

# Geometric deep learning

Michael Bronstein



University of Lugano



Intel Corporation

IPAM, UCLA 9 February 2016



D. Boscaini



J. Masci



P. Vandergheynst



E. Rodolà



D. Cremers



ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE



Funded by



# Deluge of geometric data



**KINECT**  
for XBOX 360



SoftKinetic

**intel** REALSENSE

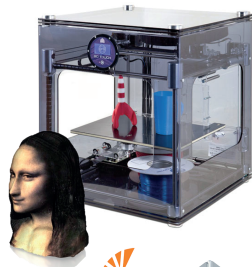
3D sensors



Google 3D warehouse

**shapeways**

Repositories

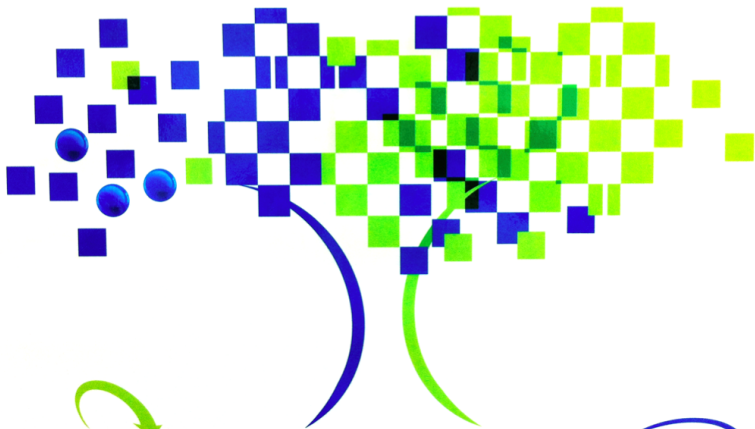


**Stratasys** 3D SYSTEMS

3D printers



intel REALSENSE™  
TECHNOLOGY



**INVISION**

*A New Dimension to*

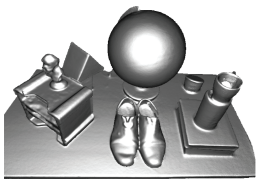


(Acquired by Intel in 2012)

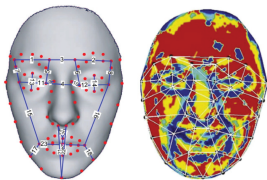


Different form factor computers featuring Intel RealSense 3D camera

# Applications



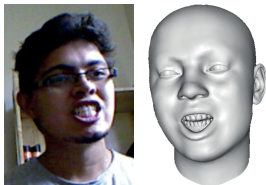
Reconstruction



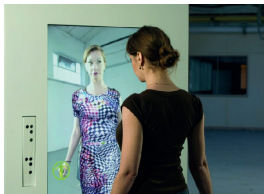
Recognition



Retrieval



Avatars



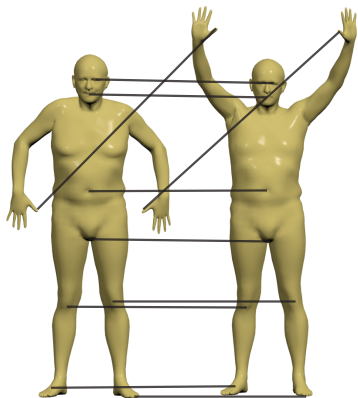
Virtual dressing



Gesture control

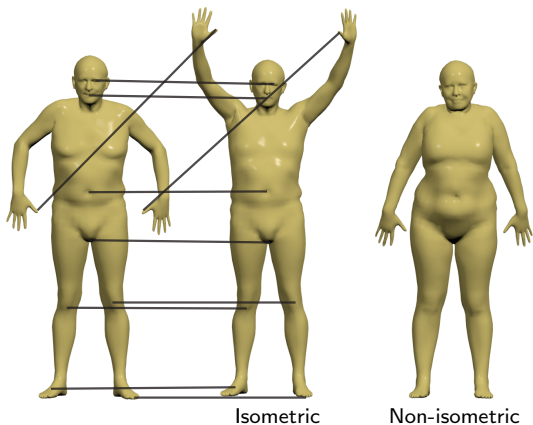
Images: Davison et al. 2011; Zaferiou et al. 2012; Kim et al. 2013; Faceshift; Fashion3D; Minority report

# Basic problems: shape similarity and correspondence

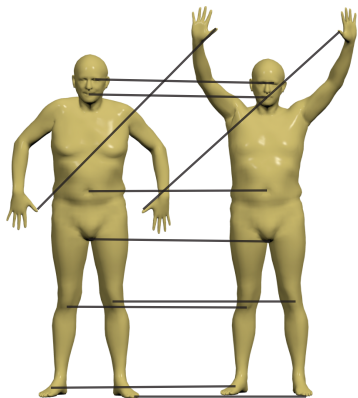


Isometric

# Basic problems: shape similarity and correspondence



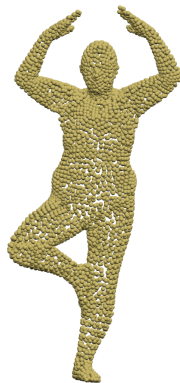
# Basic problems: shape similarity and correspondence



Isometric

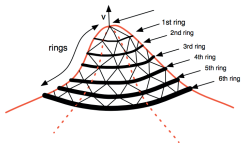
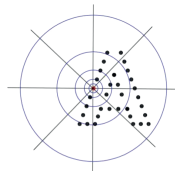
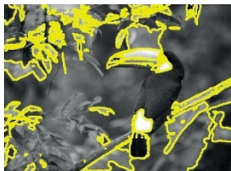


Non-isometric



Different  
representation

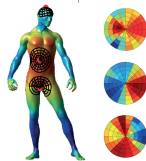
# 3D feature descriptors



SIFT<sup>1</sup> / MeshHOG<sup>2</sup>



MSER<sup>3</sup> / ShapeMSER<sup>4</sup>



(Intrinsic<sup>6</sup>) Shape context<sup>5</sup>



Spin image<sup>7</sup>



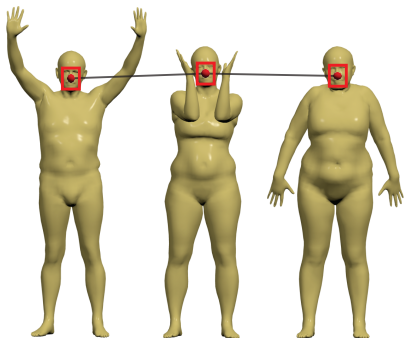
Heat kernel signature<sup>8</sup>

<sup>1</sup>Lowe 2004; <sup>2</sup>Zaharescu et al. 2009; <sup>3</sup>Matas et al. 2002; <sup>4</sup>Litman et al. 2010;

<sup>5</sup>Belongie et al. 2000; <sup>6</sup>Kokkinos et al. 2012; <sup>7</sup>Johnson et al. 1999; <sup>8</sup>Sun et al. 2009

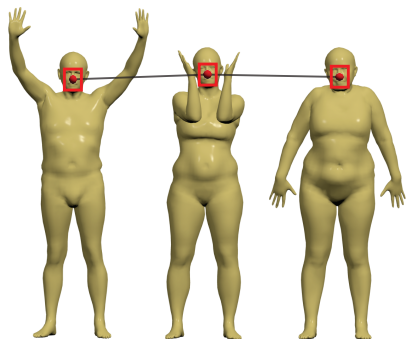
# Task-specific features

## Correspondence

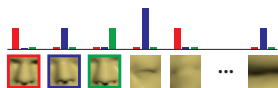
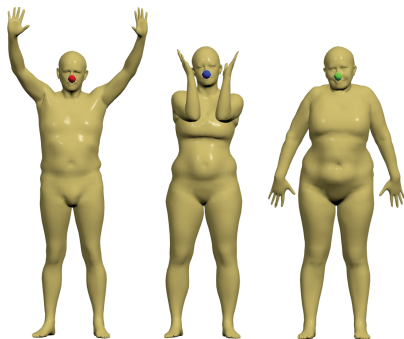


# Task-specific features

**Correspondence**



**Similarity**



# 2012

---

## ImageNet Classification with Deep Convolutional Neural Networks

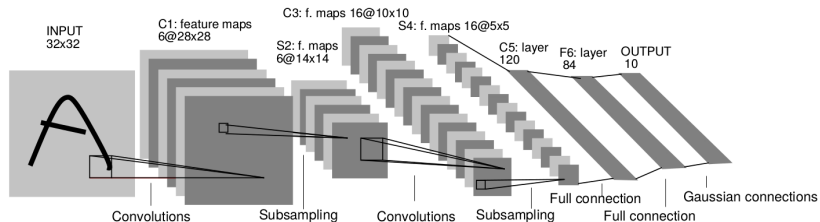
---

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

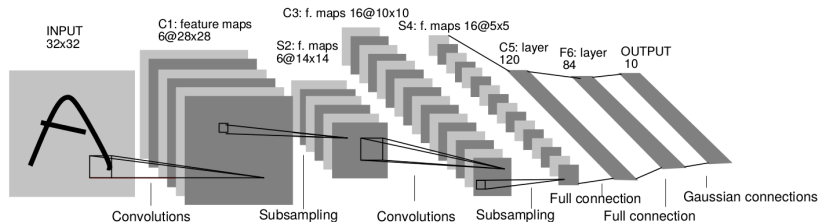
# Convolutional neural networks



Typical CNN architecture

- Combination of convolution and pooling layers

# Convolutional neural networks

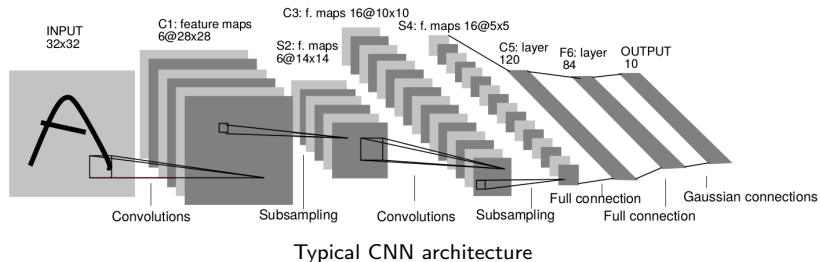


Typical CNN architecture

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

Fukushima 1980; LeCun et al. 1989; Image: H. Wang

# Convolutional neural networks

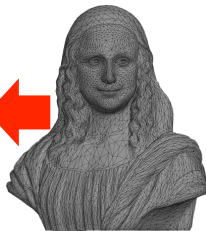
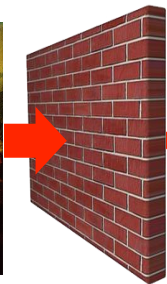


- Combination of convolution and pooling layers
- Learn hierarchical abstractions from data with little prior knowledge
- State-of-the-art performance in a wide range of applications

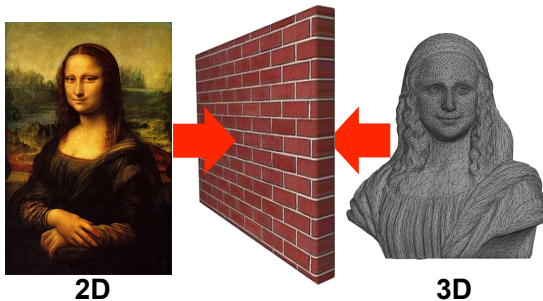
Fukushima 1980; LeCun et al. 1989; Image: H. Wang



**2D**

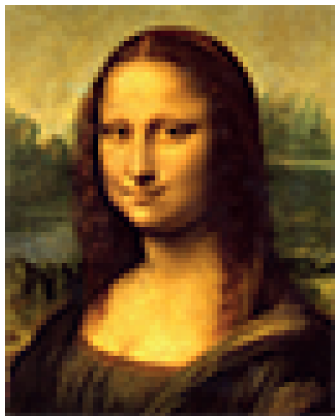


**3D**

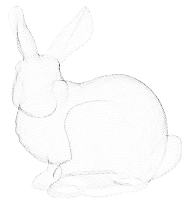


**Generalize deep learning to non-Euclidean data  
in a geometrically meaningful way**

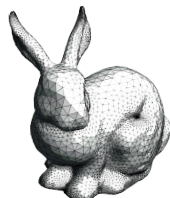
# 3D shapes vs images



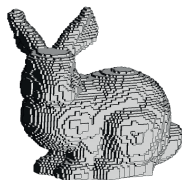
Array of pixels



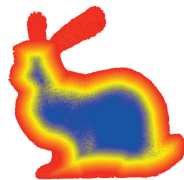
Point cloud



Mesh



Voxels

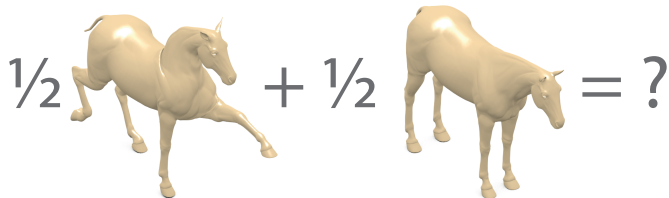


Level set

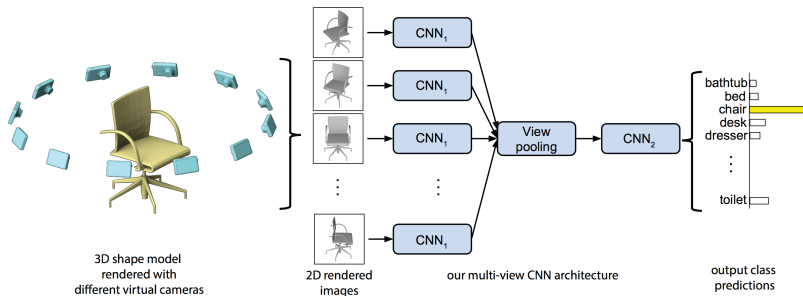
# 3D shapes vs images



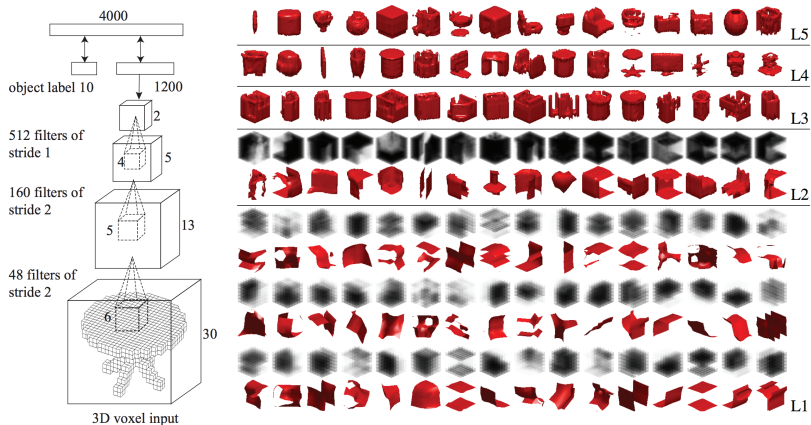
# 3D shapes vs images



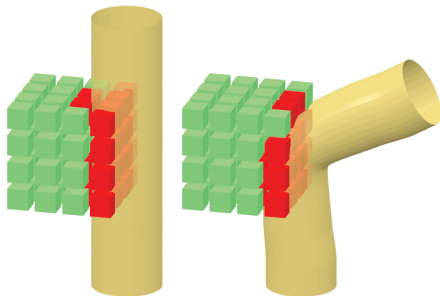
# Deep learning on 3D data



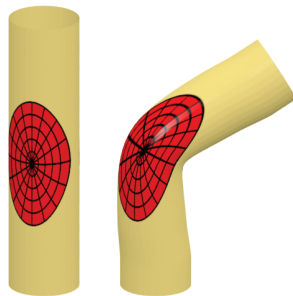
# Deep learning on 3D data



# Extrinsic vs Intrinsic



Extrinsic

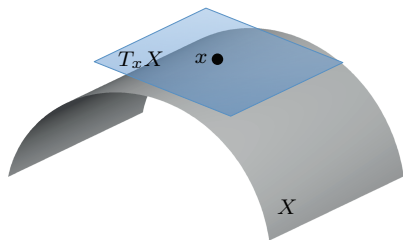


Intrinsic

- Background: Laplacians and spectral analysis on manifolds
- Spectral descriptors (heat- and wave-kernel signatures)
- Intrinsic convolutional neural networks (geodesic convolution and anisotropic diffusion)
- Optimal descriptors
- Shape correspondence
- Shape retrieval

# Riemannian geometry in one minute

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

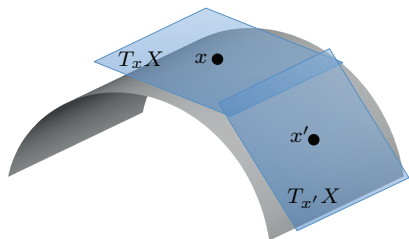


# Riemannian geometry in one minute

- **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  $x$



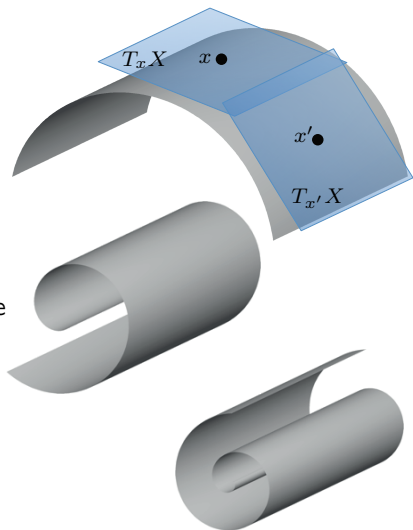
# Riemannian geometry in one minute

- **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



# Riemannian geometry in one minute

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

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

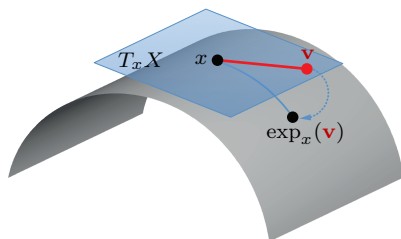
depending smoothly on  $x$

**Isometry** = metric-preserving shape deformation

- **Exponential map**

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

'unit step along geodesic'



# Riemannian geometry in one minute

- **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  $x$

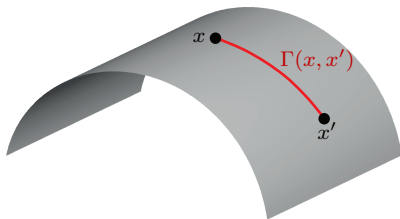
**Isometry** = metric-preserving shape deformation

- **Exponential map**

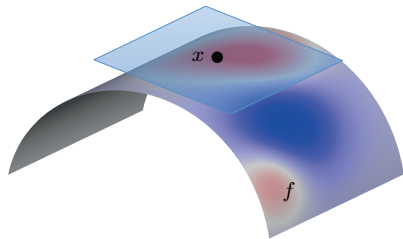
$$\exp_x : T_x X \rightarrow X$$

'unit step along geodesic'

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

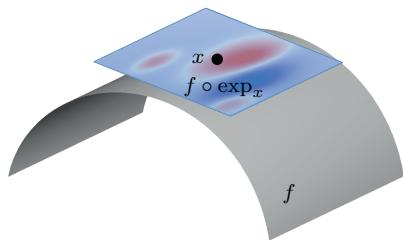


# Laplace-Beltrami operator



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

# Laplace-Beltrami operator



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

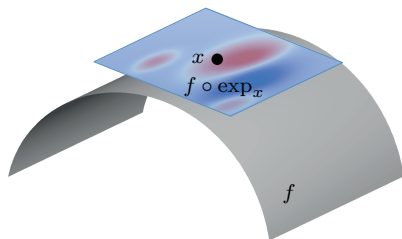
# Laplace-Beltrami operator

- Intrinsic gradient

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

Taylor expansion

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



# Laplace-Beltrami operator

- Intrinsic gradient

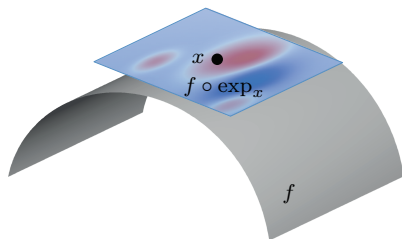
$$\nabla_X f(x) = \nabla(f \circ \exp_x)(\mathbf{0})$$

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

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



# Laplace-Beltrami operator

- Intrinsic gradient

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

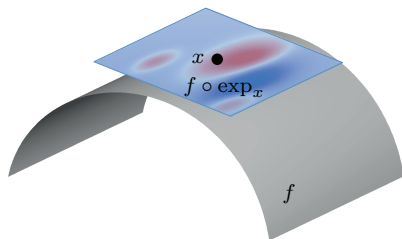
Taylor expansion

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

- Laplace-Beltrami operator

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

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



# Laplace-Beltrami operator

- Intrinsic gradient

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

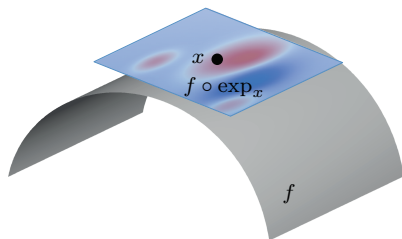
Taylor expansion

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

- Laplace-Beltrami operator

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

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



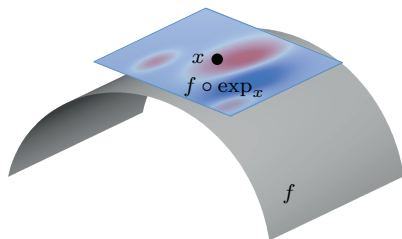
# Laplace-Beltrami operator

- 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

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

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

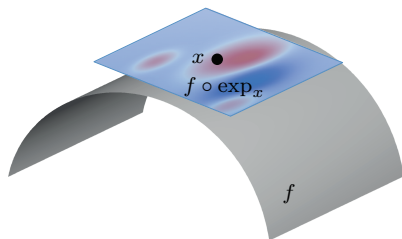
# Laplace-Beltrami operator

- 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

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

- 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

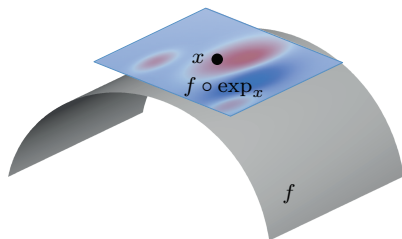
# Laplace-Beltrami operator

- 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

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

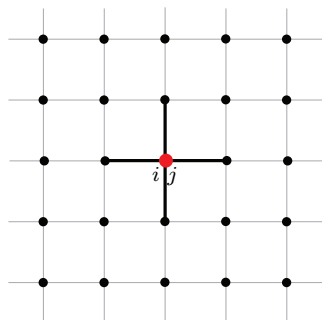
- Intrinsic (expressed solely in terms of the Riemannian metric)
- Isometry-invariant
- Self-adjoint  $\langle \Delta_X f, g \rangle_{L^2(X)} = \langle f, \Delta_X g \rangle_{L^2(X)} \Rightarrow$  orthogonal eigenfunctions
- Positive semidefinite  $\Rightarrow$  non-negative eigenvalues

# Discrete Laplacian (Euclidean)



**One-dimensional**

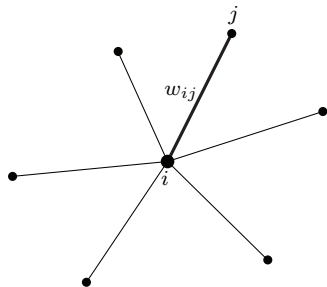
$$(\Delta f)_i \approx 2f_i - f_{i-1} - f_{i+1}$$



**Two-dimensional**

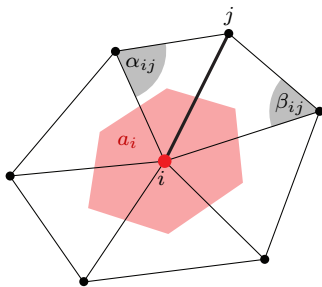
$$\begin{aligned} (\Delta f)_{ij} &\approx 4f_{ij} - f_{i-1,j} - f_{i+1,j} \\ &\quad - f_{i,j-1} - f_{i,j+1} \end{aligned}$$

# 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

$$f_t = -c\Delta f$$

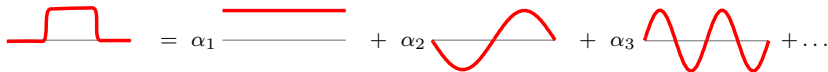
**Newton's law of cooling:** rate of change of the temperature of an object is proportional to the difference between its own temperature and the temperature of the surrounding

$c$  [m<sup>2</sup>/sec] = **thermal diffusivity constant** (assumed = 1)

# Fourier analysis (Euclidean spaces)

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

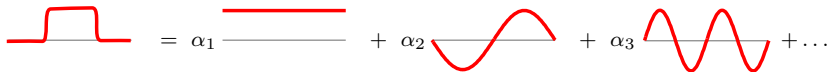
$$f(x) = \sum_{\omega} \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\xi) e^{i\omega\xi} d\xi e^{-i\omega x}$$



# Fourier analysis (Euclidean spaces)

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

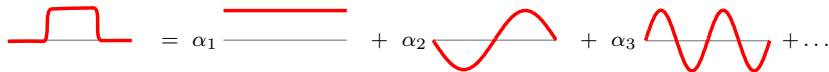
$$f(x) = \sum_{\omega} \underbrace{\frac{1}{2\pi} \int_{-\pi}^{\pi} f(\xi) e^{i\omega\xi} d\xi}_{\hat{f}(\omega) = \langle f, e^{-i\omega x} \rangle_{L^2([-\pi, \pi])}} e^{-i\omega x}$$



# Fourier analysis (Euclidean spaces)

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

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

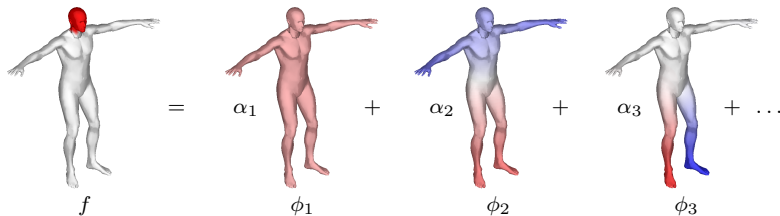


Fourier basis = **Laplacian eigenfunctions**:  $\Delta e^{-i\omega x} = \omega^2 e^{-i\omega x}$

# 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**:  $\Delta_X \phi_k(x) = \lambda_k \phi_k(x)$

# Convolution (Euclidean spaces)

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

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

# Convolution (Euclidean spaces)

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

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

**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

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

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

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

# Convolution (non-Euclidean spaces)

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

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

# Convolution (non-Euclidean spaces)

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

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

# Convolution (non-Euclidean spaces)

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

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

# Convolution (non-Euclidean spaces)

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

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

- Not shift-invariant!

# Convolution (non-Euclidean spaces)

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

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

- 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

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

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

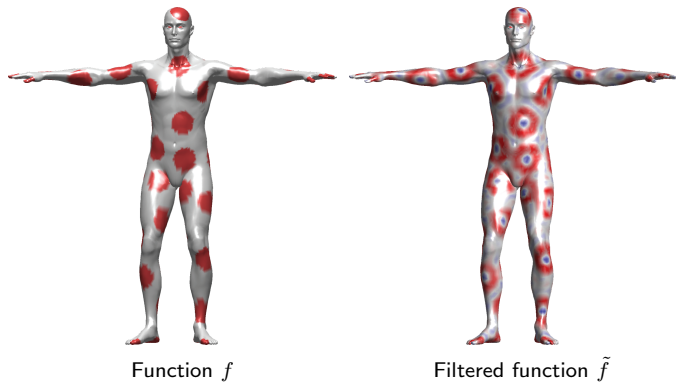
# Convolution (non-Euclidean spaces)

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

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

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

# Convolution (non-Euclidean spaces)



# Convolution (non-Euclidean spaces)



Function  $f$



Filtered function  $f$



Same filter  
another shape

# Heat diffusion on manifolds

$$\begin{cases} f_t(x, t) = -\Delta_X f(x, t) \\ f(x, 0) = f_0(x) \end{cases}$$

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

# Heat diffusion on manifolds

$$\begin{cases} f_t(x, t) = -\Delta_X f(x, t) \\ f(x, 0) = f_0(x) \end{cases}$$

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

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

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

# Heat diffusion on manifolds

$$\begin{cases} f_t(x, t) = -\Delta_X f(x, t) \\ f(x, 0) = f_0(x) \end{cases}$$

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

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

$$f(x, t) = e^{-t\Delta_X} f_0(x) = \sum_{k \geq 1} \langle f_0, \phi_k \rangle_{L^2(X)} e^{-t\lambda_k} \phi_k(x)$$

# Heat diffusion on manifolds

$$\begin{cases} f_t(x, t) = -\Delta_X f(x, t) \\ f(x, 0) = f_0(x) \end{cases}$$

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

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

$$\begin{aligned} f(x, t) &= e^{-t\Delta_X} f_0(x) = \sum_{k \geq 1} \langle f_0, \phi_k \rangle_{L^2(X)} e^{-t\lambda_k} \phi_k(x) \\ &= \int_X f_0(\xi) \sum_{k \geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi) d\xi \end{aligned}$$

# Heat diffusion on manifolds

$$\begin{cases} f_t(x, t) = -\Delta_X f(x, t) \\ f(x, 0) = f_0(x) \end{cases}$$

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

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

$$\begin{aligned} f(x, t) &= e^{-t\Delta_X} f_0(x) = \sum_{k \geq 1} \langle f_0, \phi_k \rangle_{L^2(X)} e^{-t\lambda_k} \phi_k(x) \\ &= \int_X f_0(\xi) \underbrace{\sum_{k \geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi) d\xi}_{\text{heat kernel } h_t(x, \xi)} \end{aligned}$$

# Heat diffusion on manifolds

$$\begin{cases} f_t(x, t) = -\Delta_X f(x, t) \\ f(x, 0) = f_0(x) \end{cases}$$

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

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

$$\begin{aligned} f(x, t) &= e^{-t\Delta_X} f_0(x) = \sum_{k \geq 1} \langle f_0, \phi_k \rangle_{L^2(X)} e^{-t\lambda_k} \phi_k(x) \\ &= \int_X f_0(\xi) \underbrace{\sum_{k \geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi) d\xi}_{\text{heat kernel } h_t(x, \xi)} \end{aligned}$$

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

# Heat diffusion on manifolds

$$\begin{cases} f_t(x, t) = -\Delta_X f(x, t) \\ f(x, 0) = f_0(x) \end{cases}$$

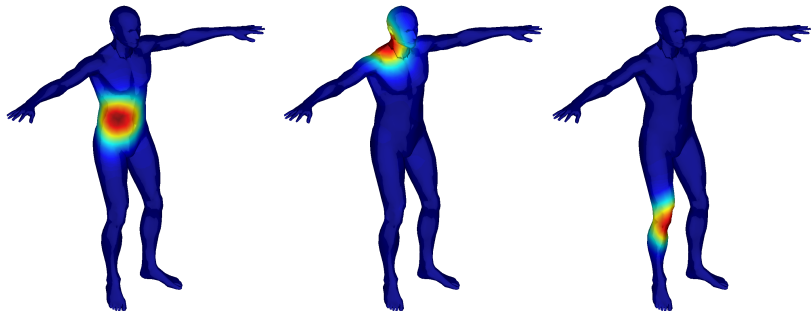
- $f(x, t)$  = amount of heat at point  $x$  at time  $t$
- $f_0(x)$  = initial heat distribution

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

$$\begin{aligned} f(x, t) &= e^{-t\Delta_X} f_0(x) = \sum_{k \geq 1} \langle f_0, \phi_k \rangle_{L^2(X)} e^{-t\lambda_k} \phi_k(x) \\ &= \int_X f_0(\xi) \underbrace{\sum_{k \geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi) d\xi}_{\text{heat kernel } h_t(x, \xi)} \end{aligned}$$

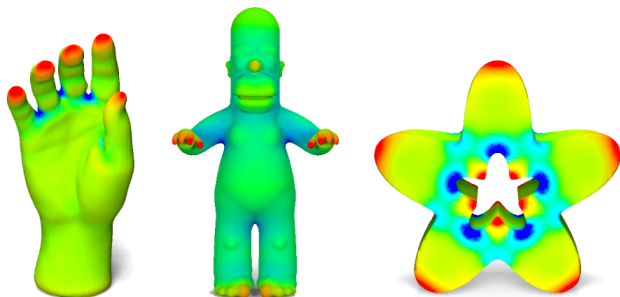
- “impulse response” to a delta-function at  $\xi$
- “how much heat is transferred from point  $x$  to  $\xi$  in time  $t$ ”

# Heat kernel



Heat kernels at different points  
(rows/columns of matrix  $e^{-t\Delta_x}$ )

# Autodiffusivity



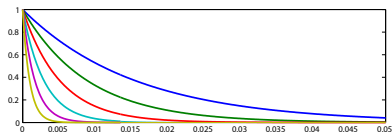
Autodiffusivity = diagonal of matrix  $e^{-t\Delta_x}$

Related to [Gaussian curvature](#) by virtue of the Taylor expansion

$$h_t(x, x) \approx \frac{1}{4\pi t} + \frac{K(x)}{12\pi} + \mathcal{O}(t)$$

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

## Heat Kernel Signature (HKS)

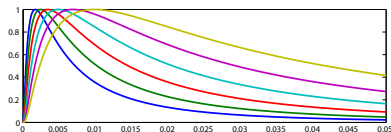


Low-pass filter bank

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

Heat autodiffusivity

## Wave Kernel Signature (WKS)



Band-pass filter bank

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

Probability of a quantum particle

# Optimal spectral descriptors

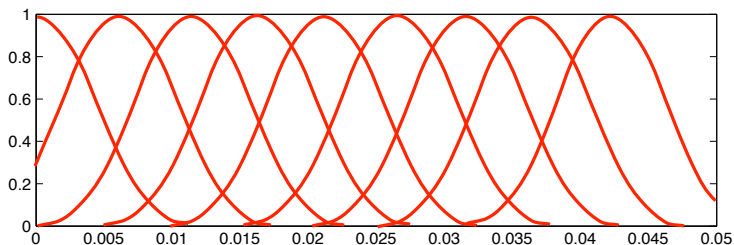
$$\mathbf{f}_{\boldsymbol{\tau}}(x) = \sum_{k \geq 1} \begin{pmatrix} \tau_1(\lambda_k) \\ \vdots \\ \tau_Q(\lambda_k) \end{pmatrix} \phi_k^2(x)$$

parametrized by frequency responses  $\boldsymbol{\tau}(\lambda) = (\tau_1(\lambda), \dots, \tau_Q(\lambda))^{\top}$

# Optimal spectral descriptors

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

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



# Optimal spectral descriptors

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

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

# Optimal spectral descriptors

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

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

# Optimal spectral descriptors

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

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

- Optimal  $\mathbf{A}$  in the spirit of **Wiener filter**:
  - attenuate frequencies with large noise content (deformation)
  - pass frequencies with large signal content (discriminative geometric features)

# Optimal spectral descriptors

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

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

- Optimal  $\mathbf{A}$  in the spirit of **Wiener filter**:
  - attenuate frequencies with large noise content (deformation)
  - pass frequencies with large signal content (discriminative geometric features)
- Hard to model axiomatically...

# Optimal spectral descriptors

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

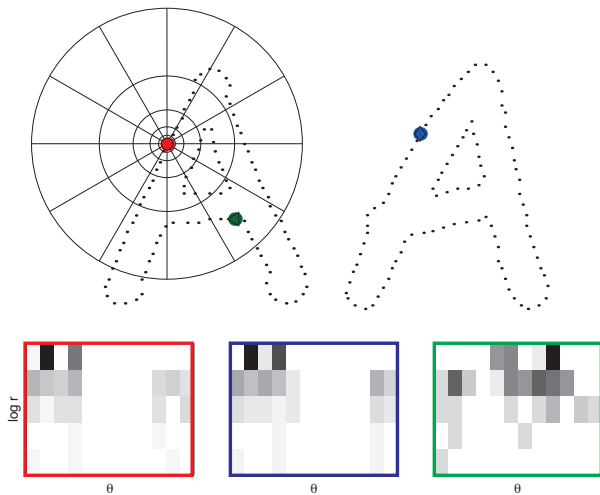
parametrized by linear combination coefficients  $\mathbf{A}$  of **geometry vectors**  
 $\mathbf{g}(x) = (g_1(x), \dots, g_M(x))^{\top}$

- Optimal  $\mathbf{A}$  in the spirit of **Wiener filter**:
  - attenuate frequencies with large noise content (deformation)
  - pass frequencies with large signal content (discriminative geometric features)
- Hard to model axiomatically...
- ...yet easy to **learn** from examples!

# The need for context



# Shape context



Belongie, Malik 2000

# 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

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

# Convolution on manifolds



# Convolution on manifolds



?

# Geodesic convolution

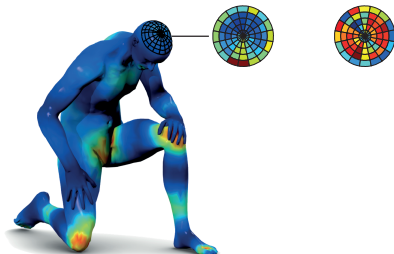
- **Geodesic convolution** = apply filter  $a$  to patches extracted from  $f \in L^2(X)$  in local geodesic polar coordinates

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

# Geodesic convolution

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

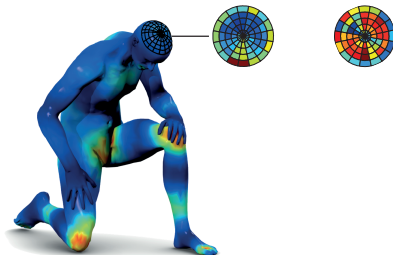
$$(f \star a)(x) = \sum_{\theta, r} \underbrace{(D(x)f)(r, \theta)}_{\text{patch}} \underbrace{a(\theta, r)}_{\text{filter}}$$



# Geodesic convolution

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

$$(f \star a)(x) = \sum_{\theta, r} \underbrace{(D(x)f)(r, \theta)}_{\text{patch}} \underbrace{a(\theta + \Delta\theta, r)}_{\text{filter}}$$

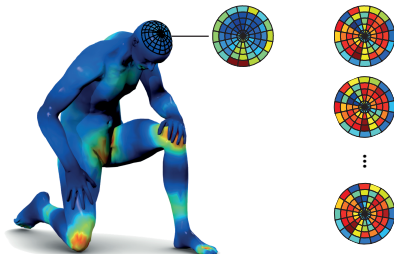


- Angular coordinate origin is arbitrary = **rotation ambiguity!**

# Geodesic convolution

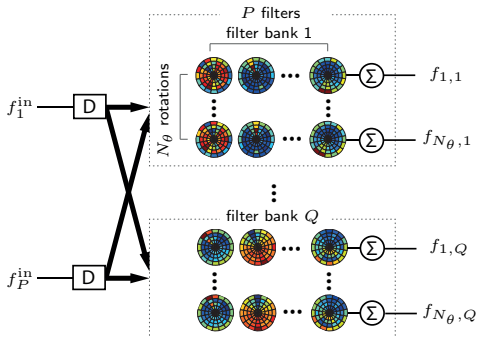
- **Geodesic convolution** = apply filter  $a$  to patches extracted from  $f \in L^2(X)$  in local geodesic polar coordinates

$$(f \star a)(x) = \sum_{\theta, r} \underbrace{(D(x)f)(r, \theta)}_{\text{patch}} \underbrace{a(\theta + \Delta\theta, r)}_{\text{filter}}$$



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

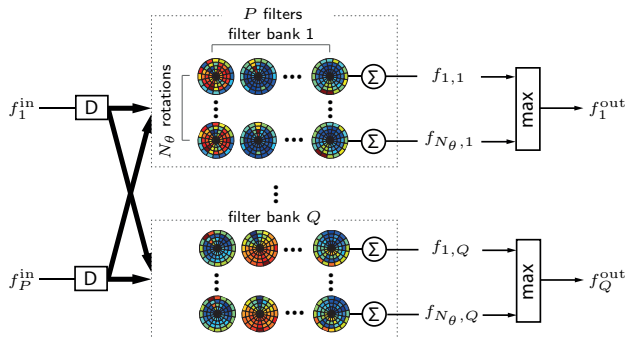
# Geodesic Convolution layer



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

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

# Geodesic Convolution layer

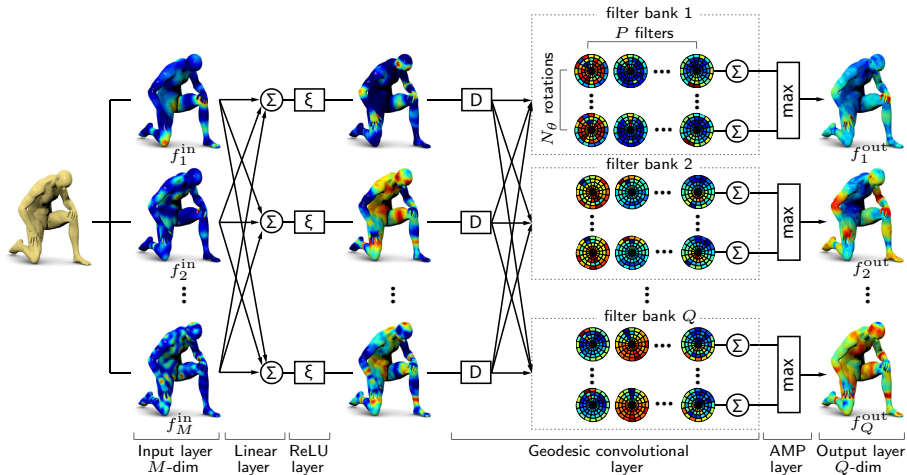


$$f_q^{\text{out}}(x) = \max_{\Delta\theta} \sum_{p=1}^P (f_p \star a_{\Delta\theta,qp})(x), \quad q = 1, \dots, Q$$

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

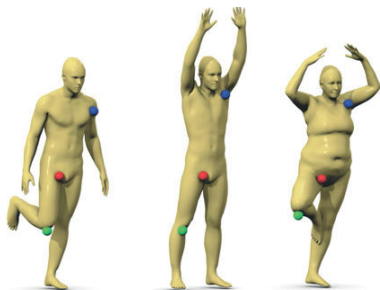
Masci, Boscaini, Bronstein, Vandergheynst 2015

# Toy ShapeNet architecture



Masci, Boscaini, Bronstein, Vandergheynst 2015

# Learning local descriptors with ShapeNet



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

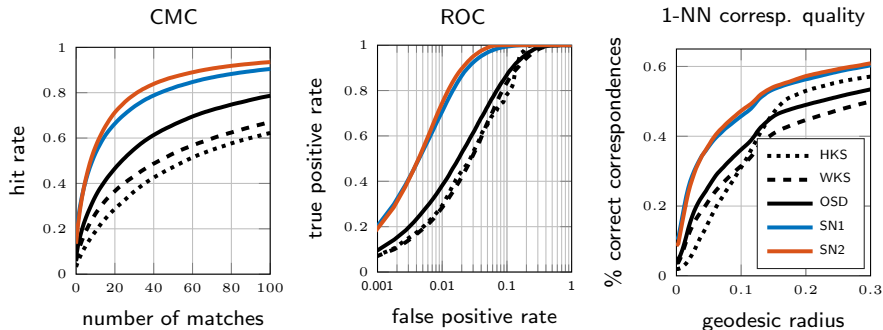
# Descriptor robustness



ShapeNet descriptor distance

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

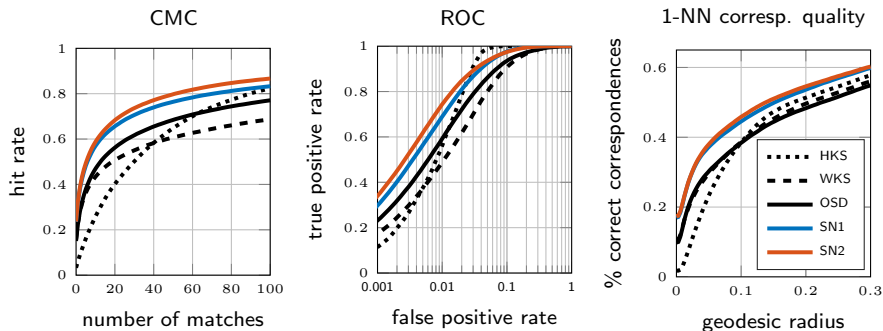
# Descriptor performance



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 (ShapeNet); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

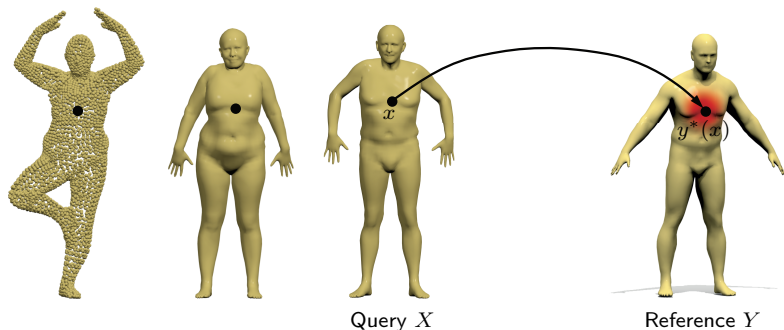
# Descriptor performance



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 (ShapeNet); data: Bogoy et al. 2014 (FAUST); Bronstein et al. 2008 (TOSCA); benchmark: Kim et al. 2011

# Learning shape correspondence with ShapeNet

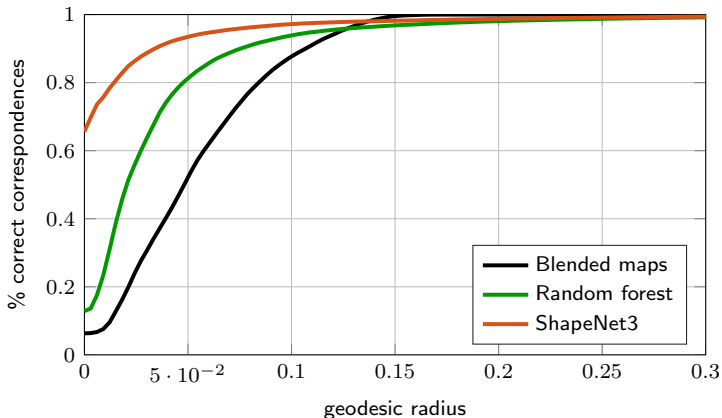


- Correspondence = **labeling problem**
- ShapeNet output  $\mathbf{f}_{\Theta}(x)$  = probability distribution on reference  $Y$
- Minimize **logistic regression** cost w.r.t. ShapeNet parameters  $\Theta$

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

Rodolà et al. 2014; Masci, Boscaini, Bronstein, Vandergheynst 2015

# ShapeNet correspondence performance



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

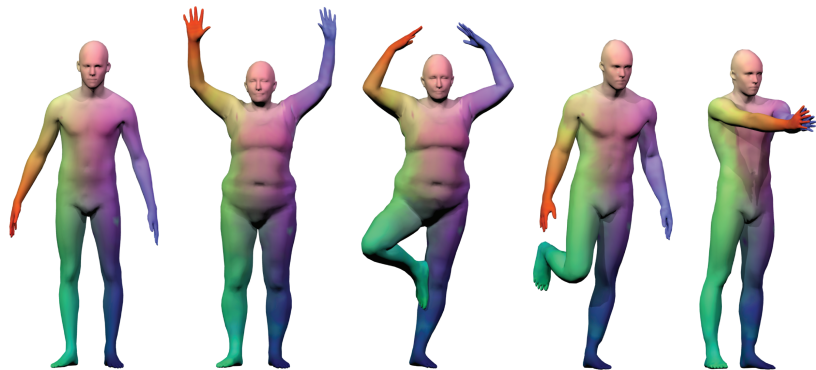
Masci, Boscaini, Bronstein, Vandergheynst 2015; Rodolà et al. 2014; Kim et al. 11

# Correspondence examples: Random forest



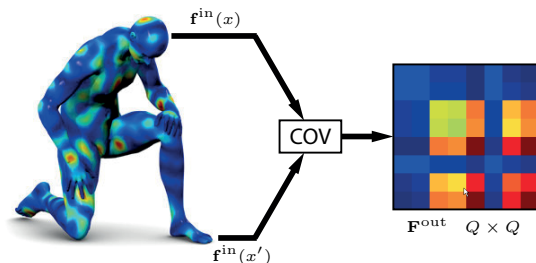
Correspondence found using random forest  
(similar colors encode corresponding points)

# Correspondence examples: ShapeNet



Correspondence found using ShapeNet  
(similar colors encode corresponding points)

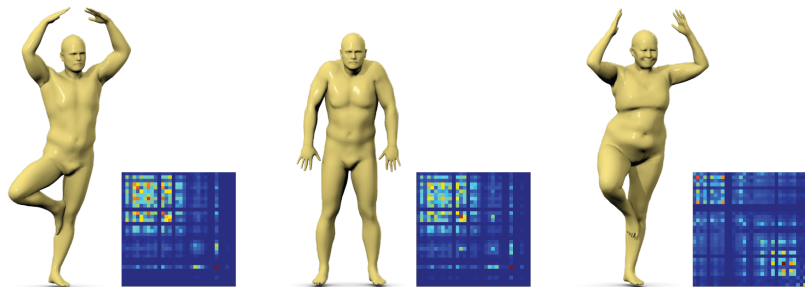
# From local to global features: covariance layer



$$\mathbf{F}^{\text{out}} = \int_X (\mathbf{f}^{\text{in}}(x) - \boldsymbol{\mu}_{\mathbf{f}^{\text{in}}})(\mathbf{f}^{\text{in}}(x) - \boldsymbol{\mu}_{\mathbf{f}^{\text{in}}})^\top dx$$
$$\boldsymbol{\mu}_{\mathbf{f}^{\text{in}}} = \int_X \mathbf{f}_{\text{in}}(x) dx$$

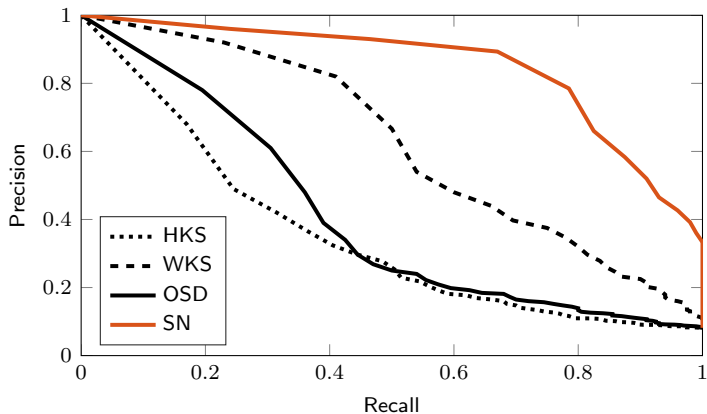
- Aggregates local features into a **global shape descriptor**

# Learning shape similarity with ShapeNet



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

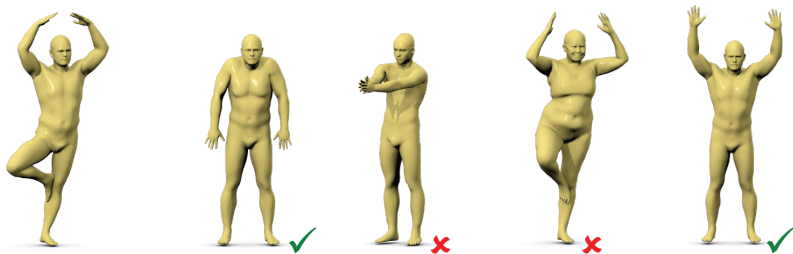
# ShapeNet retrieval performance



1-layer ShapeNet; Training and testing: FAUST

Masci, Boscaini, Bronstein, Vandergheynst 2015

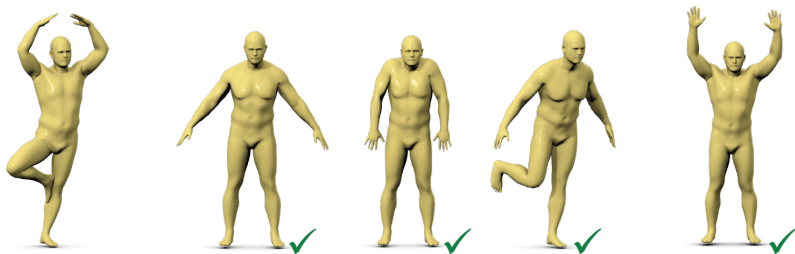
# Retrieval examples: HKS



Shape retrieval using similarity computed with HKS

Masci, Boscaini, Bronstein, Vandergheynst 2015; data: Pickup et al. 2014

# Retrieval examples: ShapeNet



Shape retrieval using similarity computed with ShapeNet

Masci, Boscaini, Bronstein, Vandergheynst 2015; data: Pickup et al. 2014

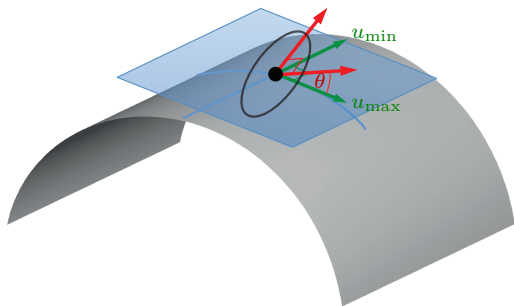
$$f_t(x) = -\operatorname{div}(c\nabla f(x))$$

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

$$f_t(x) = -\operatorname{div}(A(x)\nabla f(x))$$

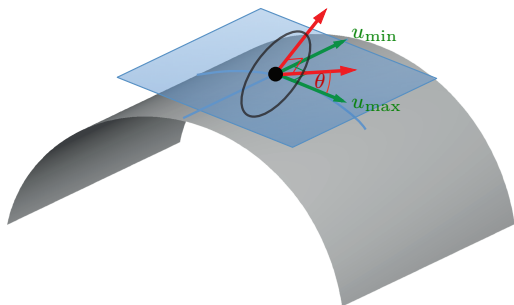
$A(x)$  = **heat conductivity tensor** describing heat conduction properties of the material (diffusion speed is position + direction dependent)

# Anisotropic diffusion on manifolds



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

# Anisotropic diffusion on manifolds



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

- **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'

Andreux et al. 2014; Boscaini, Masci, Rodolà, Bronstein, Cremers 2015

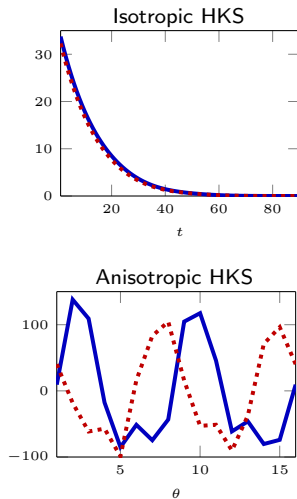
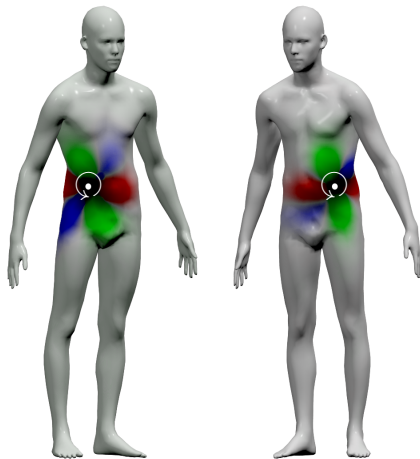
# Anisotropic heat kernels

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



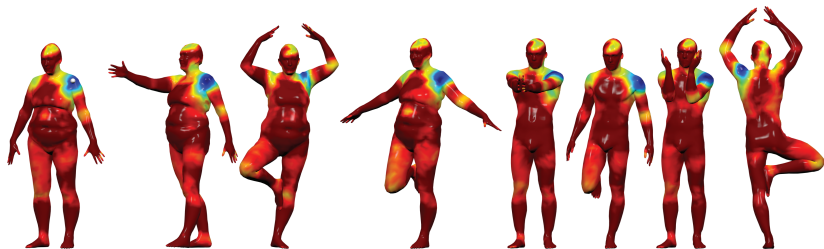
Examples of anisotropic heat kernels  $h_{\alpha\theta t}$  for different values of  $t$ ,  $\theta$  and  $\alpha$

# Isotropic vs Anisotropic HKS



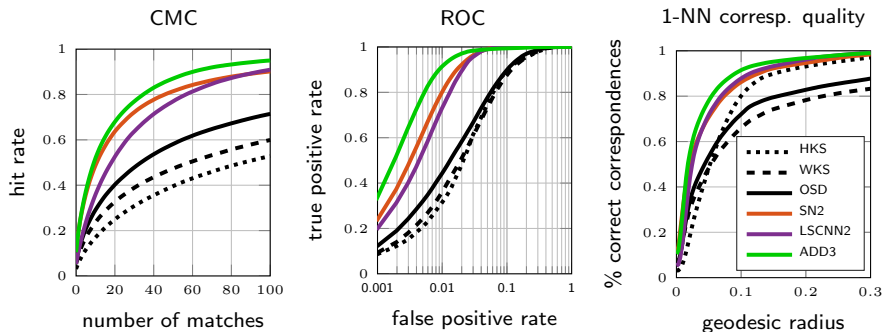
Boscaini, Masci, Rodolà, Bronstein, Cremers 2015

# Anisotropic Diffusion Descriptor (ADD) robustness



Anisotropic Diffusion Descriptor (ADD) distance

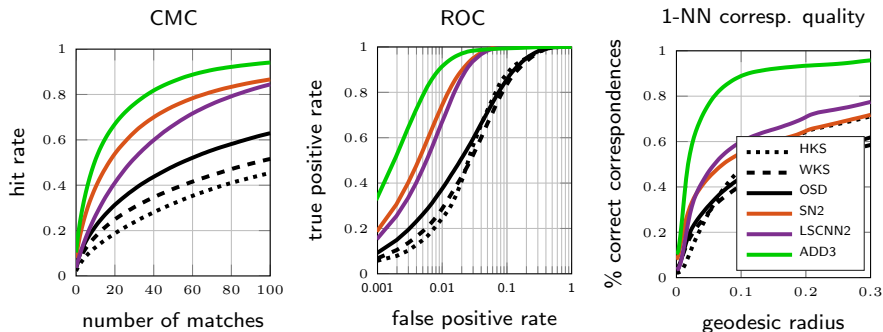
# Anisotropic Diffusion Descriptor (ADD) performance



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 (ShapeNet); 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 (ShapeNet); 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.

Thank you!