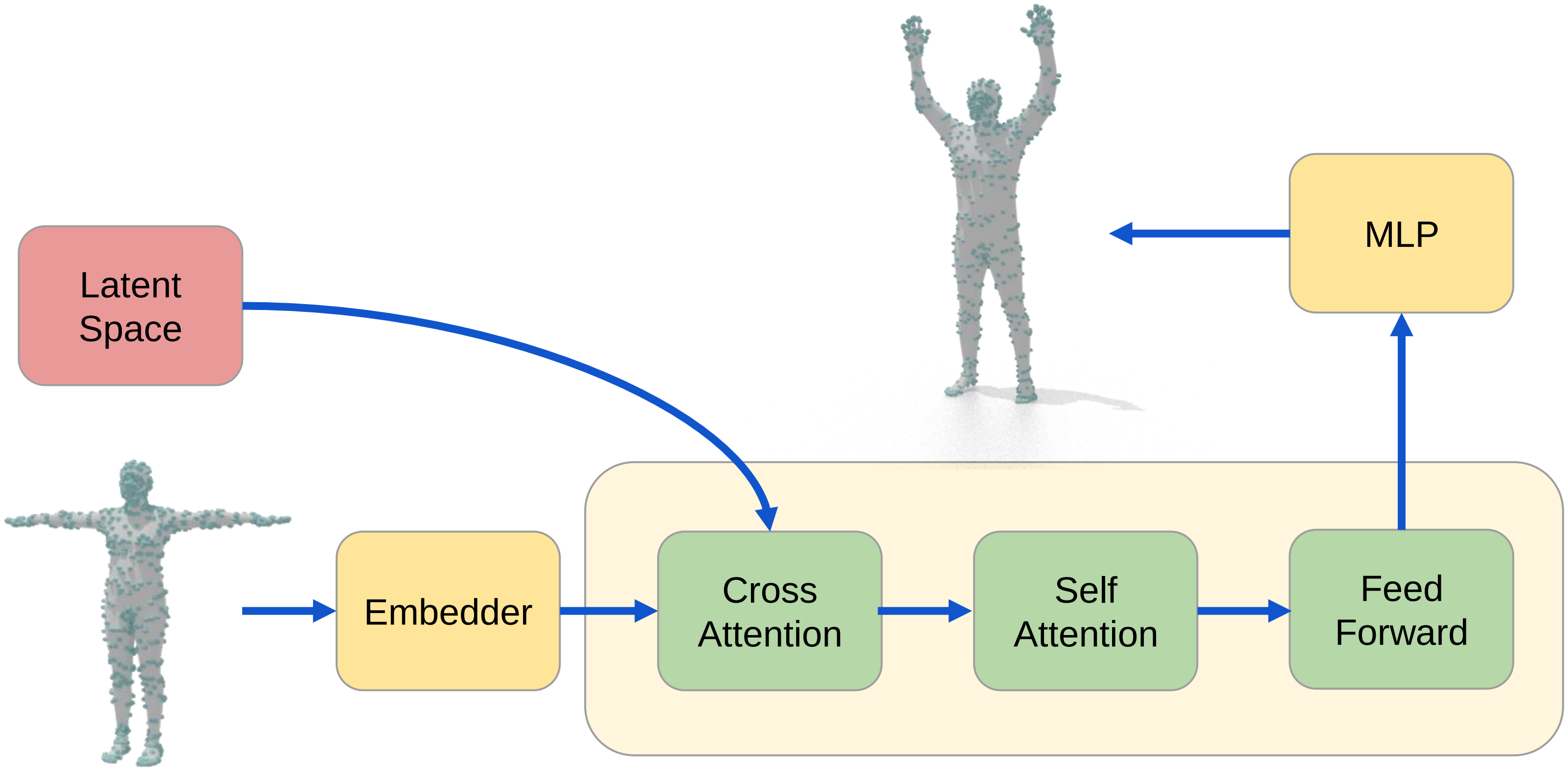


Encoder



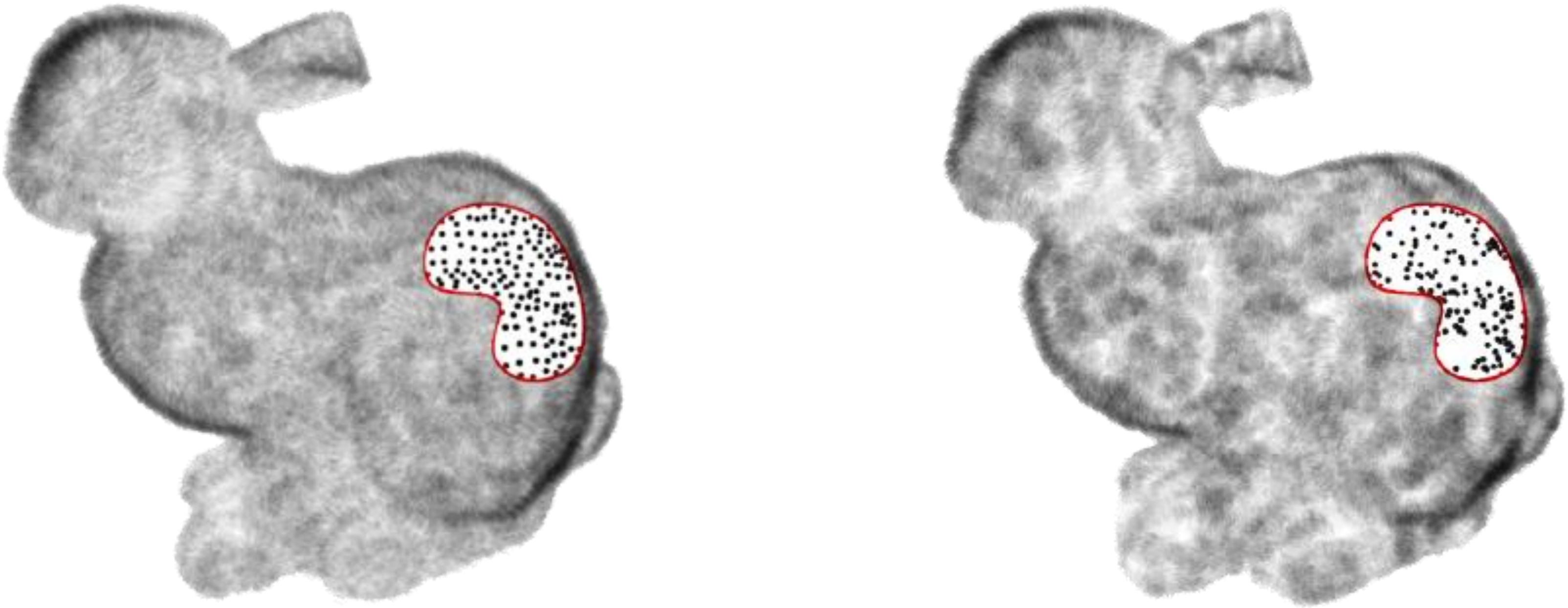
Latent Probes
Cross Attention
Self Attention
Feed Forward
Embedder
Latent Code

Decoder

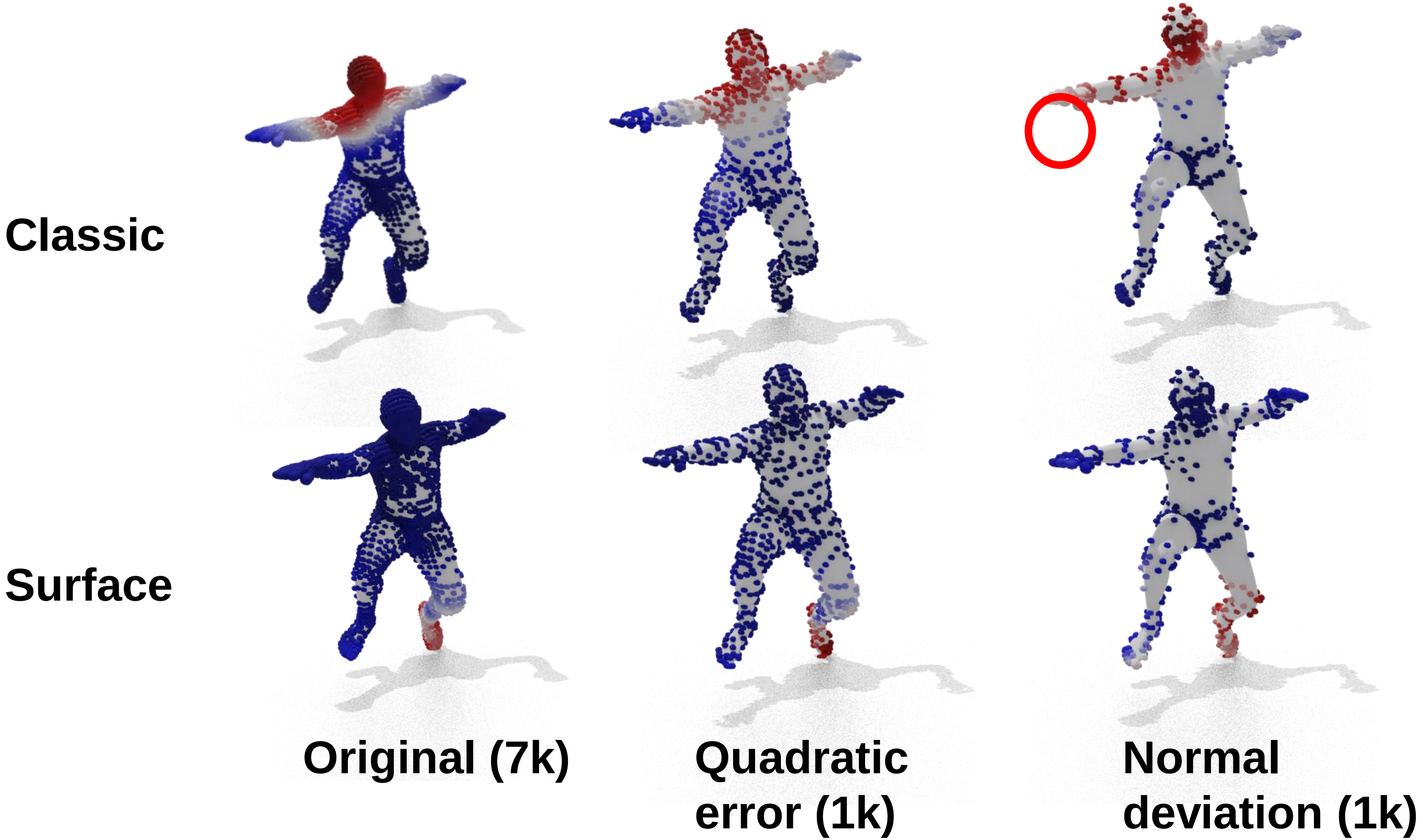


Latent
Space
MLP
Embedder
Cross
Attention
Self
Attention
Feed
Forward

Need for a different attention



Classic VS Surface Attention



Supervised loss

T S $\hat{T} = F(S)$

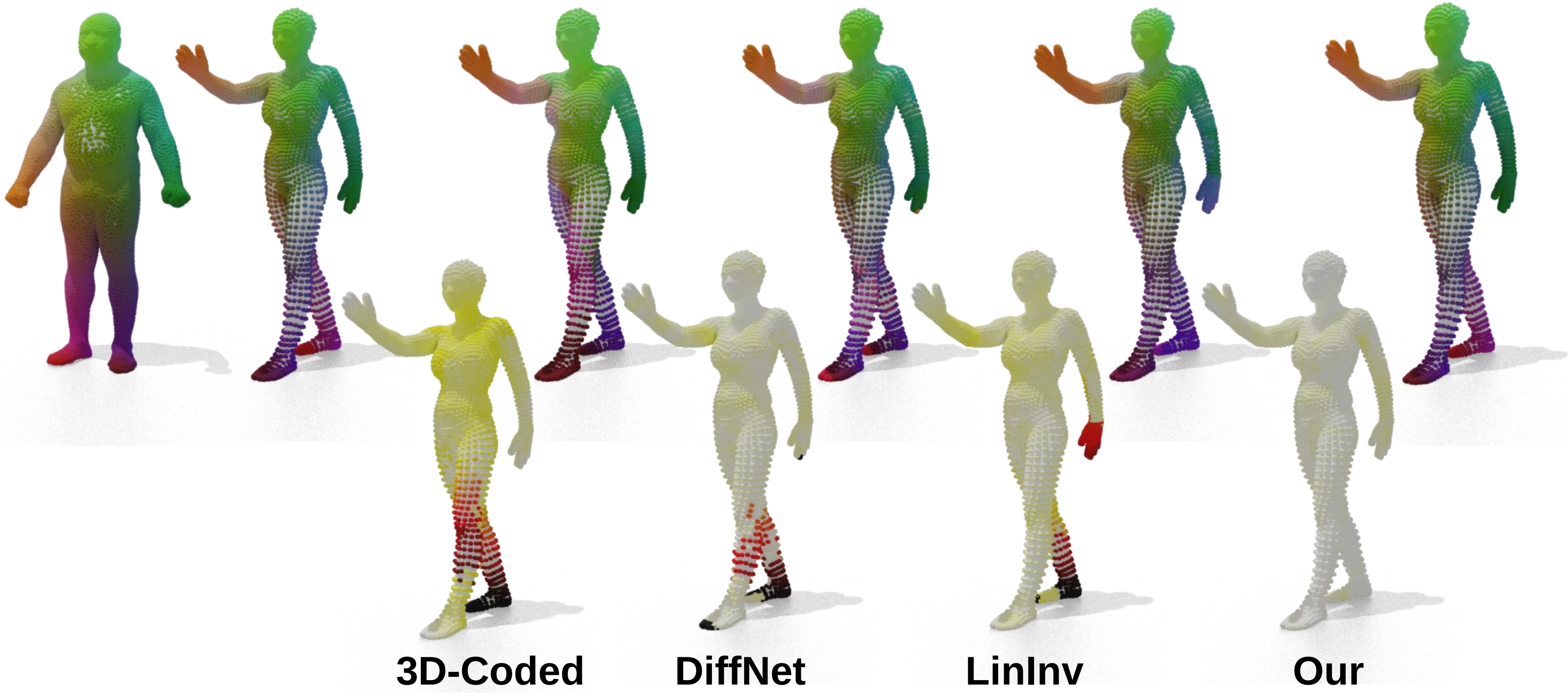
$$L = ||T - \hat{T}||_2^2$$

Unsupervised Loss

T S $\hat{T} = F(S)$

Chamfer Distance

Registration and Correspondence

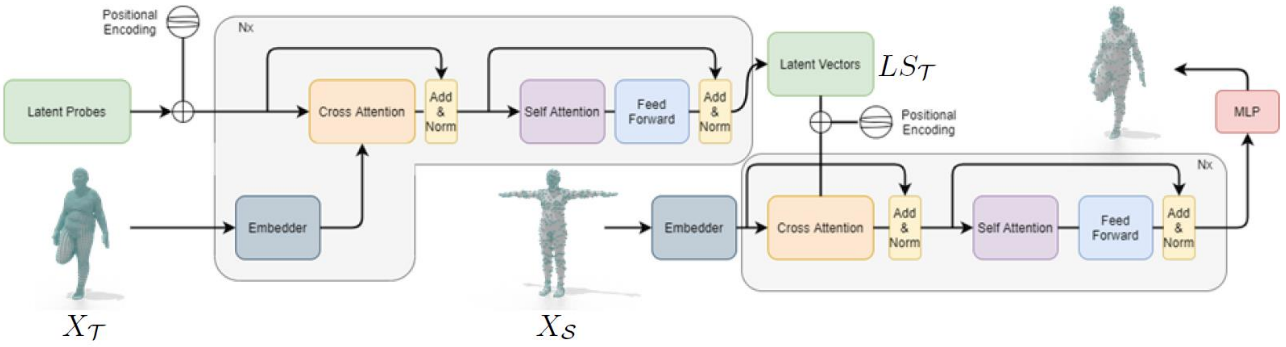


Summing up

Method	FAUST	FAUST (1K)	FAUST (noise)	SHREC 19
3DC	0.0776	0.0542	0.0712	0.2138
Diffnet	0.0656	0.0534	0.0985	0.1509
LinInv	0.0942	0.0471	0.0618	0.1284
Our	**0.0513**	**0.0419**	**0.0510**	**0.0802**
3DC - R	0.0485	0.0367	0.0526	0.1935
Our - R	**0.0369**	**0.0263**	**0.0410**	**0.0615**

- **First** Transformers for non-rigid registration.
- Introduction of an attention mechanism suitable for **surfaces.**
- Significantly **improve** on the state of the art.

Trappolini et al. NEURIPS 2021 (*SRTT*)

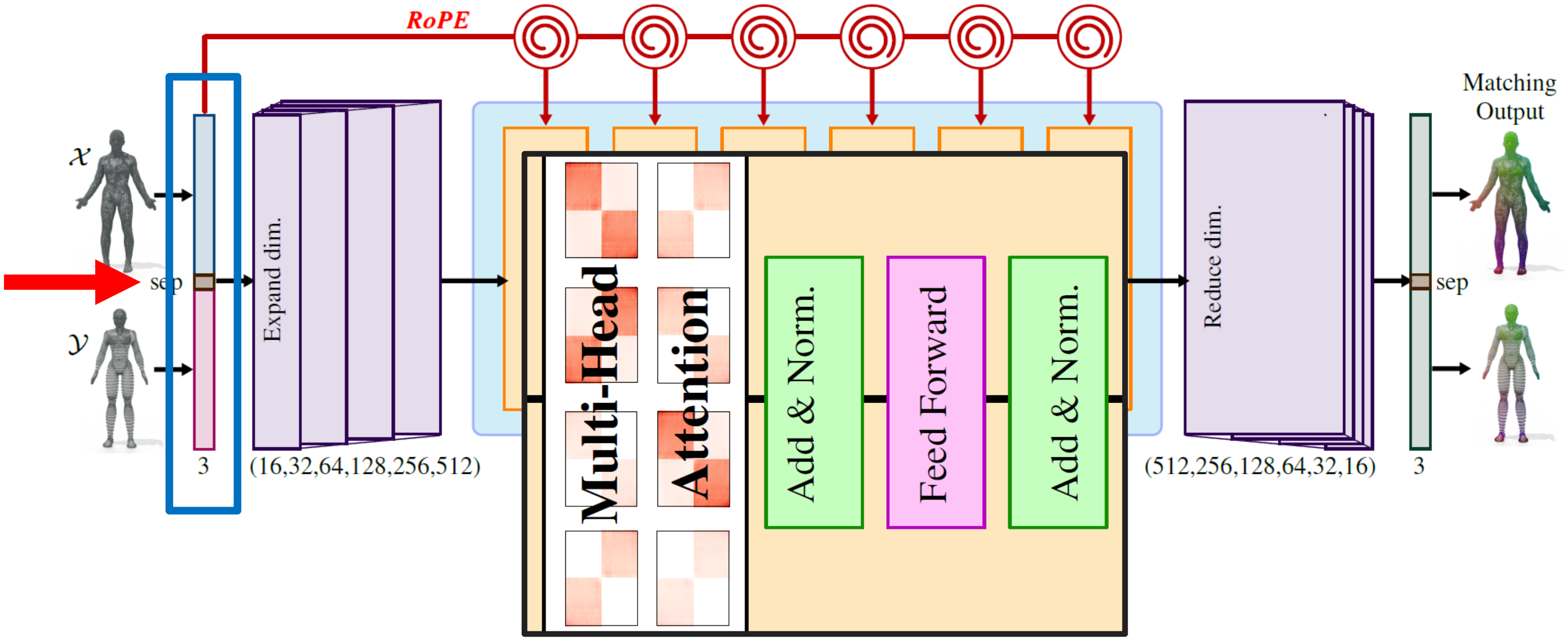


Is the first method for shape registration that exploits the transformers architecture

1. Adopts the Perceiver … Not the simplest
2. Proposes the surface attention … Geometric prior
3. Select one direction … Matching is bi-directional

Trappolini et al., "*Shape registration in the time of transformers*", NEURIPS 2021

Our Implementation

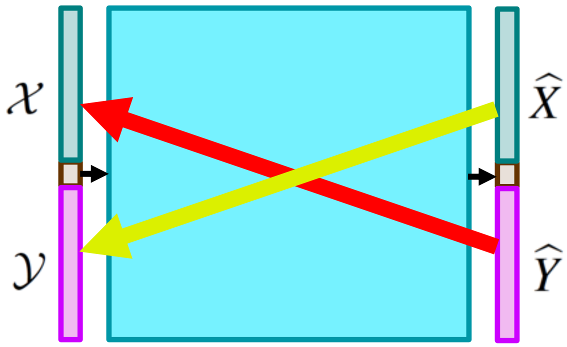


Attention and positional encoding are (almost) all you need for shape matching, Raganato, Pasi, Melzi, SGP 2023

Loss and augmentation

- The shape matching is bidirectional by nature

$$\ell = \ell_{\mathcal{X},\mathcal{Y}} + \ell_{\mathcal{Y},\mathcal{X}} = \|\widehat{Y} - \mathcal{X}\|_2^2 + \|\widehat{X} - \mathcal{Y}\|_2^2$$



- We apply a random permutation to the points representing each shape
- We apply a random rotation which belongs to one of the following types:
  1. the composition three random rotations, one for each axis in [0,2π];
  2. a random rotation along one of the axes in the interval [0,2π];
  3. the null rotation.

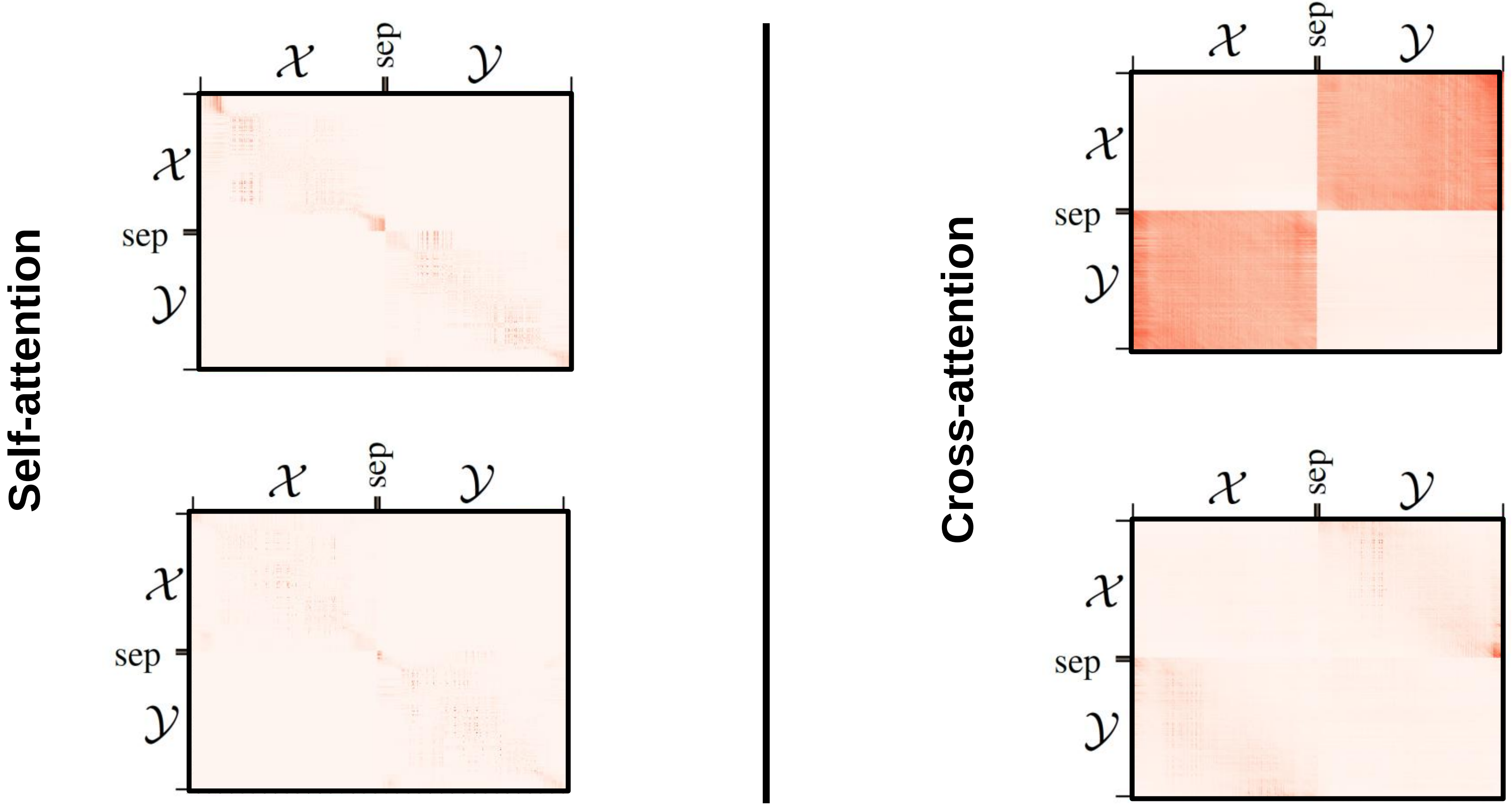
Experiments

Method	F_{1K}	$F_{1K}N$	$F_{1K}O$	$F_{\sim 7K}$	S19
3DC	0.0542	0.0712	0.2306	0.0776	0.2138
DiffNet	0.0534	0.0985	0.3509	0.0656	0.1509
LinInv	0.0471	0.0618	0.1738	0.0942	0.1284
SRTT	0.0419	0.0510	0.1657	0.0513	0.0802
Ours	**0.0135**	**0.0286**	**0.0518**	0.0236	0.0930
$3DC_R$	0.0367	0.0526	0.2101	0.0485	0.1935
$SRTT_R$	0.0263	0.0410	0.1479	0.0369	**0.0615**
$SRTT_\star$	0.0364	0.0477	0.0952	0.0436	0.0971
$Ours_\star$	**0.0133**	**0.0279**	**0.0224**	**0.0199**	0.0773

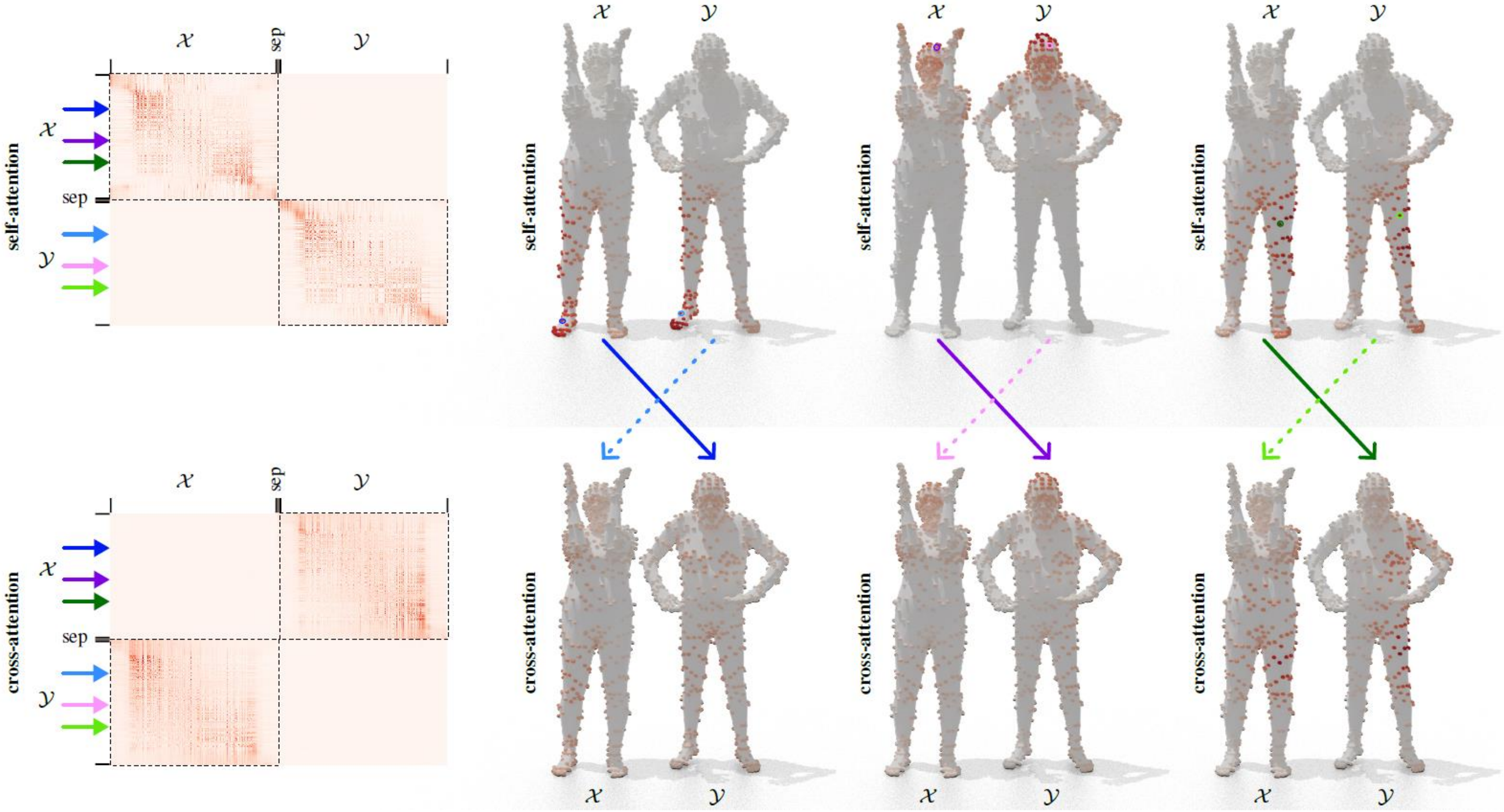
- We outperform all the competitors.
- We exceed the transformers-based method SRTT
- We are competitve or even better than method exploiting refinement.
- Our performance are better if we continue the training with a different discretization

★

Our Attention Pattern

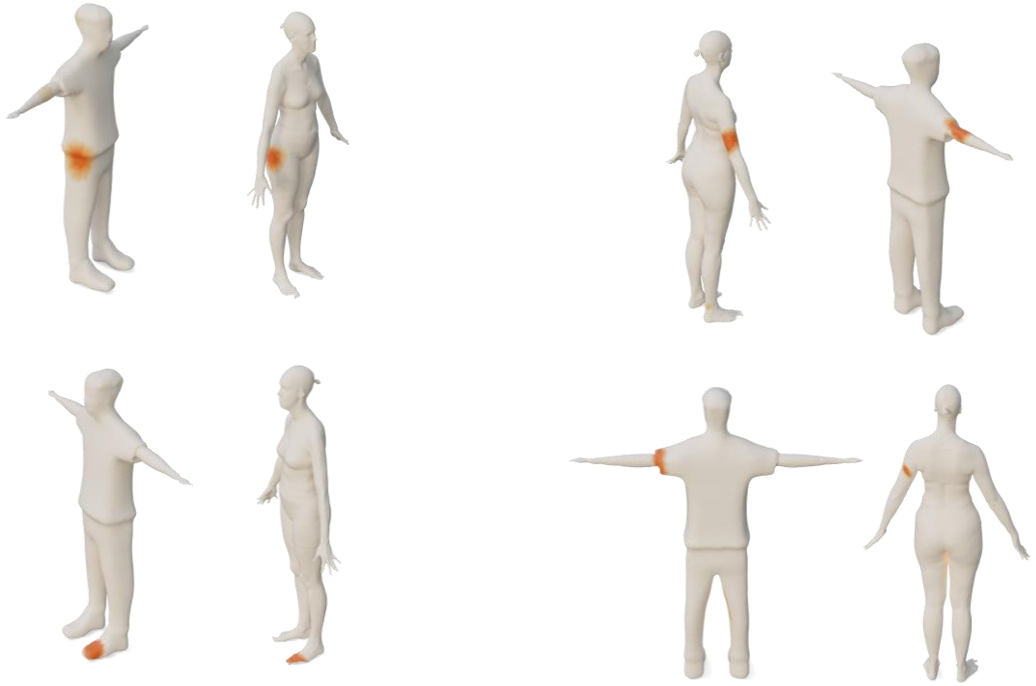


Geometry of the Attention



Visualize learned features for unsupervised Deep FMaps

Learned features tend to be consistent and well-localized even on quite non-isometric shapes.



Understanding and Improving Features Learned in Deep Functional Maps, Attaiki, et al. CVPR 2023

Transformer-based registration DEMO

https://github.com/GiovanniTRA/transmatching

Critical choices

1. The representation to encode the geometry (mesh, point clouds, …)
2. The features to inject intrinsic/extrinsic/both (coordinates, spectral, …)
3. The approach to follow (descriptors, functional maps, template registration, …)
4. The architecture to exploit (MLP, convolutions, transformers, autoencoders, …)
5. The Features extractor to adopt (MLP, PointNet, Diffusionet, …)
6. The loss to minimize (Supervised, Chamfer, regularizations,…)
7. …

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA
DIPARTIMENTO DI INFORMATICA, SISTEMISTICA E COMUNICAZIONE
Viale Sarca, 336
20126 Milano

ReGAInS

DIPARTIMENTI DI ECCELLENZA

SEMINAR ANNOUNCEMENT

Thursday February 8th, 2024

at 02:30 pm
Room "Sala Seminari" - Abacus Building (U14)

Towards general-purpose feature learning for 3D shape comparison

Speaker
Prof. Maks Ovsjanikov
Ecole Polytechnique, France