

# Deep learning on geometric data

**Davide Boscaini**

University of Lugano  
Switzerland

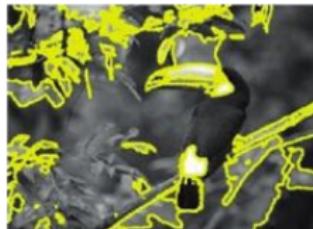


Image idea: K. Crane

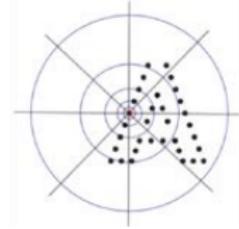
## (Hand-crafted) image descriptors



SIFT<sup>1</sup>



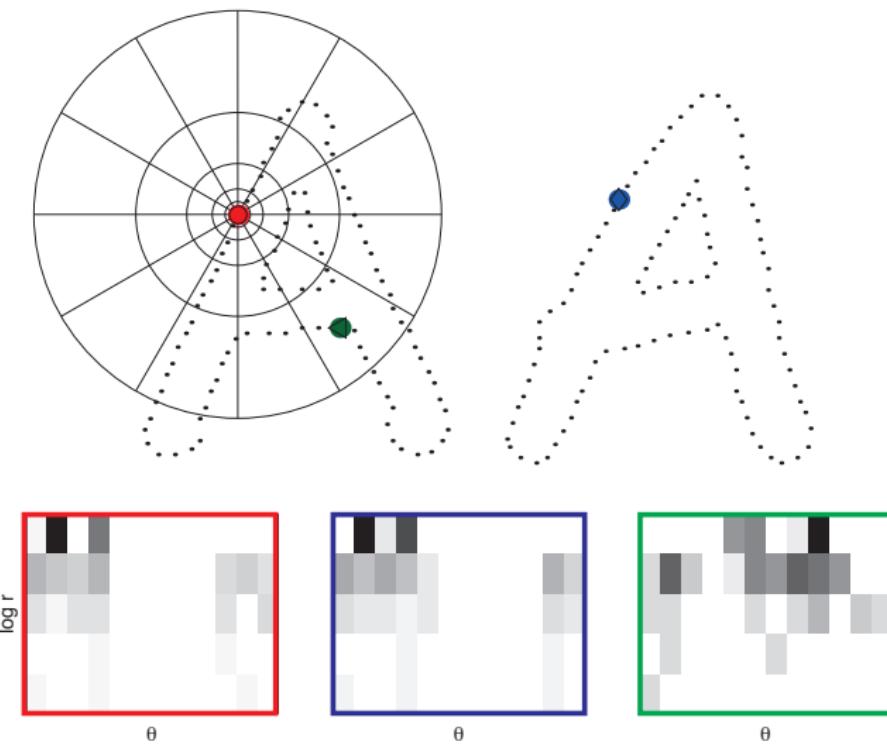
MSER<sup>2</sup>



Shape Context<sup>3</sup>

<sup>1</sup>Lowe 2004; <sup>2</sup>Matas et al. 2002; <sup>3</sup>Belongie et al. 2000

## Shape context



2012

---

## ImageNet Classification with Deep Convolutional Neural Networks

---

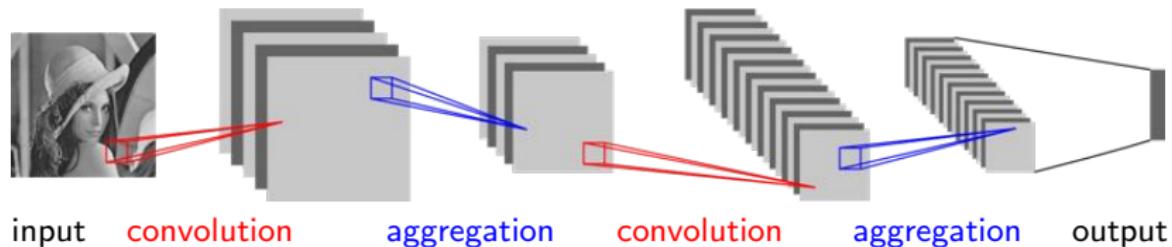
Alex Krizhevsky  
University of Toronto  
kriz@cs.utoronto.ca

Ilya Sutskever  
University of Toronto  
ilya@cs.utoronto.ca

Geoffrey E. Hinton  
University of Toronto  
hinton@cs.utoronto.ca



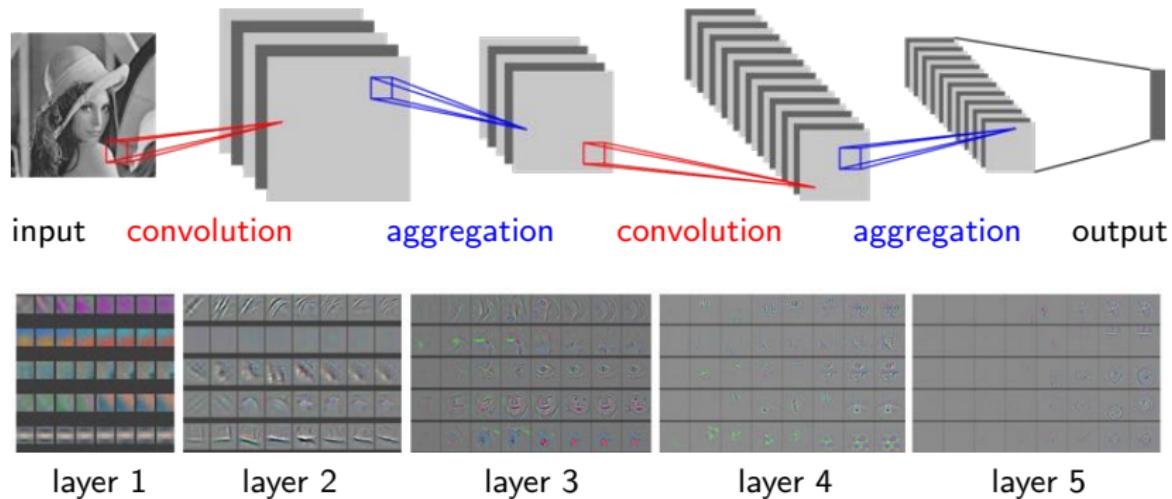
## Learning image descriptors from data



### Convolutional Neural Networks:

- Combination of **convolution** and **pooling** layers

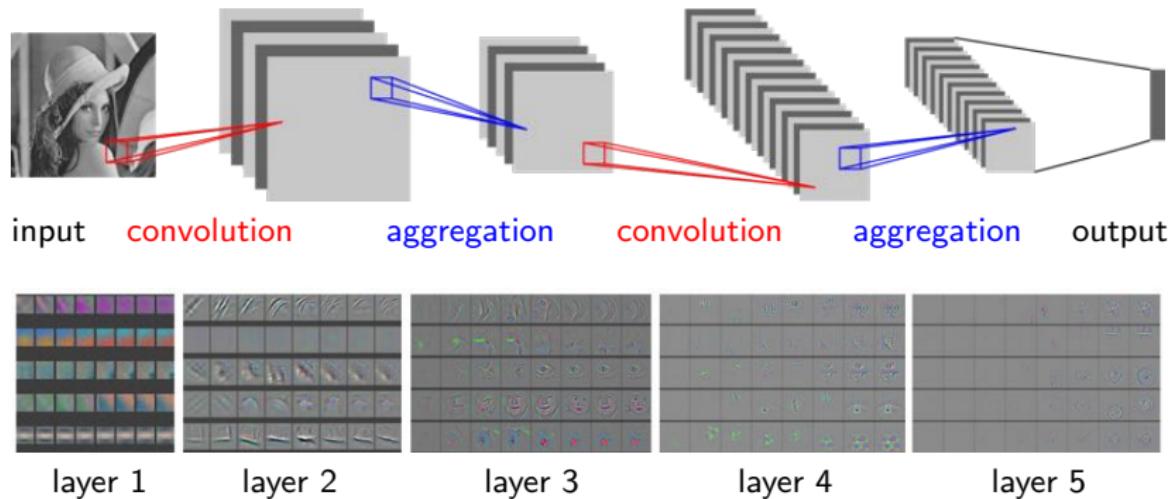
# Learning image descriptors from data



## Convolutional Neural Networks:

- Combination of **convolution** and **pooling** layers
- Learn hierarchical abstractions from data with little prior knowledge

# Learning image descriptors from data



## Convolutional Neural Networks:

- Combination of **convolution** and **pooling** layers
- Learn hierarchical abstractions from data with little prior knowledge
- State-of-the-art performance in a wide range of applications

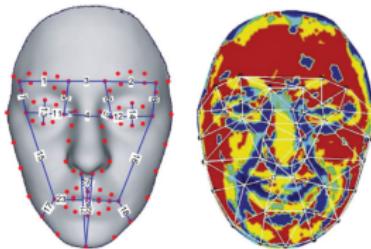


Image idea: K. Crane

# Applications



Reconstruction



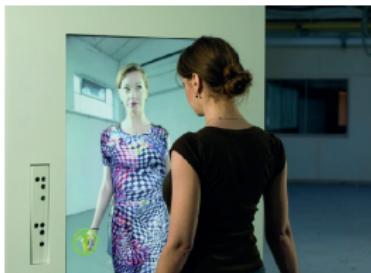
Recognition



Retrieval



Avatars

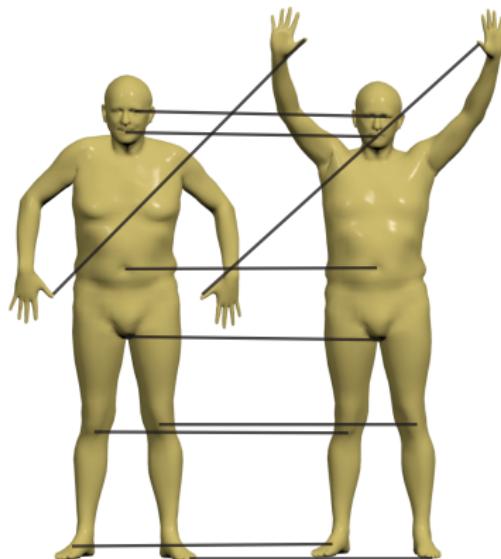


Virtual dressing



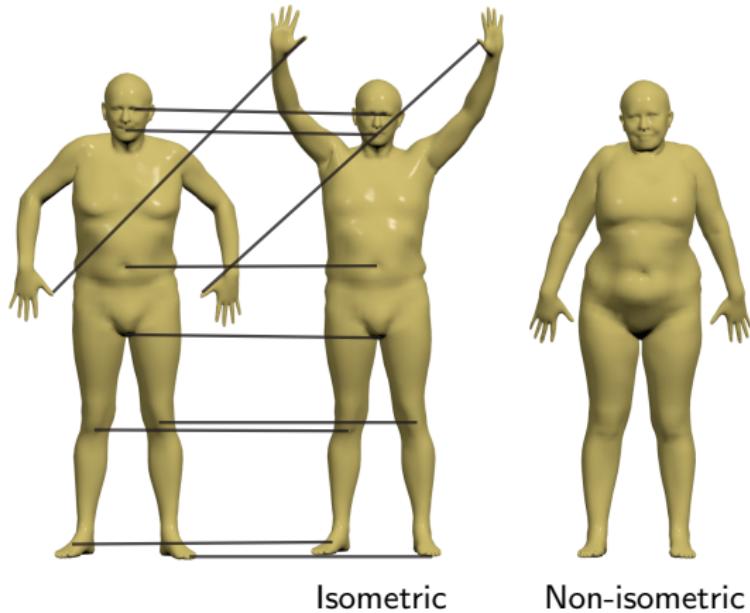
Gesture control

## Basic problems: shape similarity and correspondence

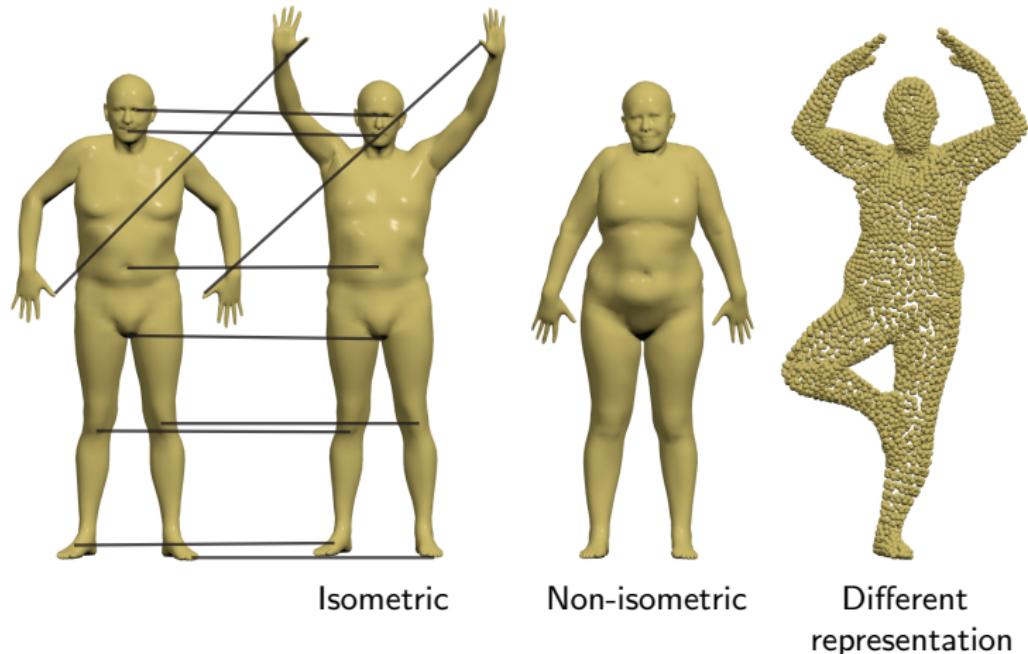


Isometric

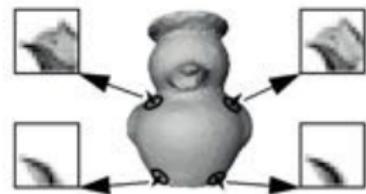
## Basic problems: shape similarity and correspondence



## Basic problems: shape similarity and correspondence



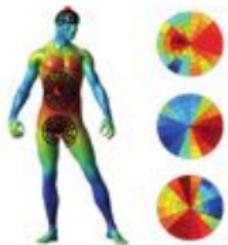
## (Hand-crafted) 3D shape descriptors



Spin image<sup>1</sup>



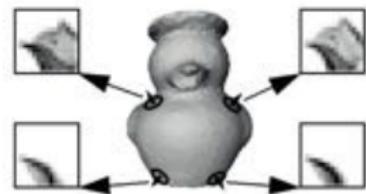
ShapeMSER<sup>2</sup>



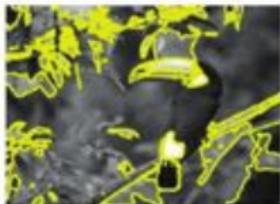
Intrinsic  
shape context<sup>3</sup>

<sup>1</sup> Johnson et al. 1999; <sup>2</sup>Litman et al. 2010; <sup>3</sup>Kokkinos et al. 2012

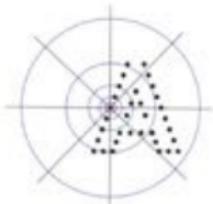
## (Hand-crafted) 3D shape descriptors



Spin image<sup>1</sup>



MSER<sup>4</sup>



Shape context<sup>5</sup>



ShapeMSER<sup>2</sup>

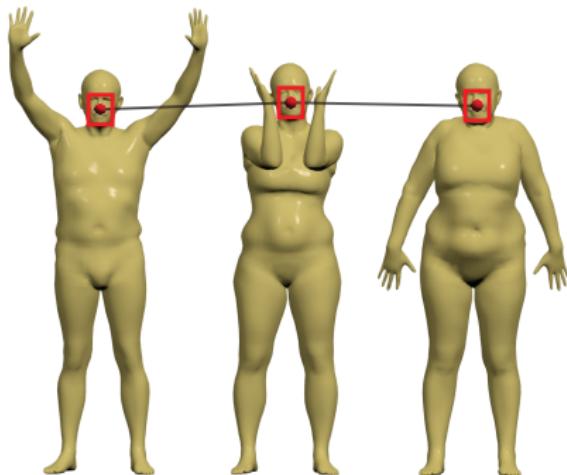


Intrinsic  
shape context<sup>3</sup>

<sup>1</sup> Johnson et al. 1999; <sup>2</sup> Litman et al. 2010; <sup>3</sup> Kokkinos et al. 2012;

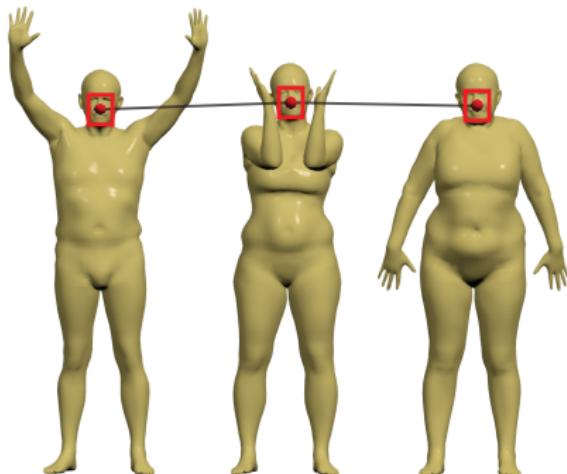
<sup>4</sup> Matas et al. 2002; <sup>5</sup> Belongie et al. 2000

### Correspondence

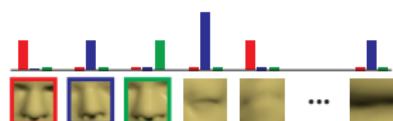


## Task-specific features

Correspondence



Similarity



# Deluge of geometric data



**KINECT**  
for XBOX 360



SoftKinetic  
The Interface Is You

**intel** REALSENSE

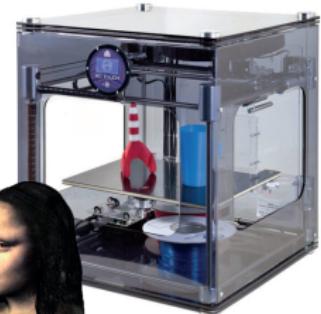
3D sensors



Google 3D warehouse

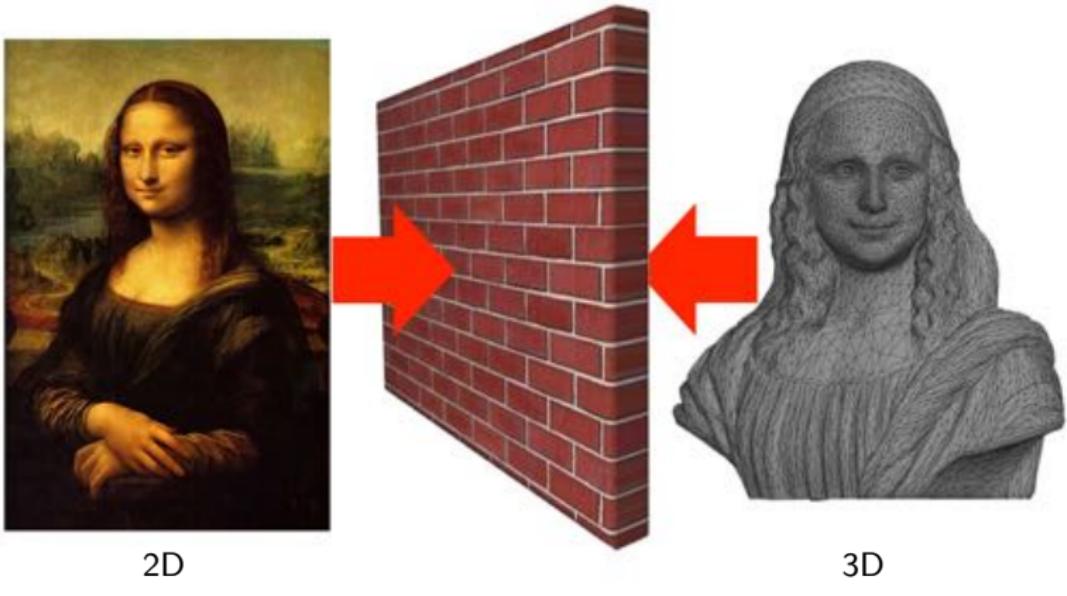
shapeways

Repositories



**Stratasys** 3D SYSTEMS

3D printers

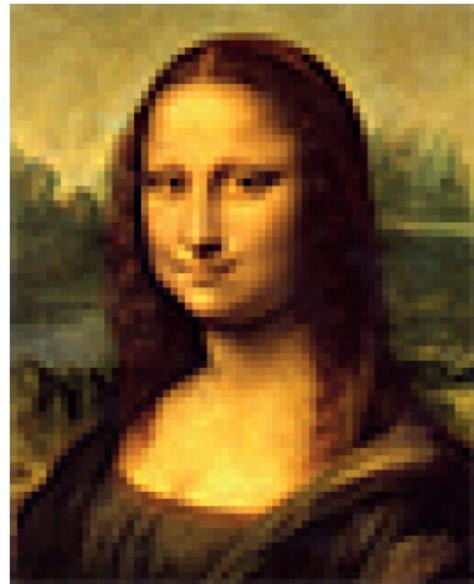


2D

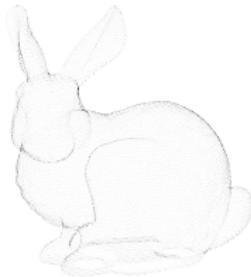
3D

Generalize deep learning to non-Euclidean data  
in a geometrically meaningful way

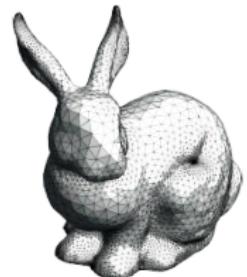
## 3D shapes vs images: representations



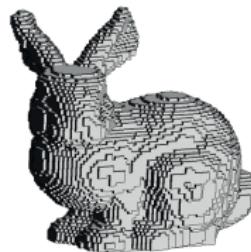
Array of pixels



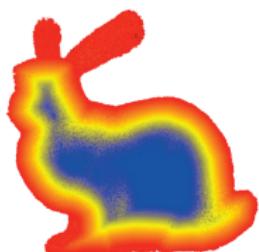
Point cloud



Mesh



Voxels

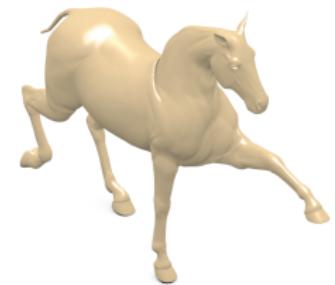


Level set

## 3D shapes vs images: transformations



Viewpoint transformations



Non-rigid deformations

## 3D shapes vs images



## 3D shapes vs images

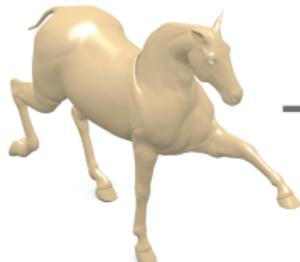
$$\frac{1}{2}$$



$$+$$

 $=$ 

$$\frac{1}{2}$$



$$+$$

 $= ?$

## Outline

- Background: definitions of manifold and Laplace-Beltrami operator

## Outline

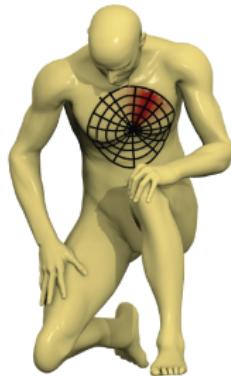
- Background: definitions of manifold and Laplace-Beltrami operator
- Spectral descriptors

## Outline

- Background: definitions of manifold and Laplace-Beltrami operator
- Spectral descriptors
- Intrinsic convolutional neural networks

# Outline

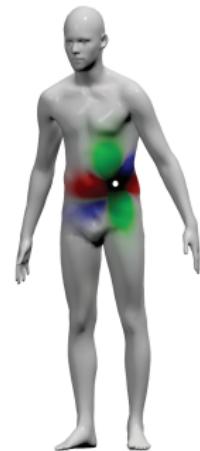
- Background: definitions of manifold and Laplace-Beltrami operator
- Spectral descriptors
- Intrinsic convolutional neural networks



Geodesic  
convolution

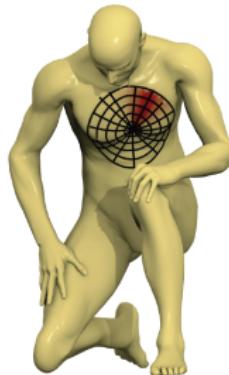


Windowed  
Fourier Transform



Anisotropic  
Diffusion

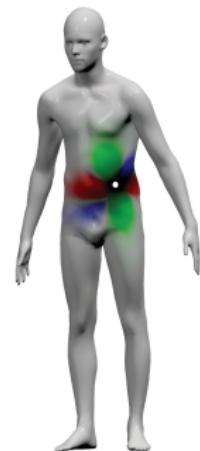
- **Background: definitions of manifold and Laplace-Beltrami operator**
- Spectral descriptors
- Intrinsic convolutional neural networks



Geodesic  
convolution

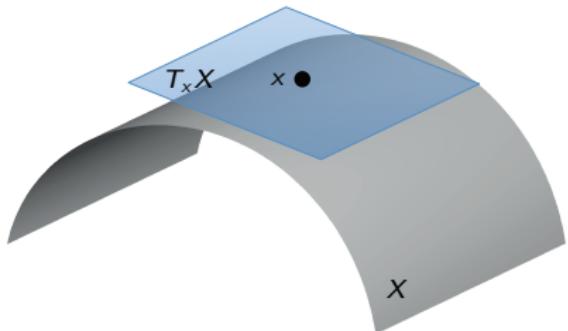


Windowed  
Fourier Transform



Anisotropic  
Diffusion

- **Tangent plane**  $T_x X$  = local Euclidean representation of manifold (surface)  $X$  around  $x$

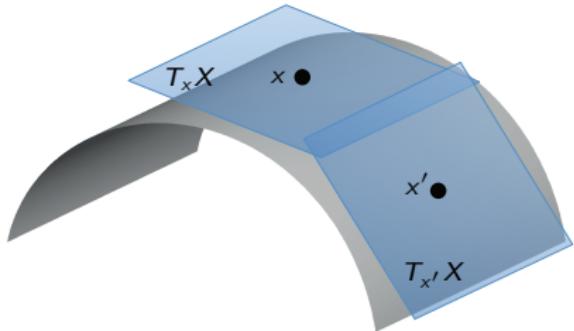


\*We assume manifolds without boundary for simplicity

- **Tangent plane**  $T_x X$  = local Euclidean representation of manifold (surface)  $X$  around  $x$
- **Riemannian metric**

$$\langle \cdot, \cdot \rangle_{T_x X} : T_x X \times T_x X \rightarrow \mathbb{R}$$

depending smoothly on  $x$



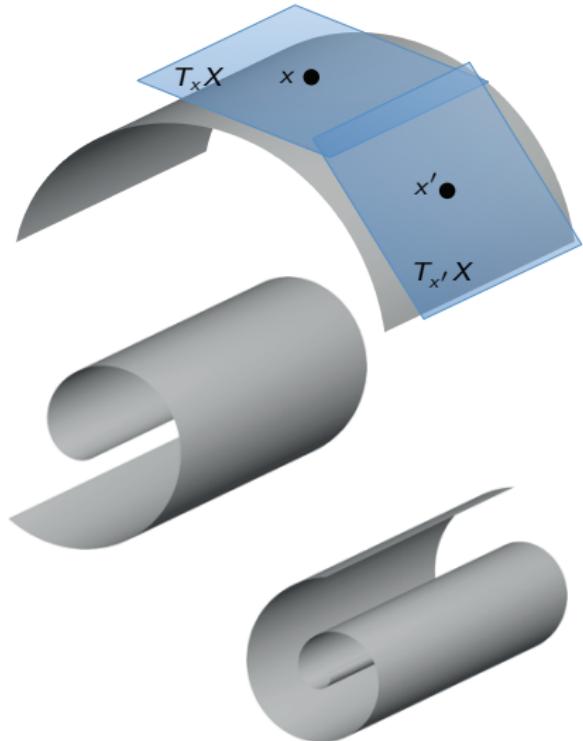
\*We assume manifolds without boundary for simplicity

- **Tangent plane**  $T_x X$  = local Euclidean representation of manifold (surface)  $X$  around  $x$
- **Riemannian metric**

$$\langle \cdot, \cdot \rangle_{T_x X} : T_x X \times T_x X \rightarrow \mathbb{R}$$

depending smoothly on  $x$

**Isometry** = metric-preserving shape deformation



\*We assume manifolds without boundary for simplicity

- **Tangent plane**  $T_x X$  = local Euclidean representation of manifold (surface)  $X$  around  $x$

- **Riemannian metric**

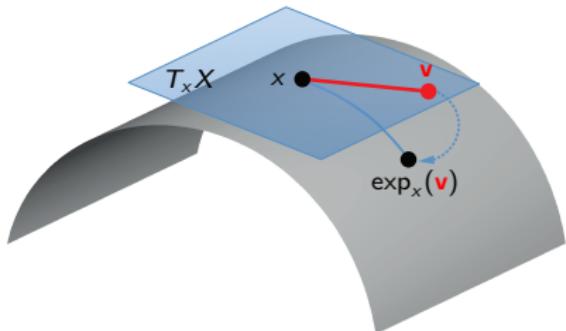
$$\langle \cdot, \cdot \rangle_{T_x X} : T_x X \times T_x X \rightarrow \mathbb{R}$$

depending smoothly on  $x$

**Isometry** = metric-preserving shape deformation

- **Exponential map**

$$\exp_x : T_x X \rightarrow X$$



\*We assume manifolds without boundary for simplicity

- **Tangent plane**  $T_x X$  = local Euclidean representation of manifold (surface)  $X$  around  $x$

- **Riemannian metric**

$$\langle \cdot, \cdot \rangle_{T_x X} : T_x X \times T_x X \rightarrow \mathbb{R}$$

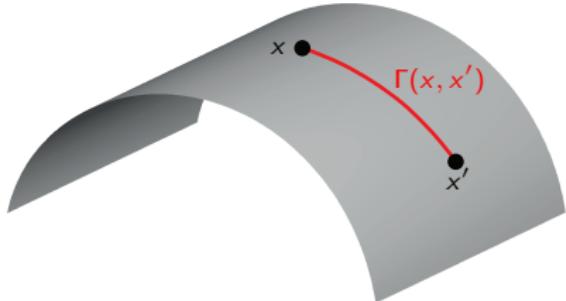
depending smoothly on  $x$

**Isometry** = metric-preserving shape deformation

- **Exponential map**

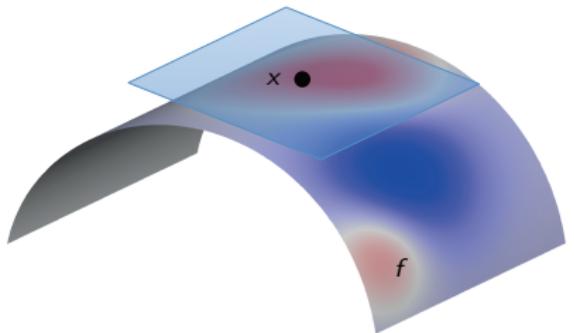
$$\exp_x : T_x X \rightarrow X$$

- **Geodesic** = shortest path on  $X$  between  $x$  and  $x'$



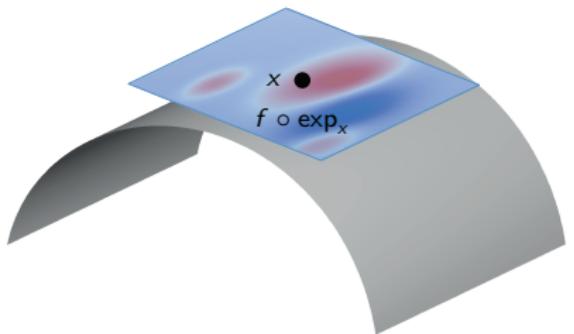
\*We assume manifolds without boundary for simplicity

## Laplace-Beltrami operator



Smooth field  $f: X \rightarrow \mathbb{R}$

## Laplace-Beltrami operator



Smooth field  $f \circ \exp_x: T_x X \rightarrow \mathbb{R}$

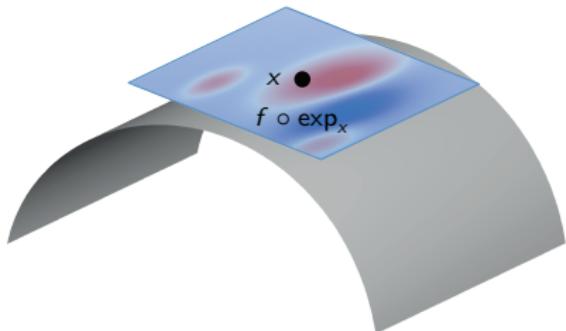
# Laplace-Beltrami operator

- **Intrinsic gradient**

$$\nabla_X f(x) = \nabla(f \circ \exp_x)(\mathbf{0})$$

Taylor expansion

$$(f \circ \exp_x)(\mathbf{v}) \approx f(x) + \langle \nabla_X f(x), \mathbf{v} \rangle_{T_x X}$$



Smooth field  $f \circ \exp_x: T_x X \rightarrow \mathbb{R}$

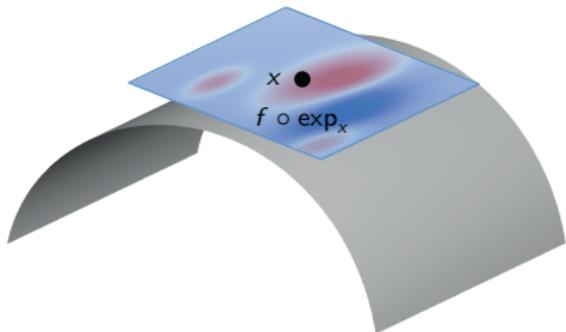
# Laplace-Beltrami operator

- **Intrinsic gradient**

$$\nabla_X f(x) = \nabla(f \circ \exp_x)(\mathbf{0})$$

Taylor expansion

$$(f \circ \exp_x)(\mathbf{v}) \approx f(x) + \langle \nabla_X f(x), \mathbf{v} \rangle_{T_x X}$$



- **Laplace-Beltrami operator**

$$\Delta_X f(x) = \Delta(f \circ \exp_x)(\mathbf{0})$$

Smooth field  $f \circ \exp_x: T_x X \rightarrow \mathbb{R}$

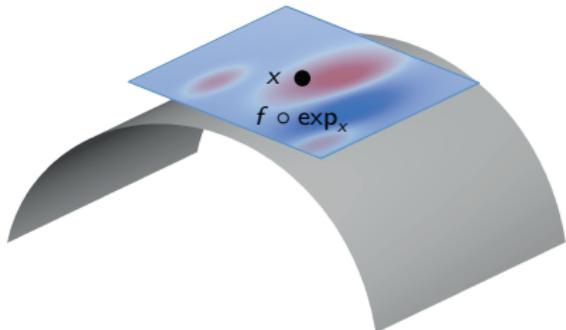
# Laplace-Beltrami operator

- **Intrinsic gradient**

$$\nabla_X f(x) = \nabla(f \circ \exp_x)(\mathbf{0})$$

Taylor expansion

$$(f \circ \exp_x)(\mathbf{v}) \approx f(x) + \langle \nabla_X f(x), \mathbf{v} \rangle_{T_x X}$$



- **Laplace-Beltrami operator**

$$\Delta_X f(x) = \Delta(f \circ \exp_x)(\mathbf{0})$$

Smooth field  $f \circ \exp_x: T_x X \rightarrow \mathbb{R}$

- Intrinsic (expressed solely in terms of the Riemannian metric)

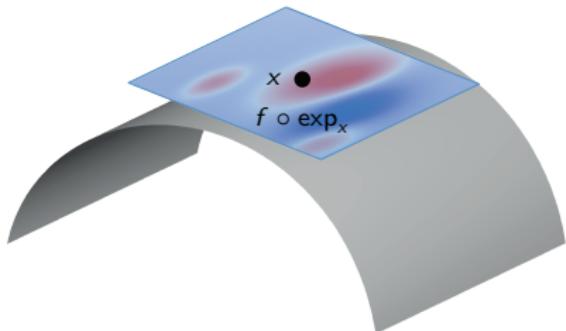
# Laplace-Beltrami operator

- **Intrinsic gradient**

$$\nabla_X f(x) = \nabla(f \circ \exp_x)(\mathbf{0})$$

Taylor expansion

$$(f \circ \exp_x)(\mathbf{v}) \approx f(x) + \langle \nabla_X f(x), \mathbf{v} \rangle_{T_x X}$$



- **Laplace-Beltrami operator**

$$\Delta_X f(x) = \Delta(f \circ \exp_x)(\mathbf{0})$$

Smooth field  $f \circ \exp_x: T_x X \rightarrow \mathbb{R}$

- Intrinsic (expressed solely in terms of the Riemannian metric)
- Isometry-invariant

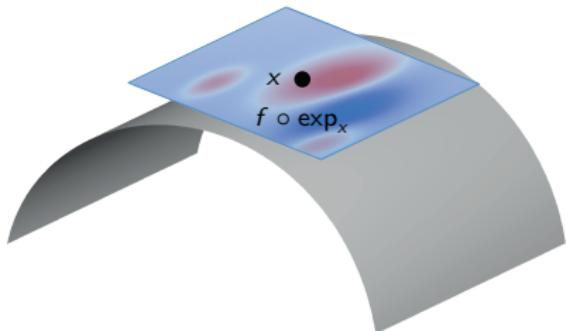
# Laplace-Beltrami operator

- **Intrinsic gradient**

$$\nabla_X f(x) = \nabla(f \circ \exp_x)(\mathbf{0})$$

Taylor expansion

$$(f \circ \exp_x)(\mathbf{v}) \approx f(x) + \langle \nabla_X f(x), \mathbf{v} \rangle_{T_x X}$$



- **Laplace-Beltrami operator**

$$\Delta_X f(x) = \Delta(f \circ \exp_x)(\mathbf{0})$$

Smooth field  $f \circ \exp_x: T_x X \rightarrow \mathbb{R}$

- Intrinsic (expressed solely in terms of the Riemannian metric)
- Isometry-invariant
- Self-adjoint  $\langle \Delta_X f, g \rangle_{L^2(X)} = \langle f, \Delta_X g \rangle_{L^2(X)} \Rightarrow$  orthogonal eigenfunctions

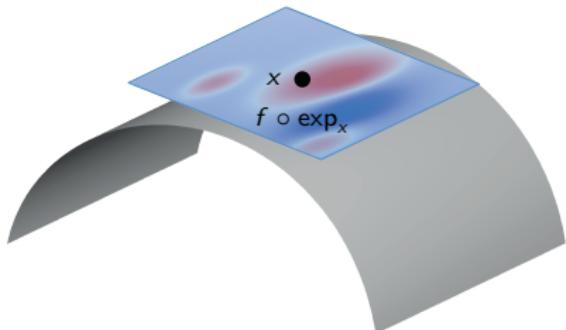
# Laplace-Beltrami operator

- **Intrinsic gradient**

$$\nabla_X f(x) = \nabla(f \circ \exp_x)(\mathbf{0})$$

Taylor expansion

$$(f \circ \exp_x)(\mathbf{v}) \approx f(x) + \langle \nabla_X f(x), \mathbf{v} \rangle_{T_x X}$$



- **Laplace-Beltrami operator**

$$\Delta_X f(x) = \Delta(f \circ \exp_x)(\mathbf{0})$$

Smooth field  $f \circ \exp_x: T_x X \rightarrow \mathbb{R}$

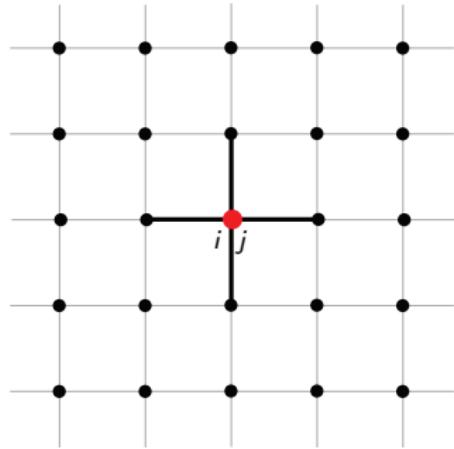
- Intrinsic (expressed solely in terms of the Riemannian metric)
- Isometry-invariant
- Self-adjoint  $\langle \Delta_X f, g \rangle_{L^2(X)} = \langle f, \Delta_X g \rangle_{L^2(X)} \Rightarrow$  orthogonal eigenfunctions
- Positive semidefinite  $\Rightarrow$  non-negative eigenvalues

# Discrete Laplacian (Euclidean)



**One-dimensional**

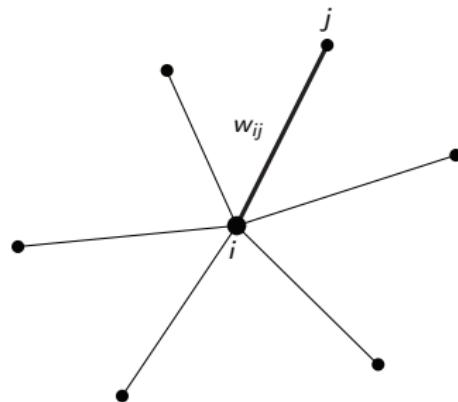
$$(\Delta f)_i \approx 2f_i - f_{i-1} - f_{i+1}$$



**Two-dimensional**

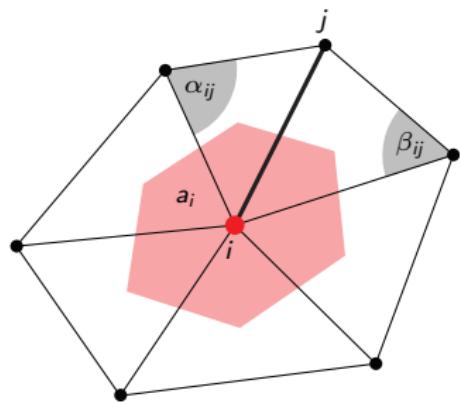
$$\begin{aligned} (\Delta f)_{ij} &\approx 4f_{ij} - f_{i-1,j} - f_{i+1,j} \\ &\quad - f_{i,j-1} - f_{i,j+1} \end{aligned}$$

# Discrete Laplacian (non-Euclidean)



**Undirected graph** ( $V, E$ )

$$(\Delta f)_i \approx \sum_{(i,j) \in E} w_{ij} (f_i - f_j)$$



**Triangular mesh** ( $V, E, F$ )

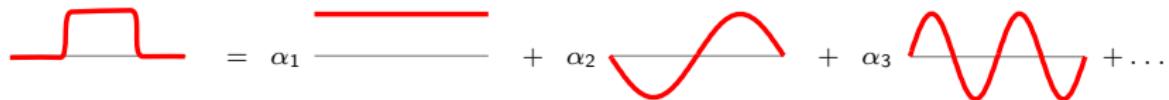
$$(\Delta f)_i \approx \frac{1}{a_i} \sum_{(i,j) \in E} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2} (f_i - f_j)$$

$a_i$  = local area element

# Fourier analysis (Euclidean spaces)

A function  $f: [-\pi, \pi] \rightarrow \mathbb{R}$  can be written as **Fourier series**

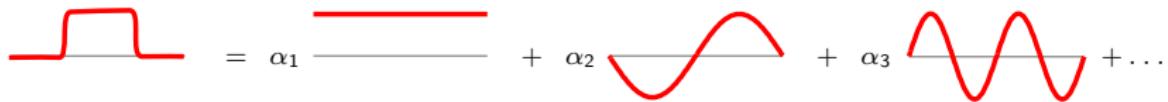
$$f(x) = \sum_{\omega} \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\xi) e^{i\omega\xi} d\xi \quad e^{-i\omega x}$$



# Fourier analysis (Euclidean spaces)

A function  $f: [-\pi, \pi] \rightarrow \mathbb{R}$  can be written as **Fourier series**

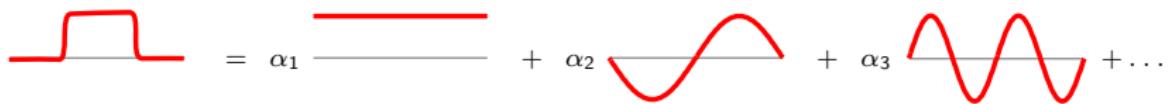
$$f(x) = \sum_{\omega} \underbrace{\frac{1}{2\pi} \int_{-\pi}^{\pi} f(\xi) e^{i\omega\xi} d\xi}_{\hat{f}(\omega)} e^{-i\omega x}$$
$$\hat{f}(\omega) = \langle f, e^{-i\omega x} \rangle_{L^2([-\pi, \pi])}$$



# Fourier analysis (Euclidean spaces)

A function  $f: [-\pi, \pi] \rightarrow \mathbb{R}$  can be written as **Fourier series**

$$f(x) = \sum_{\omega} \underbrace{\frac{1}{2\pi} \int_{-\pi}^{\pi} f(\xi) e^{i\omega\xi} d\xi}_{\hat{f}(\omega)} e^{-i\omega x}$$
$$\hat{f}(\omega) = \langle f, e^{-i\omega x} \rangle_{L^2([-\pi, \pi])}$$



Fourier basis = **Laplacian eigenfunctions**:  $\Delta e^{-i\omega x} = \omega^2 e^{-i\omega x}$

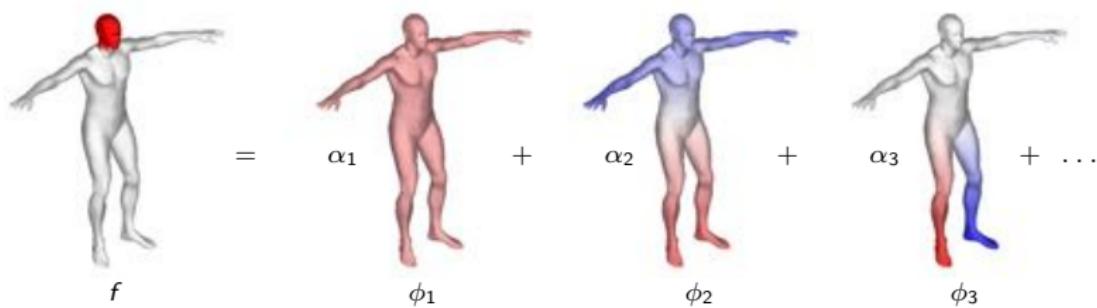
A function  $f: X \rightarrow \mathbb{R}$  can be written as **Fourier series**

$$f(x) = \sum_{k \geq 1} \underbrace{\int_X f(\xi) \phi_k(\xi) d\xi}_{\hat{f}_k = \langle f, \phi_k \rangle_{L^2(X)}} \phi_k(x)$$

## Fourier analysis (non-Euclidean spaces)

A function  $f: X \rightarrow \mathbb{R}$  can be written as **Fourier series**

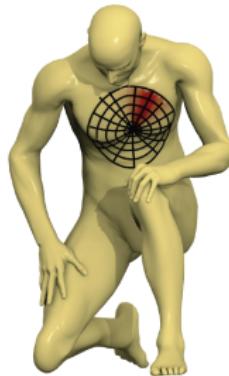
$$f(x) = \sum_{k \geq 1} \underbrace{\int_X f(\xi) \phi_k(\xi) d\xi}_{\hat{f}_k = \langle f, \phi_k \rangle_{L^2(X)}} \phi_k(x)$$



Fourier basis = **Laplacian eigenfunctions**:  $\Delta_X \phi_k(x) = \lambda_k \phi_k(x)$

# Outline

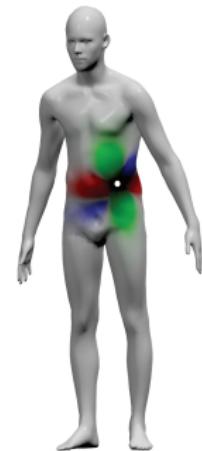
- Background: definitions of manifold and Laplace-Beltrami operator
- **Spectral descriptors**
- Intrinsic convolutional neural networks



Geodesic  
convolution



Windowed  
Fourier Transform



Anisotropic  
Diffusion

## Heat diffusion on manifolds

Heat propagation on a manifold is governed by the **heat equation**:

$$\begin{cases} \frac{\partial}{\partial t} f(x, t) = -\Delta_x f(x, t), \\ f(x, 0) = f_0(x) \end{cases}$$

- $f(x, t)$  = amount of heat at point  $x$  at time  $t$
- $f_0(x)$  = initial heat distribution

## Heat kernel signature (HKS)

Solution of the heat equation expressed through the **heat operator**

$$f(x, t) = e^{-t\Delta_x} f_0(x)$$

## Heat kernel signature (HKS)

Solution of the heat equation expressed through the **heat operator**

$$f(x, t) = e^{-t\Delta_x} f_0(x) = \int_X f_0(\xi) \sum_{k \geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi) d\xi$$

## Heat kernel signature (HKS)

Solution of the heat equation expressed through the **heat operator**

$$f(x, t) = e^{-t\Delta_x} f_0(x) = \int_X f_0(\xi) \underbrace{\sum_{k \geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi)}_{\text{heat kernel } h_t(x, \xi)} d\xi$$

## Heat kernel signature (HKS)

Solution of the heat equation expressed through the **heat operator**

$$f(x, t) = e^{-t\Delta_x} f_0(x) = \int_X f_0(\xi) \underbrace{\sum_{k \geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi)}_{\text{heat kernel } h_t(x, \xi)} d\xi$$

- “impulse response” to a delta-function at  $\xi$

## Heat kernel signature (HKS)

Solution of the heat equation expressed through the **heat operator**

$$f(x, t) = e^{-t\Delta_x} f_0(x) = \int_X f_0(\xi) \underbrace{\sum_{k \geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi)}_{\text{heat kernel } h_t(x, \xi)} d\xi$$

- “impulse response” to a delta-function at  $\xi$
- “how much heat is transferred from point  $x$  to  $\xi$  in time  $t$ ”

## Heat kernel signature (HKS)

Solution of the heat equation expressed through the **heat operator**

$$f(x, t) = e^{-t\Delta_x} f_0(x) = \int_X f_0(\xi) \underbrace{\sum_{k \geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi)}_{\text{heat kernel } h_t(x, \xi)} d\xi$$

- “impulse response” to a delta-function at  $\xi$
- “how much heat is transferred from point  $x$  to  $\xi$  in time  $t$ ”

**Autodiffusivity** = diagonal of the heat kernel

$$h_t(x, x) = \sum_{k \geq 1} e^{-t\lambda_k} \phi_k^2(x)$$

## Heat kernel signature (HKS)

Solution of the heat equation expressed through the **heat operator**

$$f(x, t) = e^{-t\Delta_x} f_0(x) = \int_X f_0(\xi) \underbrace{\sum_{k \geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi)}_{\text{heat kernel } h_t(x, \xi)} d\xi$$

- “impulse response” to a delta-function at  $\xi$
- “how much heat is transferred from point  $x$  to  $\xi$  in time  $t$ ”

**Autodiffusivity** = diagonal of the heat kernel

$$h_t(x, x) = \sum_{k \geq 1} e^{-t\lambda_k} \phi_k^2(x)$$

- “how much heat remains at point  $x$  after time  $t$ ”

## Heat kernel signature (HKS)

Solution of the heat equation expressed through the **heat operator**

$$f(x, t) = e^{-t\Delta_x} f_0(x) = \int_X f_0(\xi) \underbrace{\sum_{k \geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi)}_{\text{heat kernel } h_t(x, \xi)} d\xi$$

- “impulse response” to a delta-function at  $\xi$
- “how much heat is transferred from point  $x$  to  $\xi$  in time  $t$ ”

**Autodiffusivity** = diagonal of the heat kernel

$$h_t(x, x) = \sum_{k \geq 1} e^{-t\lambda_k} \phi_k^2(x)$$

- “how much heat remains at point  $x$  after time  $t$ ”

**Q-dim heat kernel signature (HKS)**

$$\mathbf{f}: x \rightarrow (h_{t_1}(x, x), h_{t_2}(x, x), \dots, h_{t_Q}(x, x))$$

## Spectral descriptors

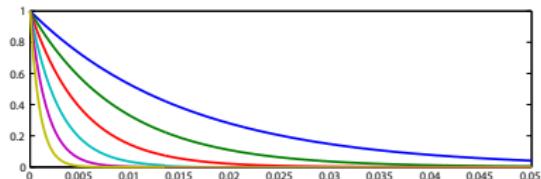
$$f(x) = \sum_{k \geq 1} \begin{pmatrix} \tau_1(\lambda_k) \\ \vdots \\ \tau_Q(\lambda_k) \end{pmatrix} \phi_k^2(x)$$

## Spectral descriptors

$$f(x) = \sum_{k \geq 1} \begin{pmatrix} \tau_1(\lambda_k) \\ \vdots \\ \tau_Q(\lambda_k) \end{pmatrix} \phi_k^2(x)$$

### Heat kernel signature (HKS)

$$\tau_i(\lambda) = \exp(-t_i \lambda)$$



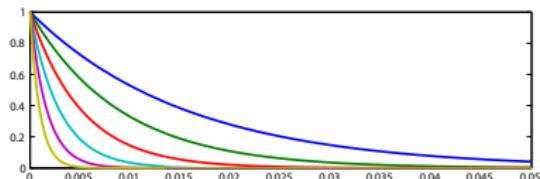
low-pass filter bank

# Spectral descriptors

$$f(x) = \sum_{k \geq 1} \begin{pmatrix} \tau_1(\lambda_k) \\ \vdots \\ \tau_Q(\lambda_k) \end{pmatrix} \phi_k^2(x)$$

## Heat kernel signature (HKS)

$$\tau_i(\lambda) = \exp(-t_i \lambda)$$



low-pass filter bank

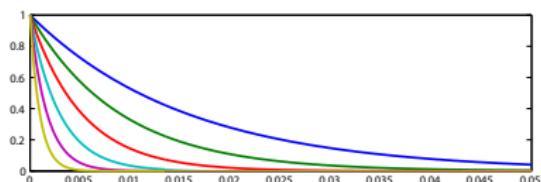


# Spectral descriptors

$$f(x) = \sum_{k \geq 1} \begin{pmatrix} \tau_1(\lambda_k) \\ \vdots \\ \tau_Q(\lambda_k) \end{pmatrix} \phi_k^2(x)$$

**Heat kernel signature (HKS)**

$$\tau_i(\lambda) = \exp(-t_i \lambda)$$

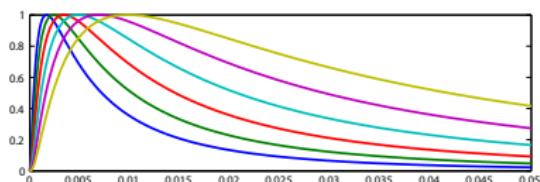


low-pass filter bank

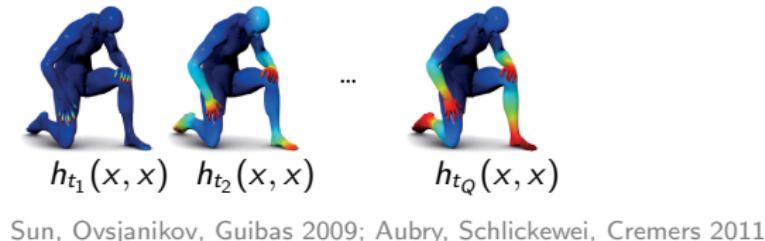


**Wave kernel signature (WKS)**

$$\tau_i(\lambda) = \exp\left(-\frac{(\log e - \log \lambda)^2}{2\sigma^2}\right)$$



band-pass filter bank



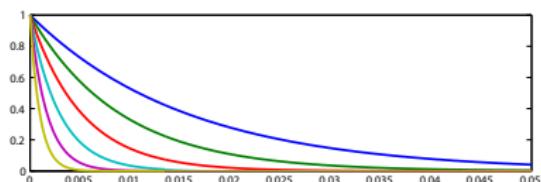
Sun, Ovsjanikov, Guibas 2009; Aubry, Schlickewei, Cremers 2011

# Spectral descriptors

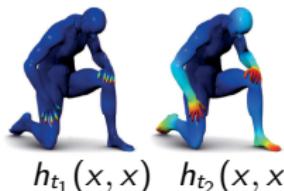
$$f(x) = \sum_{k \geq 1} \begin{pmatrix} \tau_1(\lambda_k) \\ \vdots \\ \tau_Q(\lambda_k) \end{pmatrix} \phi_k^2(x)$$

**Heat kernel signature (HKS)**

$$\tau_i(\lambda) = \exp(-t_i \lambda)$$

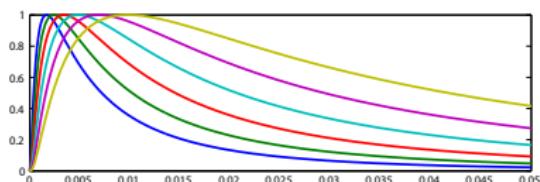


low-pass filter bank

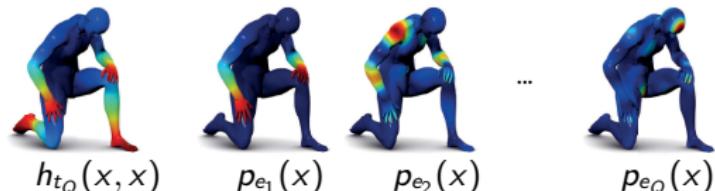


**Wave kernel signature (WKS)**

$$\tau_i(\lambda) = \exp\left(-\frac{(\log e - \log \lambda)^2}{2\sigma^2}\right)$$



band-pass filter bank



Sun, Ovsjanikov, Guibas 2009; Aubry, Schlickewei, Cremers 2011

## Optimal spectral descriptor

A generic  $Q$ -dimensional spectral descriptor of the form

$$\mathbf{f}_\tau(x) = \sum_{k \geq 1} \begin{pmatrix} \tau_1(\lambda_k) \\ \vdots \\ \tau_Q(\lambda_k) \end{pmatrix} \phi_k^2(x)$$

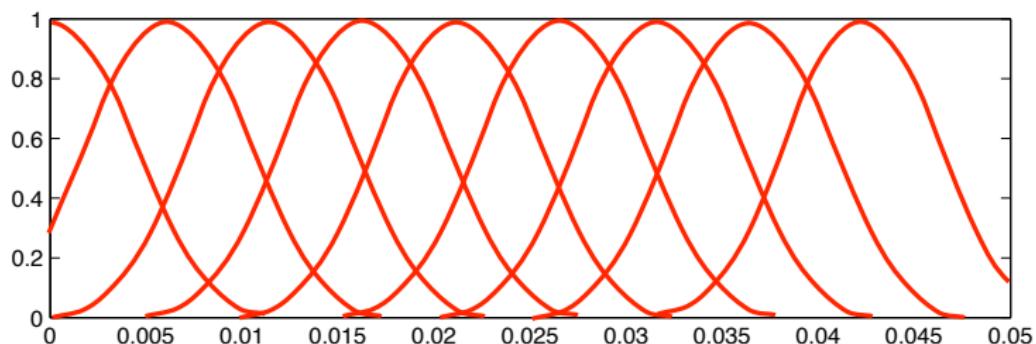
can be parametrized by frequency responses  $\tau(\lambda) = (\tau_1(\lambda), \dots, \tau_Q(\lambda))^\top$

## Optimal spectral descriptor

A generic  $Q$ -dimensional spectral descriptor of the form

$$f_{\mathbf{A}}(x) = \sum_{k \geq 1} \mathbf{A} \begin{pmatrix} \beta_1(\lambda_k) \\ \vdots \\ \beta_M(\lambda_k) \end{pmatrix} \phi_k^2(x)$$

can be parametrized by frequency responses  $\tau(\lambda) = (\tau_1(\lambda), \dots, \tau_Q(\lambda))^{\top}$   
represented in some fixed basis  $\beta_1(\lambda), \dots, \beta_M(\lambda)$  by an  $Q \times M$  matrix  $\mathbf{A}$



## Optimal spectral descriptor

A generic  $Q$ -dimensional spectral descriptor of the form

$$f_A(x) = A \underbrace{\sum_{k \geq 1} \begin{pmatrix} \beta_1(\lambda_k) \\ \vdots \\ \beta_M(\lambda_k) \end{pmatrix}}_{g(x)} \phi_k^2(x)$$

parametrized by linear combination coefficients  $A$  of **geometry vectors**

$$g(x) = (g_1(x), \dots, g_M(x))^\top$$

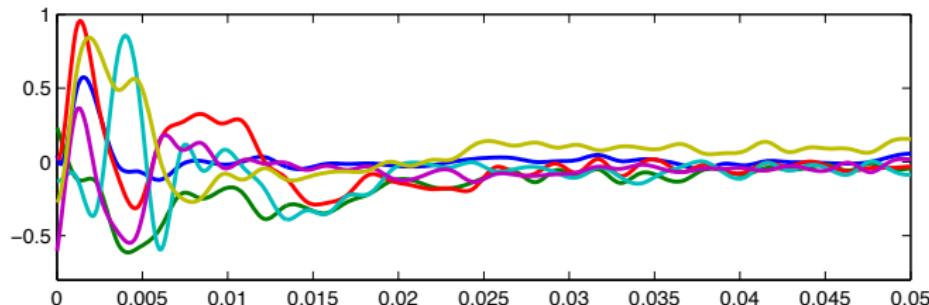
## Optimal spectral descriptor

A generic  $Q$ -dimensional spectral descriptor of the form

$$f_A(x) = A \sum_{k \geq 1} \underbrace{\begin{pmatrix} \beta_1(\lambda_k) \\ \vdots \\ \beta_M(\lambda_k) \end{pmatrix}}_{g(x)} \phi_k^2(x)$$

parametrized by linear combination coefficients  $A$  of **geometry vectors**  
 $g(x) = (g_1(x), \dots, g_M(x))^\top$

Hard to model axiomatically... yet easy to **learn** from examples!



# Outline

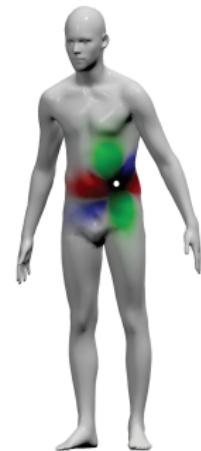
- Background: definitions of manifold and Laplace-Beltrami operator
- Spectral descriptors
- **Intrinsic convolutional neural networks**



**Geodesic  
convolution**

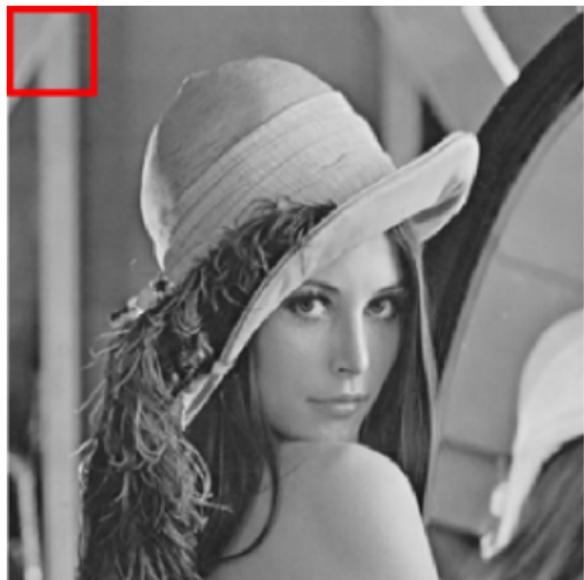


**Windowed  
Fourier Transform**



**Anisotropic  
Diffusion**

## Convolution on manifolds



# Convolution on manifolds



## Convolution on manifolds



# Convolution on manifolds



?

## Geodesic polar coordinates

- Local system of **geodesic polar coordinates** at  $x$ 
  - $\rho$ -level set of geodesic distance function  $d_X(x, \xi)$ , truncated at  $\rho_0$
  - points along geodesic  $\Gamma_\theta(x)$  emanating from  $x$  in direction  $\theta$

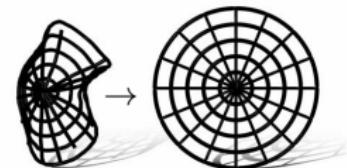


# Geodesic polar coordinates

- Local system of **geodesic polar coordinates** at  $x$ 
  - $\rho$ -level set of geodesic distance function  $d_X(x, \xi)$ , truncated at  $\rho_0$
  - points along geodesic  $\Gamma_\theta(x)$  emanating from  $x$  in direction  $\theta$
- Local chart: bijective map

$$\Omega(x) : B_{\rho_0}(x) \rightarrow [0, \rho_0] \times [0, 2\pi)$$

from manifold to local coordinates  $(\rho, \theta)$   
around  $x$



# Geodesic polar coordinates

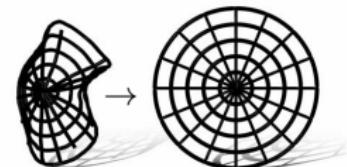
- Local system of **geodesic polar coordinates** at  $x$ 
  - $\rho$ -level set of geodesic distance function  $d_X(x, \xi)$ , truncated at  $\rho_0$
  - points along geodesic  $\Gamma_\theta(x)$  emanating from  $x$  in direction  $\theta$
- Local chart: bijective map

$$\Omega(x) : B_{\rho_0}(x) \rightarrow [0, \rho_0] \times [0, 2\pi)$$

from manifold to local coordinates  $(\rho, \theta)$   
around  $x$

- **Patch operator** applied to  $f \in L^2(X)$

$$(D(x)f)(\rho, \theta) = (f \circ \Omega^{-1}(x))(\rho, \theta)$$



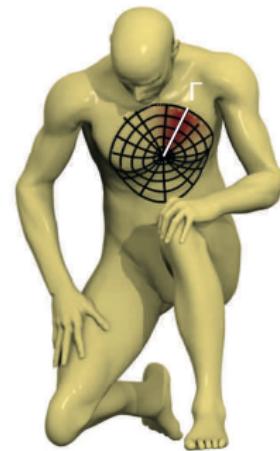
## Patch operator construction

$$(D(x)f)(\rho, \theta) = \frac{\int_X v_\rho(x, \xi) v_\theta(x, \xi) f(\xi) d\xi}{\int_X v_\rho(x, \xi) v_\theta(x, \xi) d\xi}$$



Radial weight

$$v_\rho(x, \xi) \propto e^{-(d_X(x, \xi) - \rho)^2 / \sigma_\rho^2}$$



Angular weight

$$v_\theta(x, \xi) \propto e^{-d_X^2(\Gamma(x, \theta), \xi) / \sigma_\theta^2}$$

## Geodesic convolution

- **Geodesic convolution** = apply filter  $a$  to patches extracted from  $f \in L^2(X)$  in local geodesic polar coordinates

$$(f * a)(x) = \sum_{\theta, r} (D(x)f)(r, \theta) a(\theta, r)$$

## Geodesic convolution

- **Geodesic convolution** = apply filter  $a$  to patches extracted from  $f \in L^2(X)$  in local geodesic polar coordinates

$$(f * a)(x) = \sum_{\theta, r} \underbrace{(D(x)f)(r, \theta)}_{\text{Patch}} \underbrace{a(\theta, r)}_{\text{Filter}}$$



## Geodesic convolution

- **Geodesic convolution** = apply filter  $a$  to patches extracted from  $f \in L^2(X)$  in local geodesic polar coordinates

$$(f * a)(x) = \sum_{\theta, r} \underbrace{(D(x)f)(r, \theta)}_{\text{Patch}} \underbrace{a(\theta + \Delta\theta, r)}_{\text{Filter}}$$

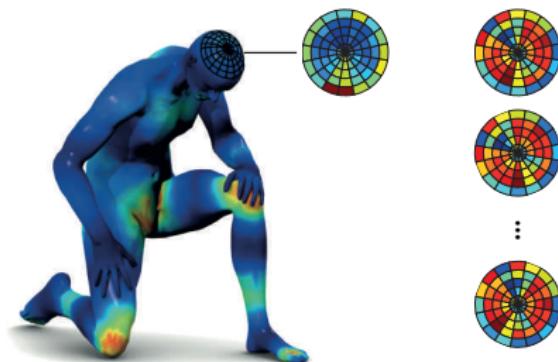


- Angular coordinate origin is arbitrary = **rotation ambiguity!**

## Geodesic convolution

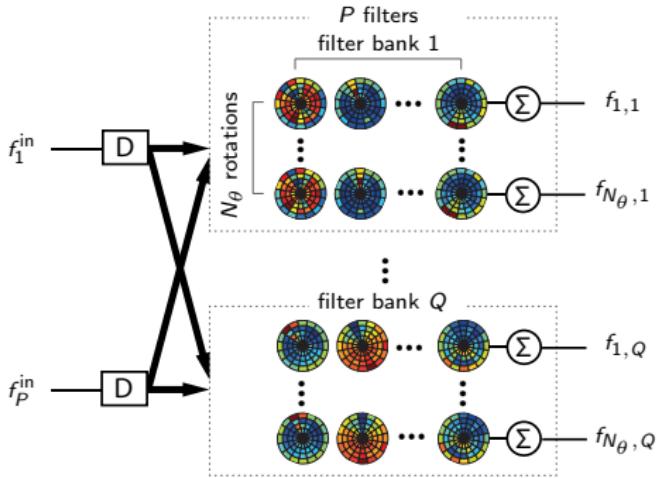
- **Geodesic convolution** = apply filter  $a$  to patches extracted from  $f \in L^2(X)$  in local geodesic polar coordinates

$$(f * a)(x) = \sum_{\theta, r} \underbrace{(D(x)f)(r, \theta)}_{\text{Patch}} \underbrace{a(\theta + \Delta\theta, r)}_{\text{Filter}}$$



- Angular coordinate origin is arbitrary = **rotation ambiguity!**
- Keep all possible rotations

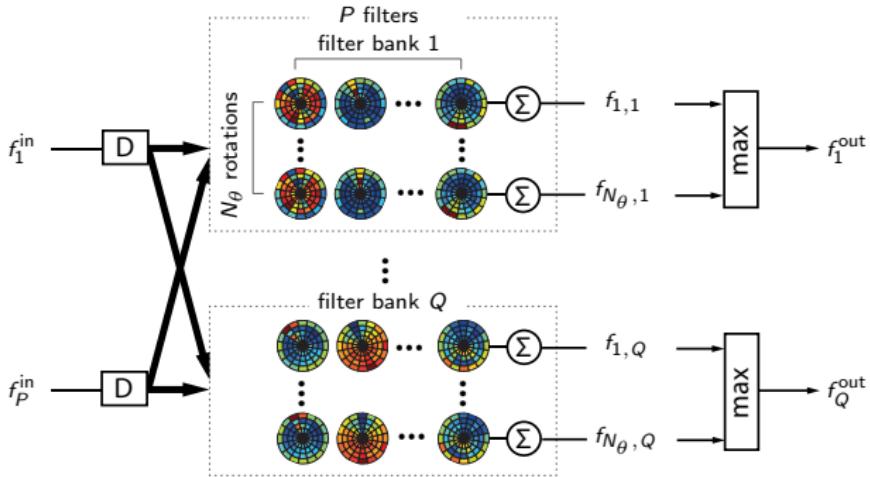
# Geodesic Convolution layer



$$f_{\Delta\theta,q}^{\text{out}}(x) = \sum_{p=1}^P (f_p \star a_{\Delta\theta,qp})(x), \quad q = 1, \dots, Q$$

- $a_{\Delta\theta,qp}(\theta, r) = a_{qp}(\theta + \Delta\theta, r)$  are coefficients of  $p$ th filter in  $q$ th filter bank rotated by  $\Delta\theta$

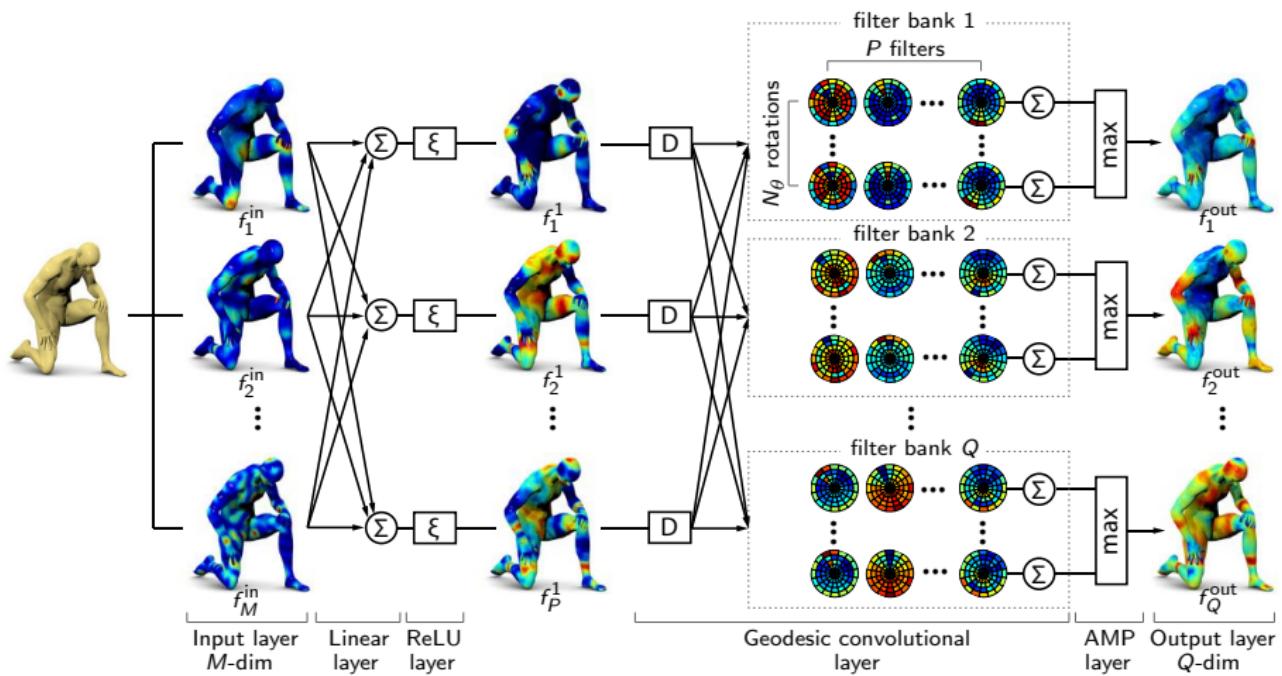
# Geodesic Convolution layer



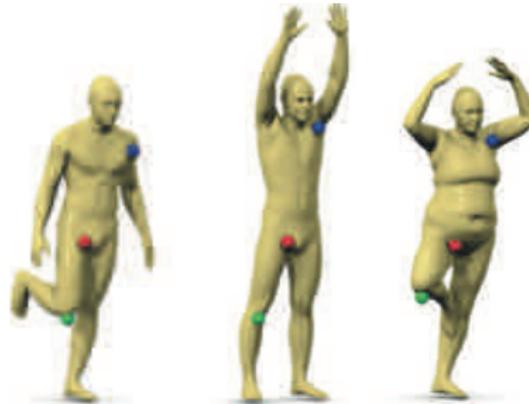
$$f_q^{\text{out}}(x) = \max_{\Delta\theta} \sum_{p=1}^P (f_p * a_{\Delta\theta, qp})(x), \quad q = 1, \dots, Q$$

- $a_{\Delta\theta, qp}(\theta, r) = a_{qp}(\theta + \Delta\theta, r)$  are coefficients of  $p$ th filter in  $q$ th filter bank rotated by  $\Delta\theta$
- **Angular max pooling** to remove rotation ambiguity

# Toy GCNN architecture



# Learning local descriptors with GCNN



- As similar as possible on **positives**  $\mathcal{T}^+$
- As dissimilar as possible on **negatives**  $\mathcal{T}^-$
- Minimize **siamese loss** w.r.t. GCNN parameters  $\Theta$

$$\begin{aligned}\ell(\Theta) &= (1 - \gamma) \sum_{(x, x^+) \in \mathcal{T}^+} \|\mathbf{f}_\Theta(x) - \mathbf{f}_\Theta(x^+)\| \\ &+ \gamma \sum_{(x, x^-) \in \mathcal{T}^-} \max\{\mu - \|\mathbf{f}_\Theta(x) - \mathbf{f}_\Theta(x^-)\|, 0\}\end{aligned}$$

## Descriptor robustness



HKS descriptor distance

Masci, Boscaini, Bronstein, Vandergheynst 2015; data: Bronstein et al. 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)

## Descriptor robustness



WKS descriptor distance

Masci, Boscaini, Bronstein, Vandergheynst 2015; data: Bronstein et al. 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)

## Descriptor robustness



Optimal Spectral descriptor distance

Masci, Boscaini, Bronstein, Vandergheynst 2015; data: Bronstein et al. 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)

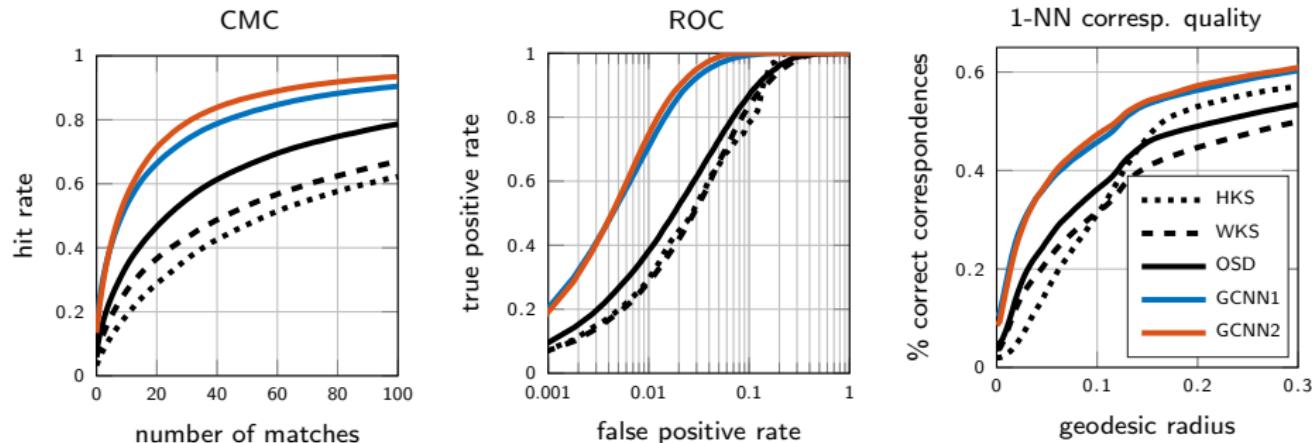
## Descriptor robustness



GCNN descriptor distance

Masci, Boscaini, Bronstein, Vandergheynst 2015; data: Bronstein et al. 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)

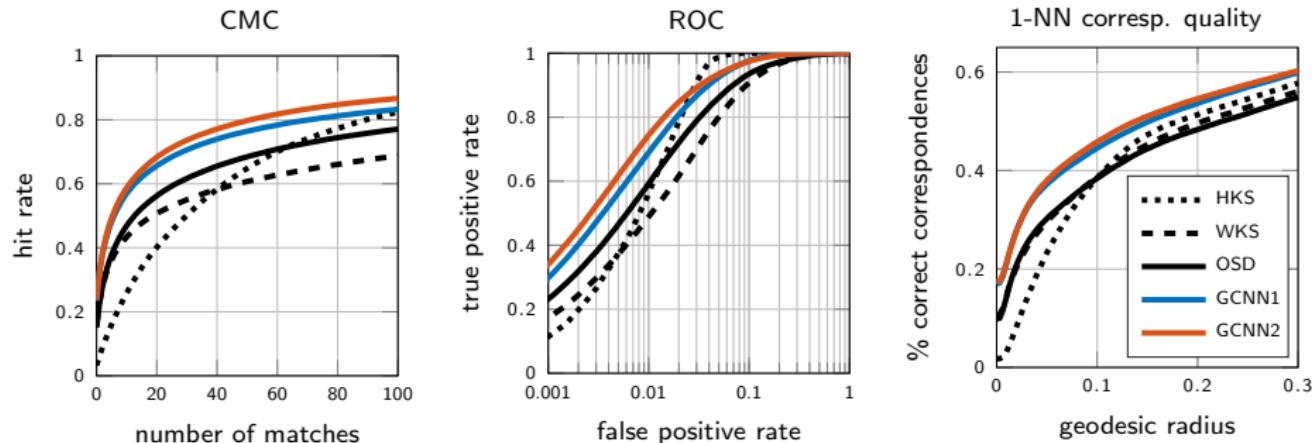
# Descriptor performance



Descriptor performance using symmetric Princeton benchmark  
(training and testing: FAUST)

Methods: Sun et al. 2009 (HKS); Aubry et al. 2011 (WKS); Litman, Bronstein 2014 (OSD); Masci, Boscaini, Bronstein, Vandergheynst 2015 (GCNN); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

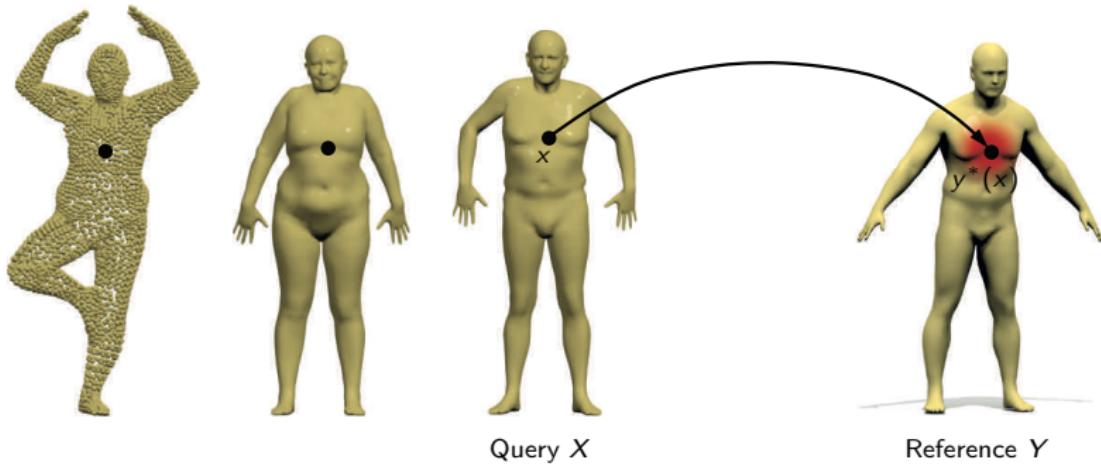
# Descriptor performance



Descriptor performance using symmetric Princeton benchmark  
(training: FAUST, testing: TOSCA)

Methods: Sun et al. 2009 (HKS); Aubry et al. 2011 (WKS); Litman, Bronstein 2014 (OSD); Masci, Boscaini, Bronstein, Vandergheynst 2015 (GCNN); data: Bogo et al. 2014 (FAUST); Bronstein et al. 2008 (TOSCA); benchmark: Kim et al. 2011

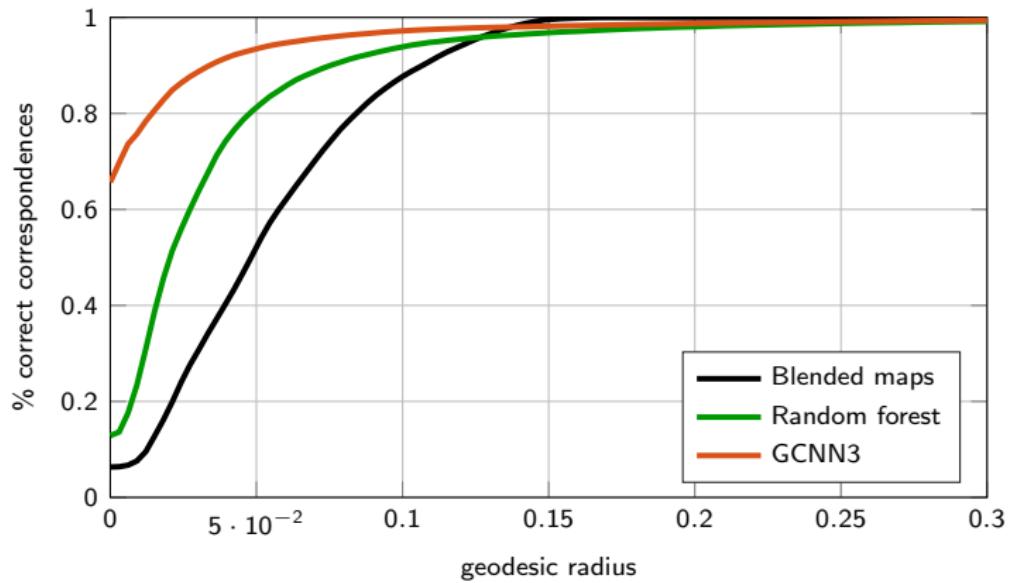
# Learning shape correspondence with GCNN



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

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

## GCNN correspondence performance



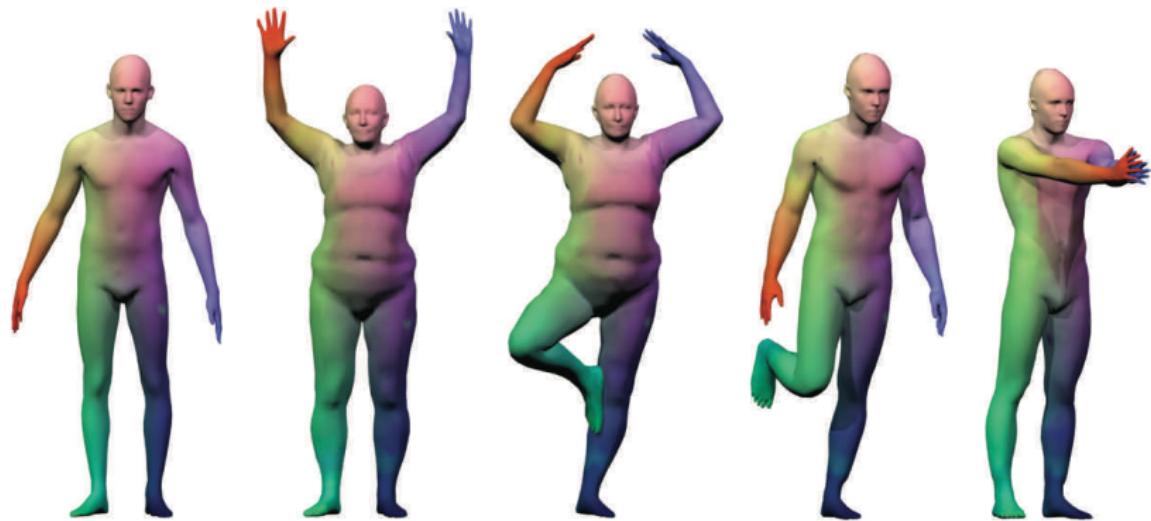
Correspondence evaluated using symmetric Princeton benchmark  
(training and testing: FAUST)

## Correspondence examples: Random forest



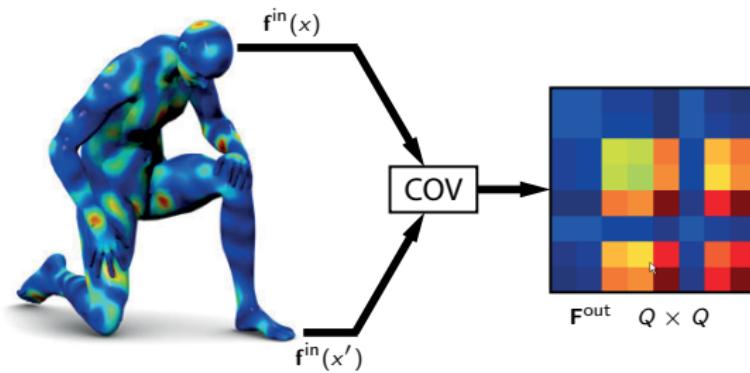
Correspondence found using random forest  
(similar colors encode corresponding points)

## Correspondence examples: GCNN



Correspondence found using ShapeNet  
(similar colors encode corresponding points)

## From local to global features: covariance layer



$$\begin{aligned} F^{out} &= \int_X (\mathbf{f}^{in}(x) - \mu_{\mathbf{f}^{in}})(\mathbf{f}^{in}(x) - \mu_{\mathbf{f}^{in}})^{\top} dx \\ \mu_{\mathbf{f}^{in}} &= \int_X \mathbf{f}_{in}(x) dx \end{aligned}$$

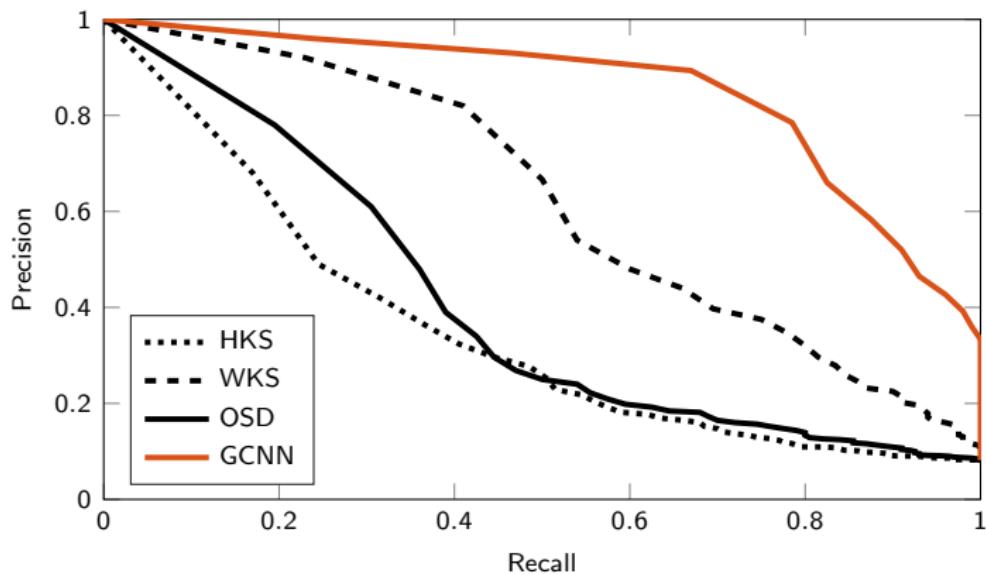
- Aggregates local features into a **global shape descriptor**

## Learning shape similarity with GCNN



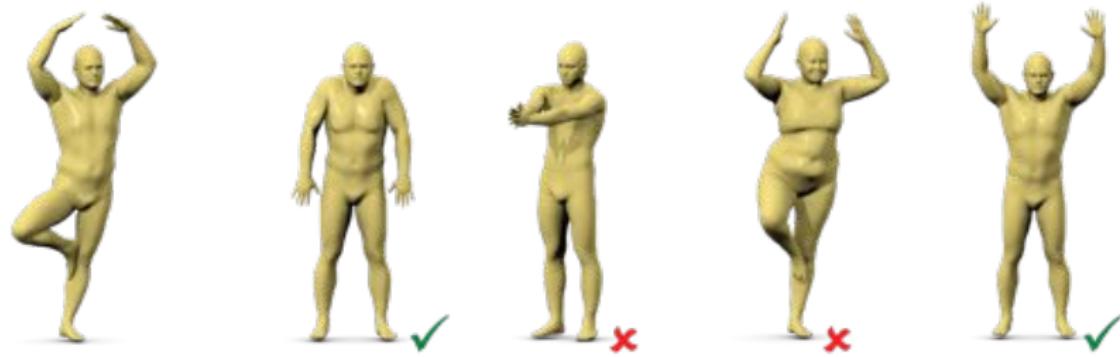
- Global shape descriptor using covariance layer in GCNN  $F_\Theta$
- As similar as possible on **positives**  $\mathcal{T}^+$
- As dissimilar as possible on **negatives**  $\mathcal{T}^-$
- Minimize **siamese loss** w.r.t. GCNN parameters  $\Theta$

## GCNN retrieval performance



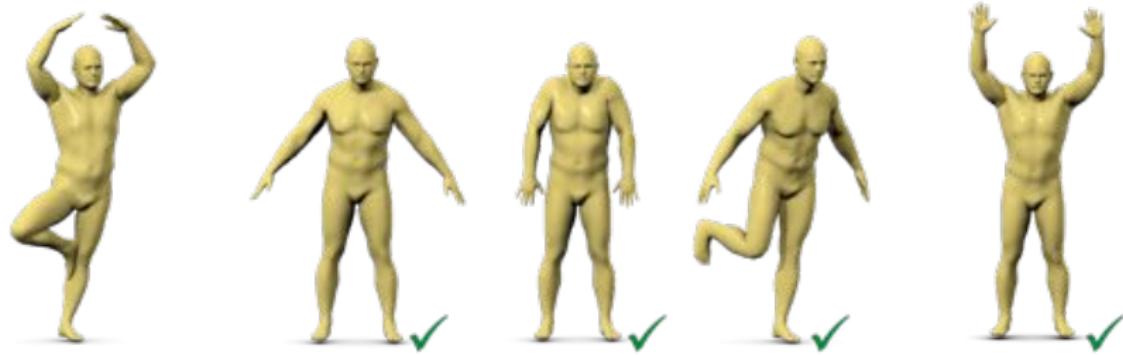
1-layer GCNN; Training and testing: FAUST

## Retrieval examples: HKS



Shape retrieval using similarity computed with HKS

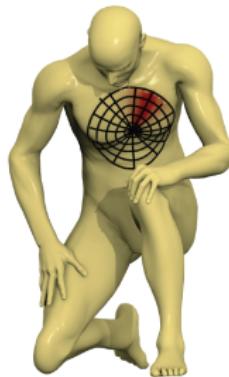
## Retrieval examples: GCNN



Shape retrieval using similarity computed with GCNN

# Outline

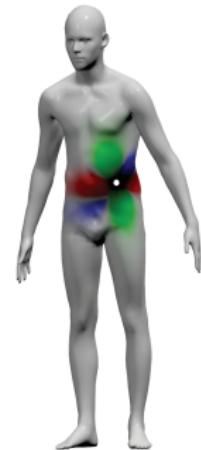
- Background: definitions of manifold and Laplace-Beltrami operator
- Spectral descriptors
- **Intrinsic convolutional neural networks**



Geodesic  
convolution



Windowed  
Fourier Transform



Anisotropic  
Diffusion

## Convolution (Euclidean spaces)

Given two functions  $f, g: [-\pi, \pi] \rightarrow \mathbb{R}$  their **convolution** is a function

$$(f * g)(x) = \int_{-\pi}^{\pi} f(\xi)g(x - \xi)d\xi$$

## Convolution (Euclidean spaces)

Given two functions  $f, g: [-\pi, \pi] \rightarrow \mathbb{R}$  their **convolution** is a function

$$(f * g)(x) = \int_{-\pi}^{\pi} f(\xi)g(x - \xi)d\xi$$

**Convolution Theorem:** Fourier transform diagonalizes the convolution operator

## Convolution (Euclidean spaces)

Given two functions  $f, g: [-\pi, \pi] \rightarrow \mathbb{R}$  their **convolution** is a function

$$(f * g)(x) = \int_{-\pi}^{\pi} f(\xi)g(x - \xi)d\xi$$

**Convolution Theorem:** Fourier transform diagonalizes the convolution operator  $\Rightarrow$  convolution can be computed in the Fourier domain as

$$f * g = \mathcal{F}^{-1}(\mathcal{F}f \cdot \mathcal{F}g)$$

## Convolution (non-Euclidean spaces)

Generalized convolution of  $f, g \in L^2(X)$  can be defined by analogy

$$(f * g)(x) = \sum_{k \geq 1} \langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)} \phi_k(x)$$

## Convolution (non-Euclidean spaces)

Generalized convolution of  $f, g \in L^2(X)$  can be defined by analogy

$$(f * g)(x) = \sum_{k \geq 1} \underbrace{\langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)}}_{\text{product in the Fourier domain}} \phi_k(x)$$

## Convolution (non-Euclidean spaces)

Generalized convolution of  $f, g \in L^2(X)$  can be defined by analogy

$$(f * g)(x) = \underbrace{\sum_{k \geq 1} \underbrace{\langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)}}_{\text{product in the Fourier domain}} \phi_k(x)}_{\text{inverse Fourier transform}}$$

## Convolution (non-Euclidean spaces)

Generalized convolution of  $f, g \in L^2(X)$  can be defined by analogy

$$(f * g)(x) = \underbrace{\sum_{k \geq 1} \underbrace{\langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)}}_{\text{product in the Fourier domain}} \phi_k(x)}_{\text{inverse Fourier transform}}$$

- Not shift-invariant!

## Convolution (non-Euclidean spaces)

Generalized convolution of  $f, g \in L^2(X)$  can be defined by analogy

$$(f * g)(x) = \underbrace{\sum_{k \geq 1} \underbrace{\langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)}}_{\text{product in the Fourier domain}} \phi_k(x)}_{\text{inverse Fourier transform}}$$

- Not shift-invariant!
- Spectral CNN: learn filter in the Fourier domain

## Convolution (non-Euclidean spaces)

Generalized convolution of  $f, g \in L^2(X)$  can be defined by analogy

$$(f * g)(x) = \underbrace{\sum_{k \geq 1} \underbrace{\langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)}}_{\text{product in the Fourier domain}} \phi_k(x)}_{\text{inverse Fourier transform}}$$

- Not shift-invariant!
- Spectral CNN: learn filter in the Fourier domain
- Problem: Filter coefficients depend on basis  $\{\phi_k\}_{k \geq 1}$

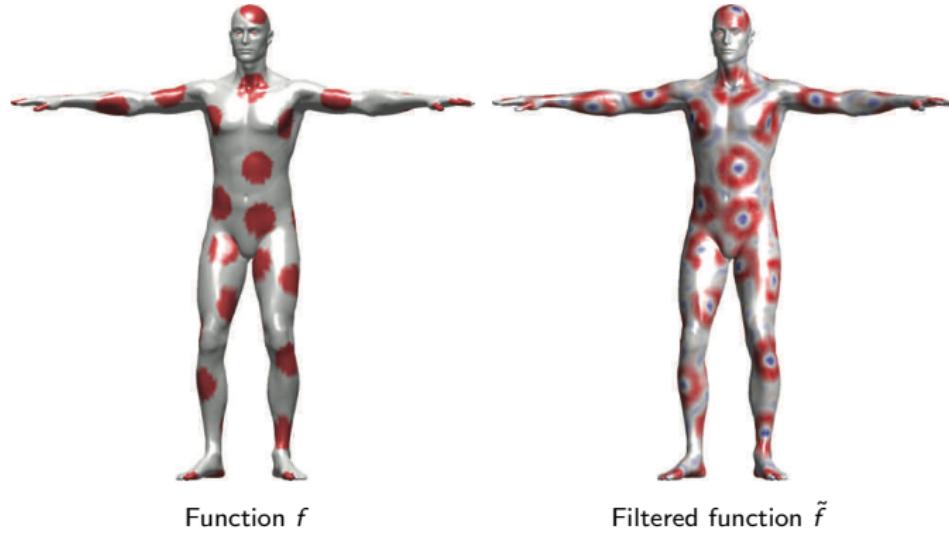
## Convolution (non-Euclidean spaces)

Generalized convolution of  $f, g \in L^2(X)$  can be defined by analogy

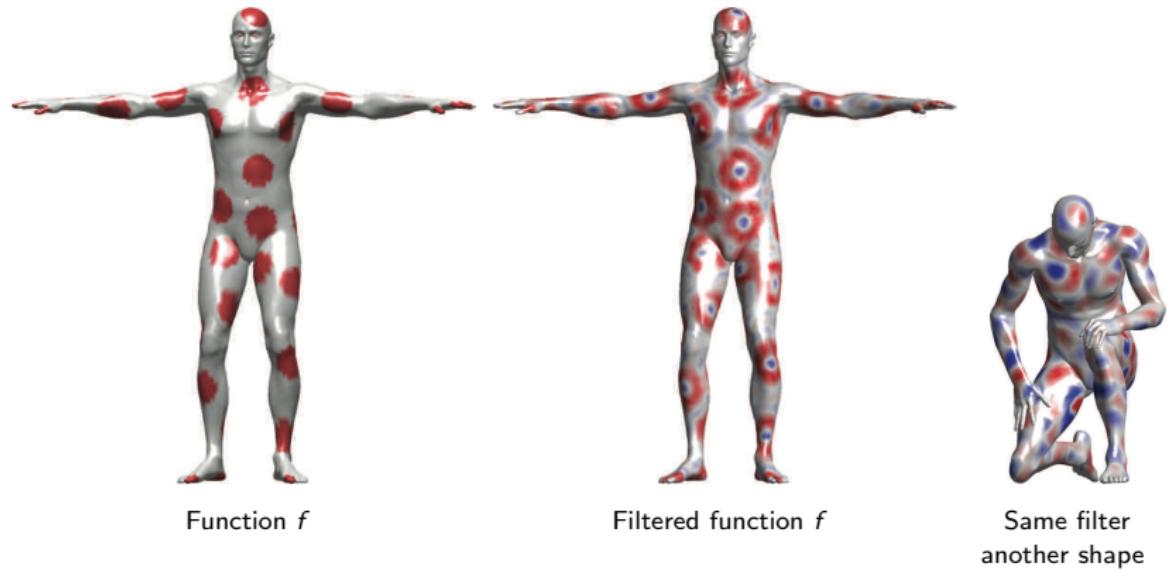
$$(f * g)(x) = \underbrace{\sum_{k \geq 1} \underbrace{\langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)}}_{\text{product in the Fourier domain}} \phi_k(x)}_{\text{inverse Fourier transform}}$$

- Not shift-invariant!
- Spectral CNN: learn filter in the Fourier domain
- Problem: Filter coefficients depend on basis  $\{\phi_k\}_{k \geq 1}$   
⇒ does not generalize to other domains!

## Convolution (non-Euclidean spaces)

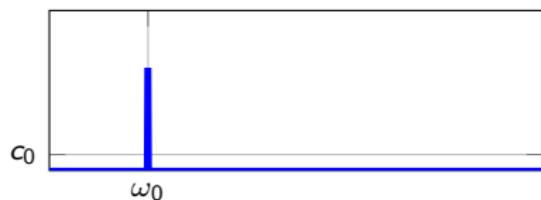
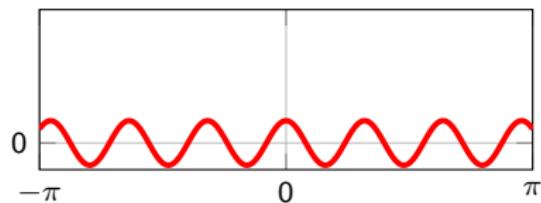


## Convolution (non-Euclidean spaces)



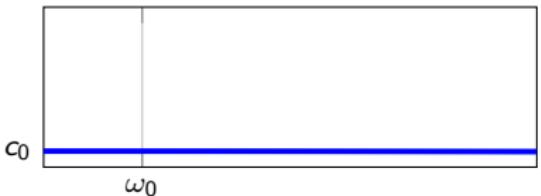
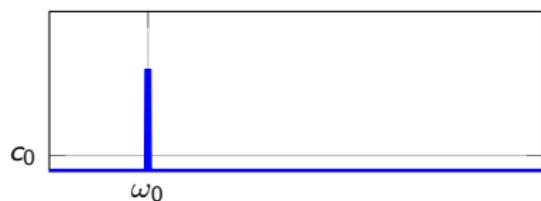
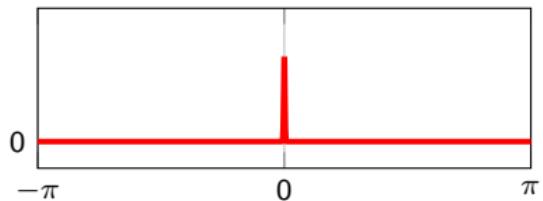
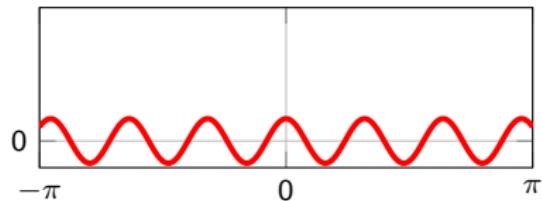
## Uncertainty principle

Spatial localization  $\times$  frequency localization = const



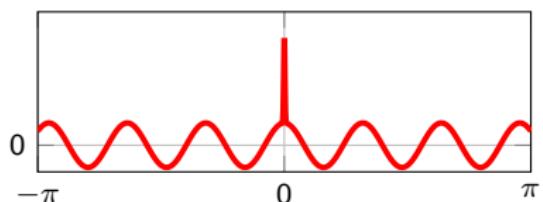
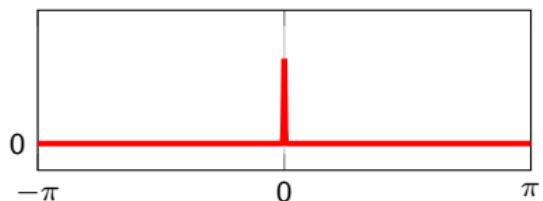
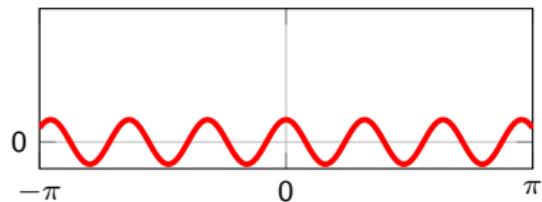
# Uncertainty principle

Spatial localization  $\times$  frequency localization = const

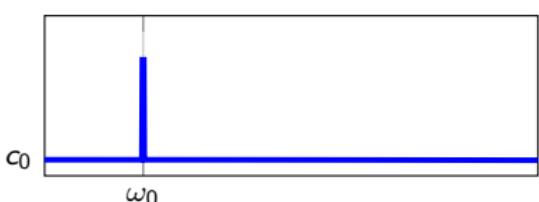
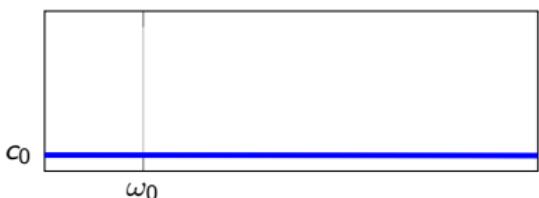
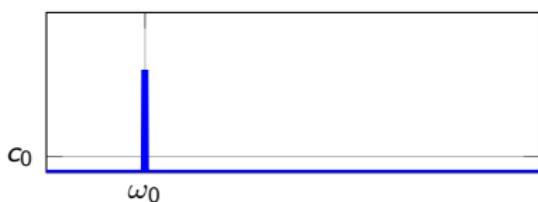


# Uncertainty principle

Spatial localization  $\times$  frequency localization = const



spatial



frequency

## Windowed Fourier Transform (WFT)

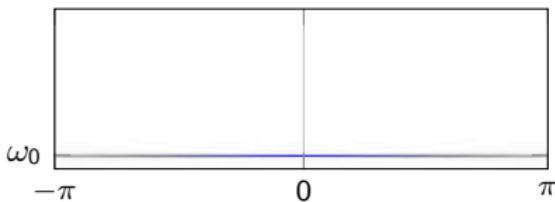
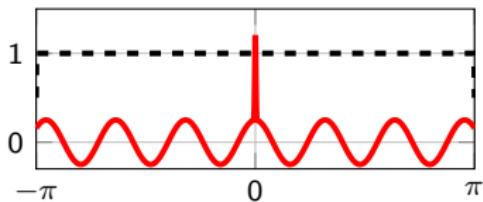
Localize Fourier transform in a window

$$WFT(f(x))(\xi, \omega) = \int_{-\pi}^{\pi} f(x)w(x - \xi)e^{-i\omega x}dx$$

# Windowed Fourier Transform (WFT)

Localize Fourier transform in a window

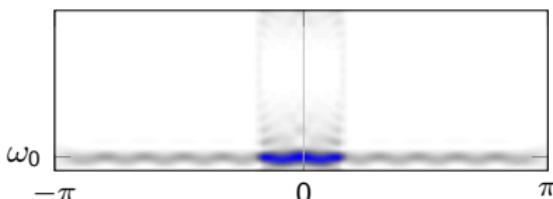
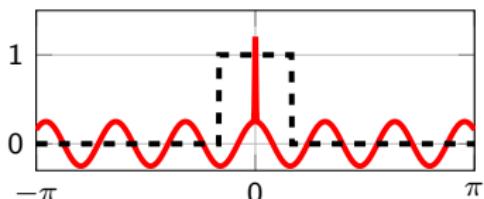
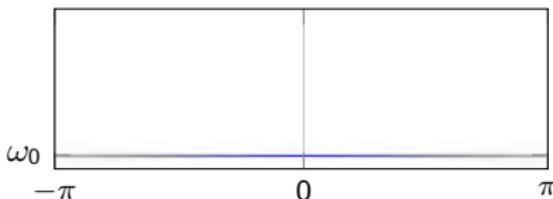
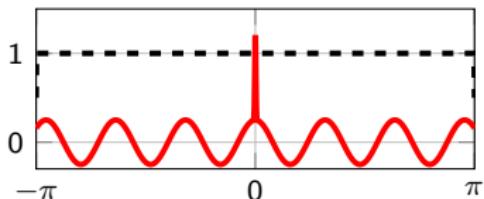
$$WFT(f(x))(\xi, \omega) = \int_{-\pi}^{\pi} f(x)w(x - \xi)e^{-i\omega x} dx$$



# Windowed Fourier Transform (WFT)

Localize Fourier transform in a window

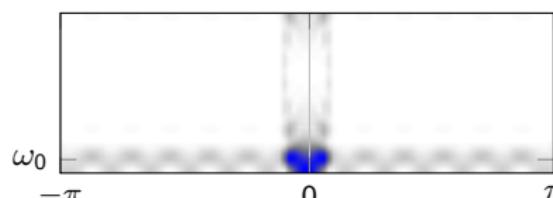
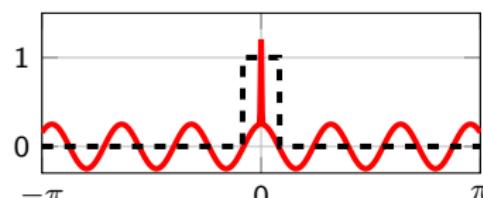
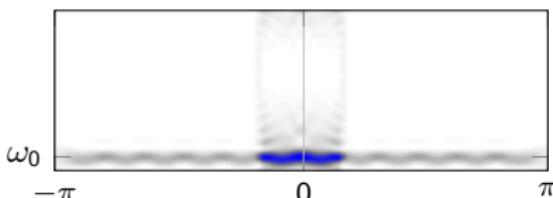
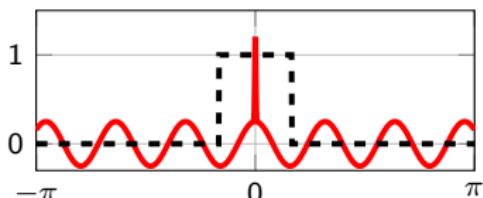
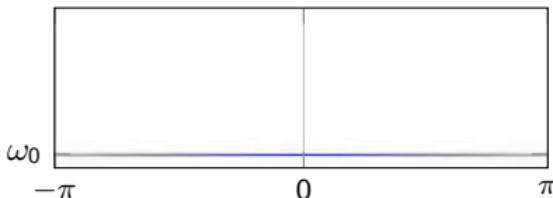
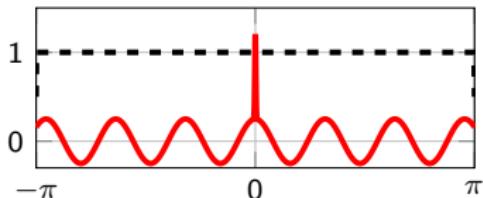
$$WFT(f(x))(\xi, \omega) = \int_{-\pi}^{\pi} f(x)w(x - \xi)e^{-i\omega x} dx$$



# Windowed Fourier Transform (WFT)

Localize Fourier transform in a window

$$WFT(f(x))(\xi, \omega) = \int_{-\pi}^{\pi} f(x)w(x - \xi)e^{-i\omega x} dx$$



# Windowed Fourier Transform (WFT)

Localize Fourier transform in a window

$$WFT(f(x))(\xi, \omega) = \int_{-\pi}^{\pi} f(x) \underbrace{w(x - \xi) e^{-i\omega x}}_{\text{atom } g_{\xi, \omega}(x)} dx$$

# Windowed Fourier Transform (WFT)

Localize Fourier transform in a window

$$WFT(f(x))(\xi, \omega) = \underbrace{\int_{-\pi}^{\pi} f(x) \underbrace{w(x - \xi)}_{\text{atom } g_{\xi, \omega}(x)} e^{-i\omega x} dx}_{\langle f(x), g_{\xi, \omega}(x) \rangle_{L^2(X)}}$$

# Windowed Fourier transform on manifolds

**Translation:** convolution with delta

$$(T_{x'} f)(x) = (f \star \delta_{x'})(x)$$

# Windowed Fourier transform on manifolds

**Translation:** convolution with delta

$$\begin{aligned}(T_{x'} f)(x) &= (f * \delta_{x'})(x) \\ &= \sum_{k \geq 1} \langle f, \phi_k \rangle_{L^2(X)} \langle \delta_{x'}, \phi_k \rangle_{L^2(X)} \phi_k(x)\end{aligned}$$

# Windowed Fourier transform on manifolds

**Translation:** convolution with delta

$$\begin{aligned}(T_{x'} f)(x) &= (f \star \delta_{x'})(x) \\&= \sum_{k \geq 1} \langle f, \phi_k \rangle_{L^2(X)} \langle \delta_{x'}, \phi_k \rangle_{L^2(X)} \phi_k(x) \\&= \sum_{k \geq 1} \hat{f}_k \phi_k(x') \phi_k(x)\end{aligned}$$

# Windowed Fourier transform on manifolds

**Translation:** convolution with delta

$$\begin{aligned}(T_{x'} f)(x) &= (f \star \delta_{x'})(x) \\&= \sum_{k \geq 1} \langle f, \phi_k \rangle_{L^2(X)} \langle \delta_{x'}, \phi_k \rangle_{L^2(X)} \phi_k(x) \\&= \sum_{k \geq 1} \hat{f}_k \phi_k(x') \phi_k(x)\end{aligned}$$

**Modulation:** multiplication by basis function

$$(M_k f)(x) = f(x) \phi_k(x)$$

# Windowed Fourier transform on manifolds

**Translation:** convolution with delta

$$\begin{aligned}(T_{x'} f)(x) &= (f * \delta_{x'})(x) \\&= \sum_{k \geq 1} \langle f, \phi_k \rangle_{L^2(X)} \langle \delta_{x'}, \phi_k \rangle_{L^2(X)} \phi_k(x) \\&= \sum_{k \geq 1} \hat{f}_k \phi_k(x') \phi_k(x)\end{aligned}$$

**Modulation:** multiplication by basis function

$$(M_k f)(x) = f(x) \phi_k(x)$$

**Windowed Fourier transform (WFT):**

$$(Sf)_{x,k} = \langle f, \underbrace{M_k T_x g}_{\text{atom}} \rangle_{L^2(X)}$$

# Windowed Fourier transform on manifolds

**Translation:** convolution with delta

$$\begin{aligned}(T_{x'} f)(x) &= (f * \delta_{x'})(x) \\&= \sum_{k \geq 1} \langle f, \phi_k \rangle_{L^2(X)} \langle \delta_{x'}, \phi_k \rangle_{L^2(X)} \phi_k(x) \\&= \sum_{k \geq 1} \hat{f}_k \phi_k(x') \phi_k(x)\end{aligned}$$

**Modulation:** multiplication by basis function

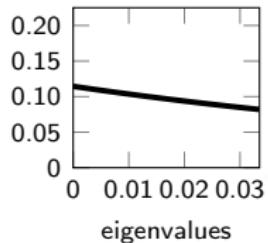
$$(M_k f)(x) = f(x) \phi_k(x)$$

**Windowed Fourier transform (WFT):**

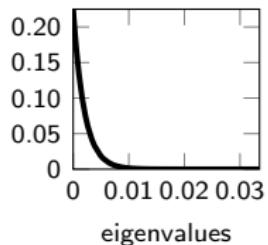
$$(Sf)_{x,k} = \langle f, \underbrace{M_k T_x g}_{\text{atom}} \rangle_{L^2(X)} = \sum_{\ell \geq 1} \hat{g}_\ell \phi_\ell(x) \langle f, \phi_\ell \phi_k \rangle_{L^2(X)}$$

# Windowed Fourier transform on manifolds

$\hat{g}_1$



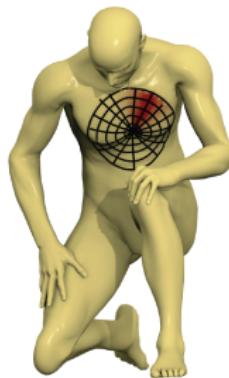
$\hat{g}_2$



Examples of WFT atoms for different windows  $\hat{g}$ .

# Outline

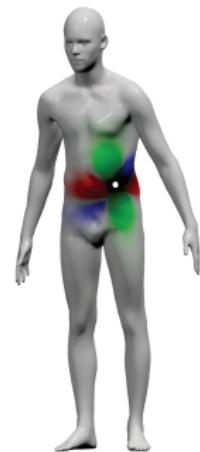
- Background: definitions of manifold and Laplace-Beltrami operator
- Spectral descriptors
- **Intrinsic convolutional neural networks**



Geodesic  
convolution



Windowed  
Fourier Transform



Anisotropic  
Diffusion

$$\frac{\partial}{\partial t} f(x) = -\operatorname{div}(c \nabla f(x))$$

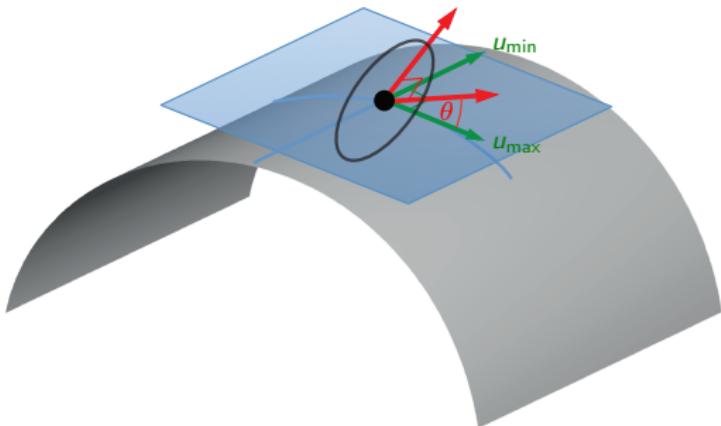
$c$  = **thermal diffusivity constant** describing heat conduction properties of the material (diffusion speed is equal everywhere)

## Anisotropic diffusion

$$\frac{\partial}{\partial t} f(x) = -\operatorname{div}(A(x)\nabla f(x))$$

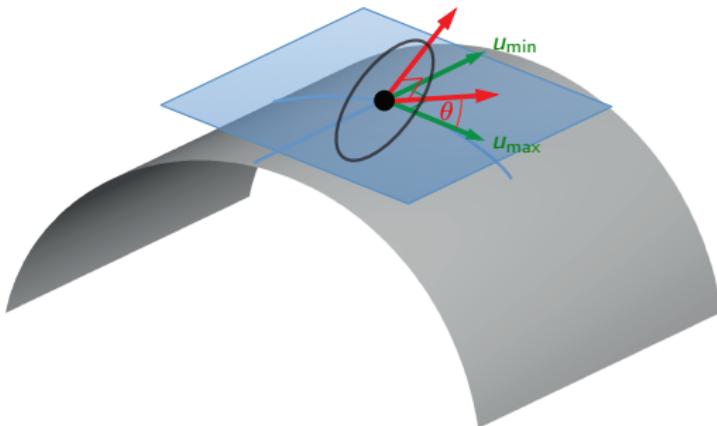
$A(x)$  = **heat conductivity tensor** describing heat conduction properties of the material (diffusion speed is position + direction dependent)

# Anisotropic diffusion on manifolds



$$\frac{\partial}{\partial t} f(x) = -\operatorname{div}_X \left( R_\theta \begin{pmatrix} \alpha & \\ & 1 \end{pmatrix} R_\theta^\top \nabla_X f(x) \right)$$

# Anisotropic diffusion on manifolds



$$\frac{\partial}{\partial t} f(x) = -\operatorname{div}_X \left( \underbrace{R_\theta \begin{pmatrix} \alpha \\ 1 \end{pmatrix} R_\theta^\top}_{D_{\alpha\theta}(x)} \nabla_X f(x) \right)$$

- **Anisotropic Laplacian**  $\Delta_{\alpha\theta} f(x) = \operatorname{div}_X (D_{\alpha\theta}(x) \nabla_X f(x))$
- $\theta$  = orientation w.r.t. max curvature direction
- $\alpha$  = 'elongation'

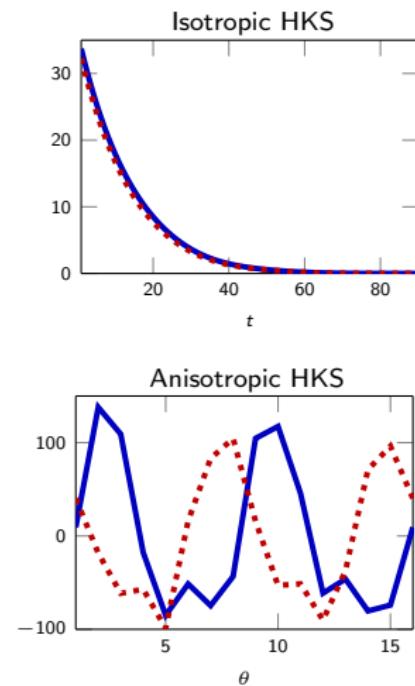
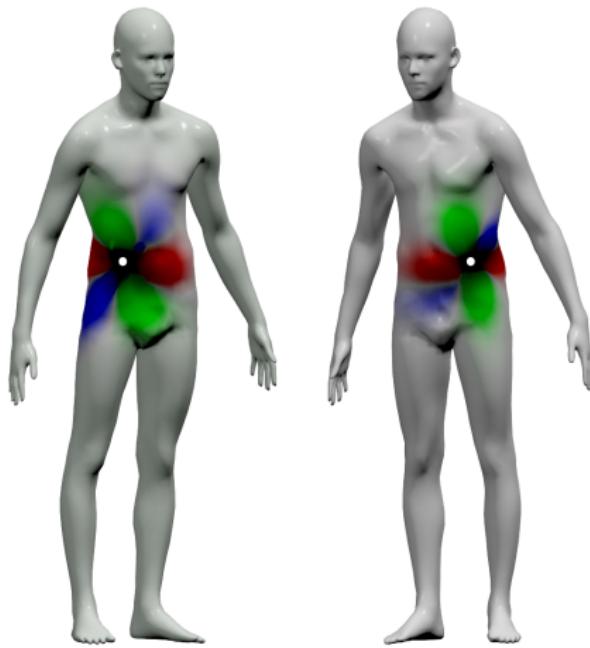
## Anisotropic heat kernels

$$h_{\alpha\theta t}(x, \xi) = \sum_{k \geq 1} e^{-t\lambda_{\alpha\theta k}} \phi_{\alpha\theta k}(x)\phi_{\alpha\theta k}(\xi)$$



Examples of anisotropic heat kernels  $h_{\alpha\theta t}$  for different values of  $t$ ,  $\theta$  and  $\alpha$

# Isotropic vs Anisotropic HKS

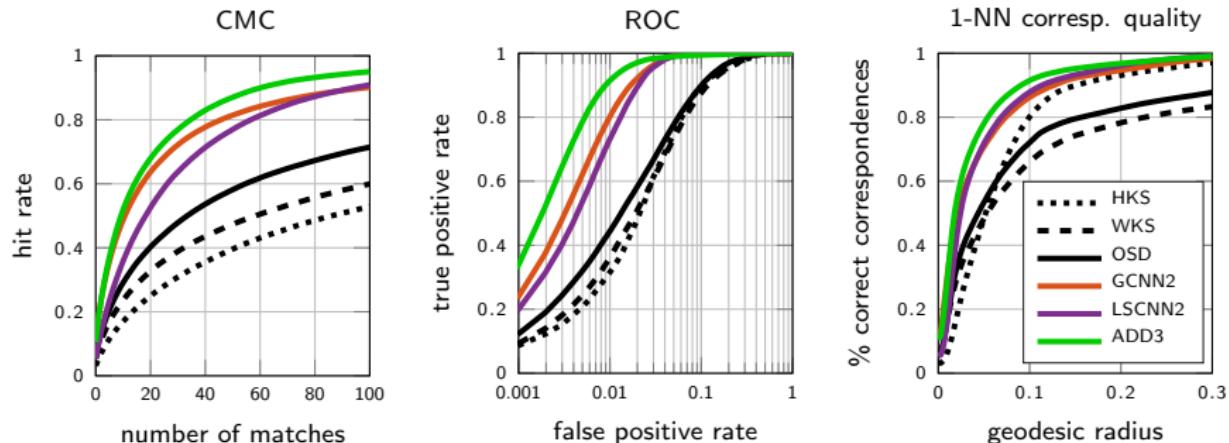


# Anisotropic Diffusion Descriptor (ADD) robustness



Anisotropic Diffusion Descriptor (ADD) distance

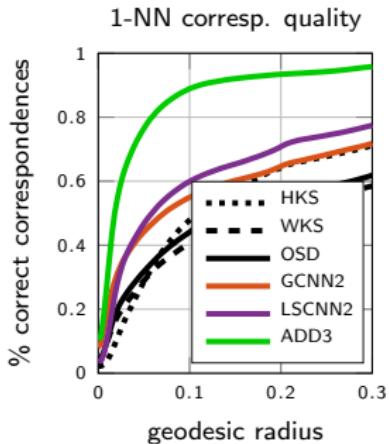
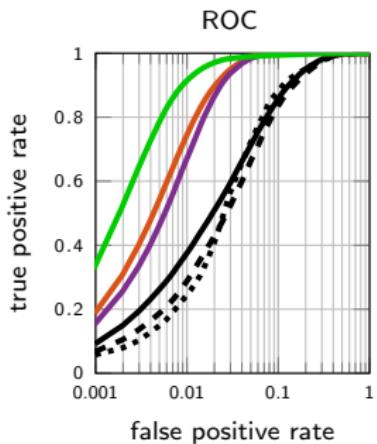
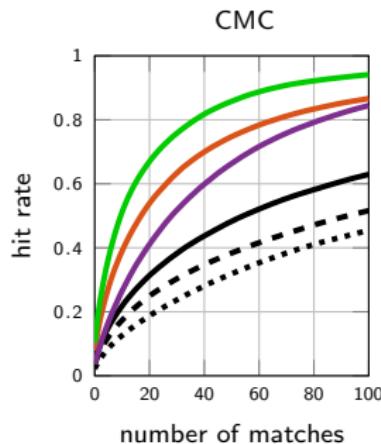
# Anisotropic Diffusion Descriptor (ADD) performance



Descriptor performance using symmetric Princeton benchmark  
(training and testing: FAUST)

Methods: Sun et al. 2009 (HKS); Aubry et al. 2011 (WKS); Litman, Bronstein 2014 (OSD); Masci, Boscaini, Bronstein, Vandergheynst 2015 (GCNN); Boscaini, Masci, Melzi, Bronstein, Castellani, Vandergheynst 2015 (LSCNN); Boscaini, Masci, Rodolà, Bronstein, Cremers 2015 (ADD); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

# Anisotropic Diffusion Descriptor (ADD) performance



Descriptor performance using asymmetric Princeton benchmark  
(training and testing: FAUST)

Methods: Sun et al. 2009 (HKS); Aubry et al. 2011 (WKS); Litman, Bronstein 2014 (OSD); Masci, Boscaini, Bronstein, Vandergheynst 2015 (GCNN); Boscaini, Masci, Melzi, Bronstein, Castellani, Vandergheynst 2015 (LSCNN); Boscaini, Masci, Rodolà, Bronstein, Cremers 2015 (ADD); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

## ADD correspondence example: meshes



Correspondence found using ADD  
(similar colors encode corresponding points)

## ADD correspondence example: point clouds



Correspondence found using ADD  
(similar colors encode corresponding points)

## Summary

- First construction of generalizable intrinsic convolutional neural networks
- Learnable, task-specific, intrinsic features
- State-of-the-art performance in a variety of applications in 3D shape analysis
- Beyond shapes: graphs, social networks, etc.



J. Masci   M. Bronstein



P. Vandergheynst



S. Melzi   U. Castellani



E. Rodolà   D. Cremers



Thank you!