Geometric deep learning

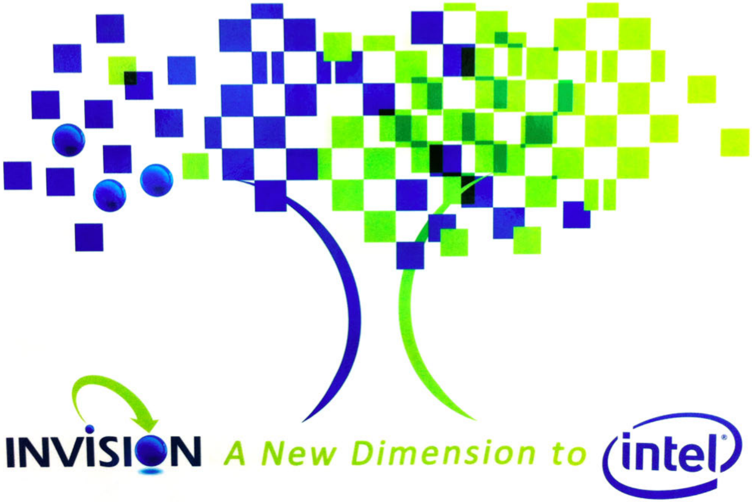
Michael Bronstein

University of Lugano

Intel Perceptual Computing

TUM, 1 August 2016

Microsoft Kinect 2010



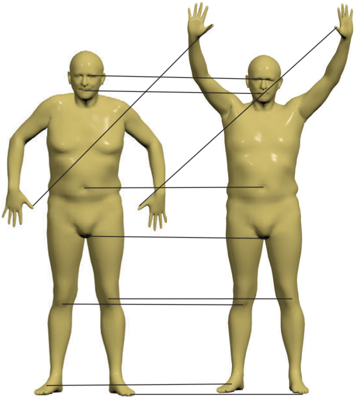
(Acquired by Intel in 2012)


intel
REALSENSE
TECHNOLOGY

Different form factor computers featuring Intel RealSense 3D camera

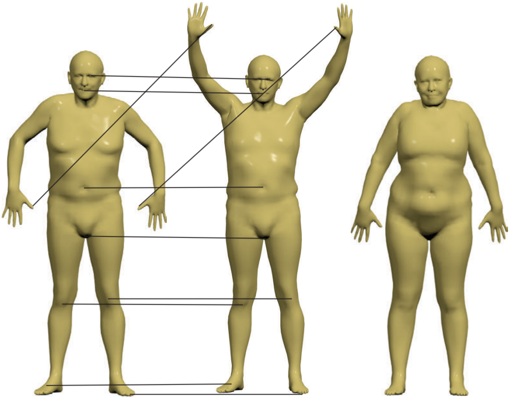
$100K	$100	$20
2005	**2010**	**2014**

Basic problems: shape similarity and correspondence



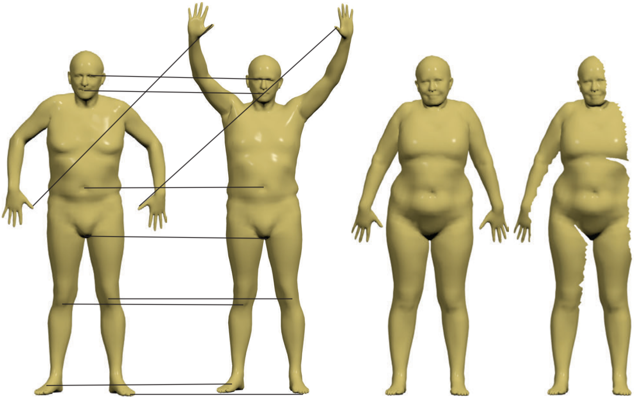
Isometric

Basic problems: shape similarity and correspondence



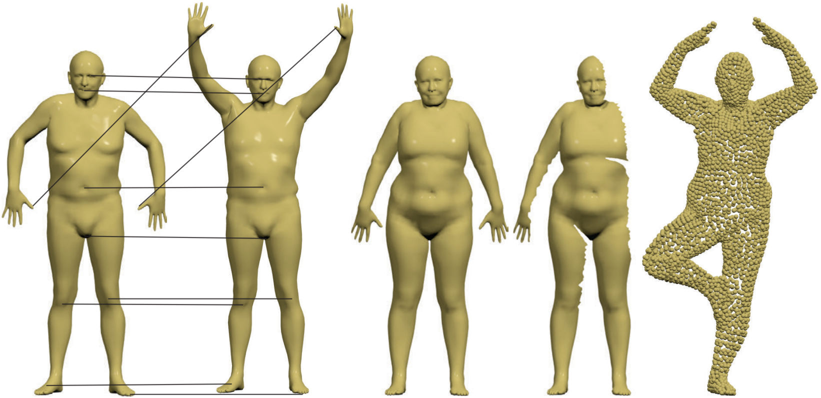
Isometric Non-isometric

Basic problems: shape similarity and correspondence



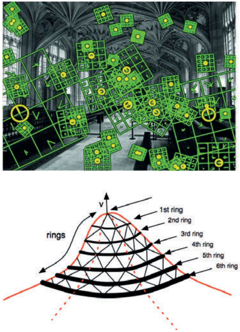
Isometric Non-isometric Partial

Basic problems: shape similarity and correspondence

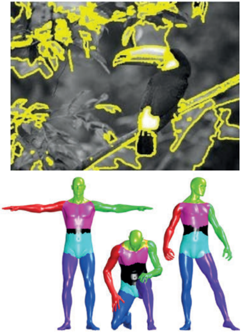


Isometric Non-isometric Partial Different representation

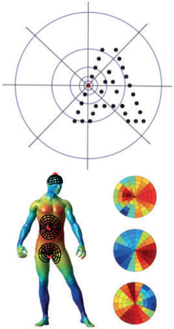
3D feature descriptors



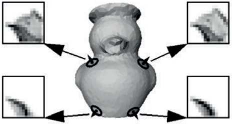
SIFT[1] / MeshHOG[2]



MSER[3] / ShapeMSER[4]



(Intrinsic[6]) Shape context[5]



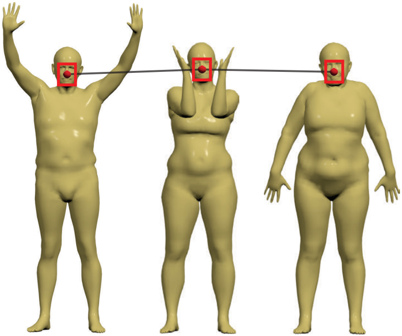
Spin image[7]

Heat kernel signature[8]

[1]Lowe 2004; [2]Zaharescu et al. 2009; [3]Matas et al. 2002; [4]Litman et al. 2010;
[5]Belongie et al. 2000; [6]Kokkinos et al. 2012; [7]Johnson et al. 1999; [8]Sun et al. 2009

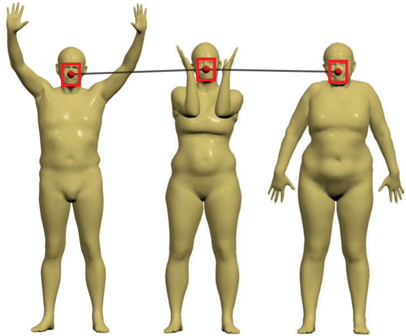
Task-specific features

Correspondence

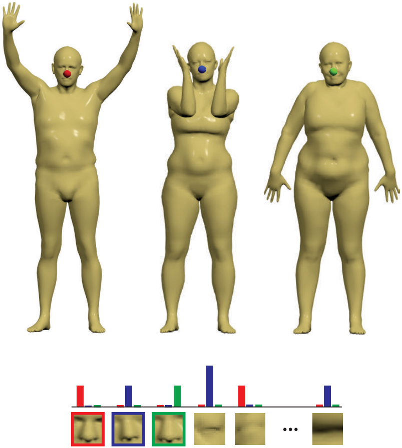


Task-specific features

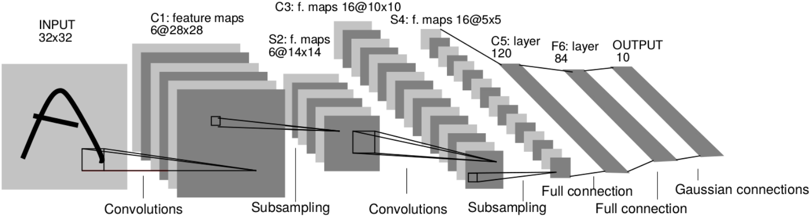
Correspondence



Similarity



Convolutional neural networks

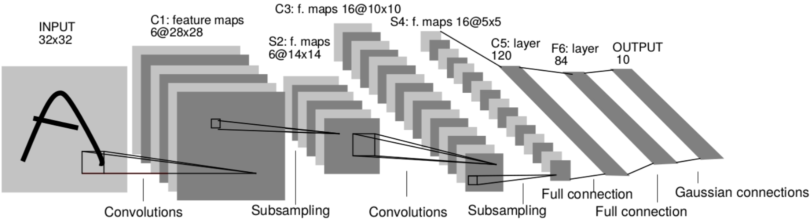


Typical CNN architecture

- Combination of convolution and pooling layers

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

Convolutional neural networks

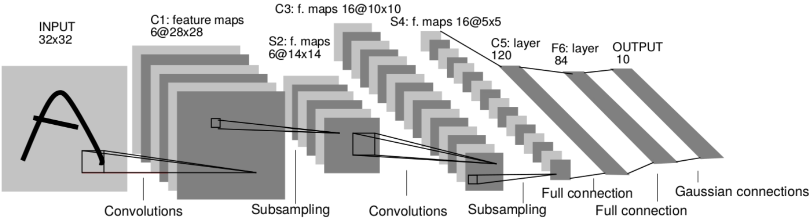


Typical CNN architecture

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

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

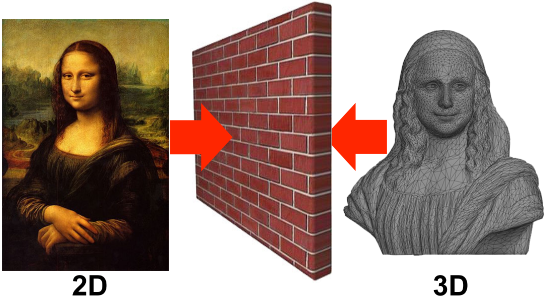
Convolutional neural networks

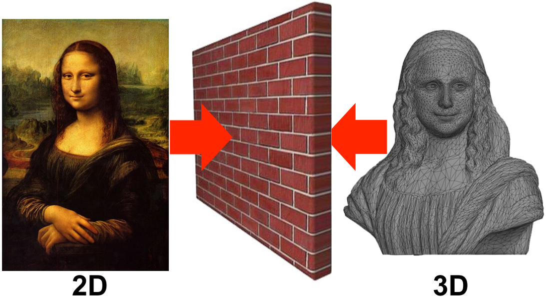


Typical CNN architecture

- 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

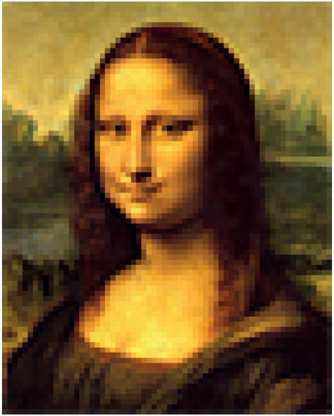
Fukushima 1980; LeCun et al. 1989; Image: H. Wang


2D
3D

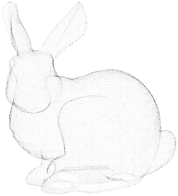


Generalize deep learning to non-Euclidean data
in a geometrically meaningful way

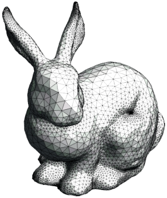
3D shapes vs images



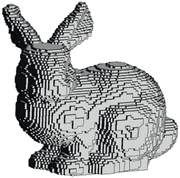
Array of pixels



Point cloud



Mesh



Voxels

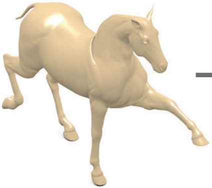
Level set

3D shapes vs images

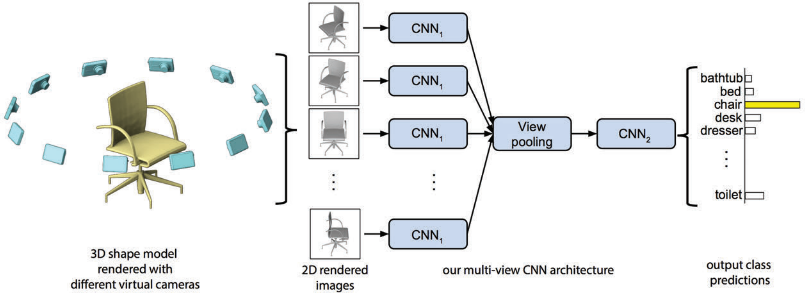
½  + ½  = 

3D shapes vs images

½  + ½  = 

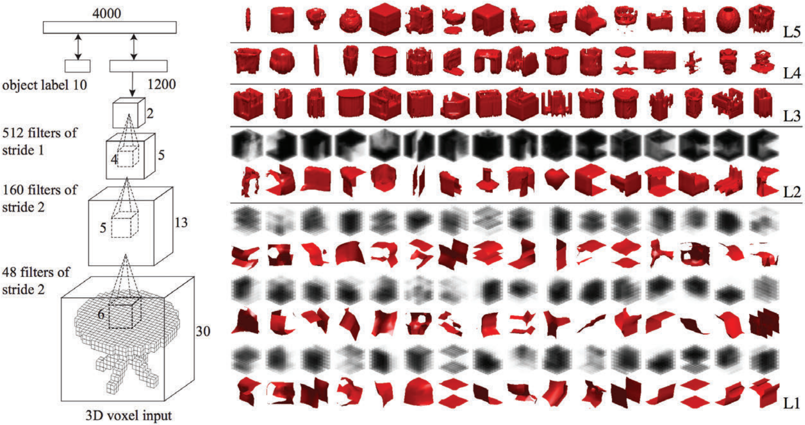
½  + ½  = ?

Deep learning on 3D data



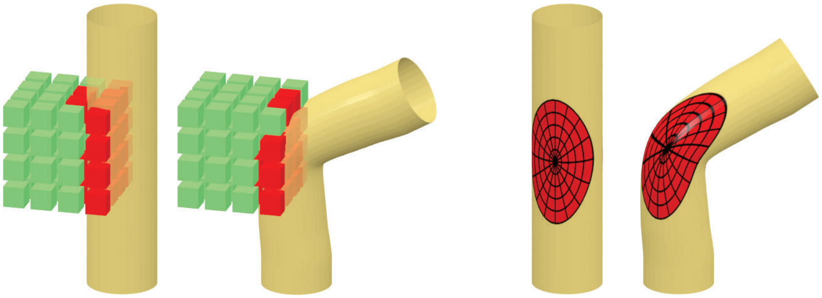
Su, Maji, Kalogerakis, Learned-Miller 2015

Deep learning on 3D data



Wu et al. 2015

Extrinsic vs Intrinsic



Extrinsic Intrinsic

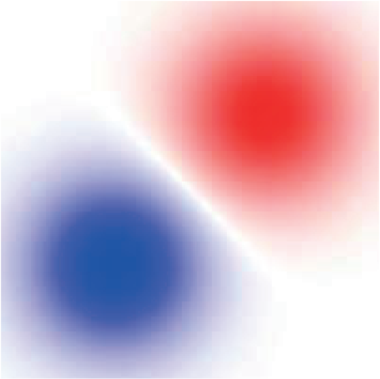
Outline

- Background: spectral analysis on manifolds
- Spectral shape descriptors
- Intrinsic convolutional neural networks on manifolds and point clouds
- Applications

Pierre-Simone de Laplace (1749-1827)

$$\Delta$$

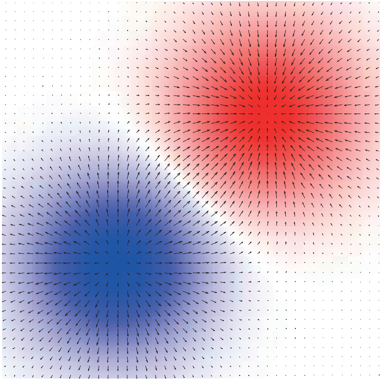
Laplacian in one minute



Smooth scalar field f

Laplacian in one minute

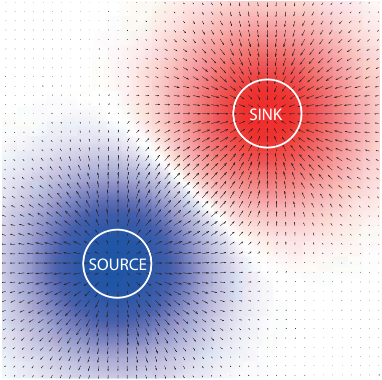
- Gradient $\nabla f(x)$ = 'direction of the steepest increase of f at x'



Smooth scalar field f

Laplacian in one minute

- Gradient $\nabla f(x) =$ 'direction of the steepest increase of f at x'
- Divergence $\mathrm{div}(F(x)) =$ 'density of an outward flux of F from an infinitesimal volume around x'



Smooth vector field F

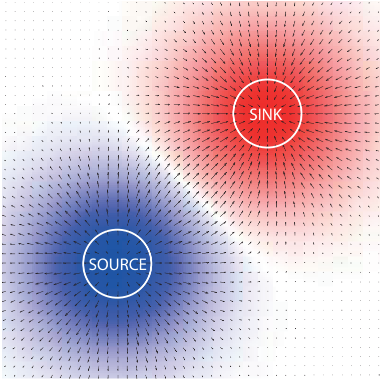
Laplacian in one minute

- Gradient $\nabla f(x)$ = 'direction of the steepest increase of f at x'
- Divergence $\mathrm{div}(F(x))$ = 'density of an outward flux of F from an infinitesimal volume around x'

  Divergence theorem:

$$\int_V \mathrm{div}(\mathrm{F}) dV = \int_{\partial V} \langle F, \hat{n} \rangle dS$$

'$\sum$ sources + sinks = net flow'



Smooth vector field F

Laplacian in one minute

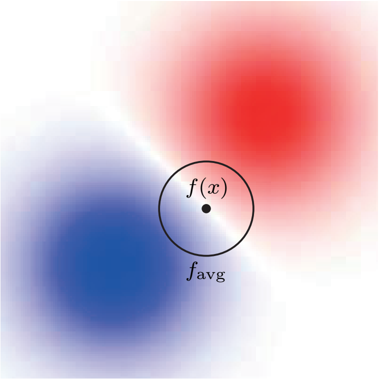
- Gradient $\nabla f(x) =$ 'direction of the steepest increase of f at x'

- Divergence $\mathrm{div}(F(x)) =$ 'density of an outward flux of F from an infinitesimal volume around x'

  Divergence theorem:

$$\int_V \mathrm{div}(\mathrm{F}) dV = \int_{\partial V} \langle F, \hat{n} \rangle dS$$

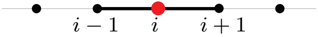
  '$\sum$ sources + sinks = net flow'



- Laplacian $\Delta f(x) = -\mathrm{div}(\nabla f(x))$
  'difference between $f(x)$ and the average of f on an infinitesimal sphere around x' (consequence of the Divergence theorem)

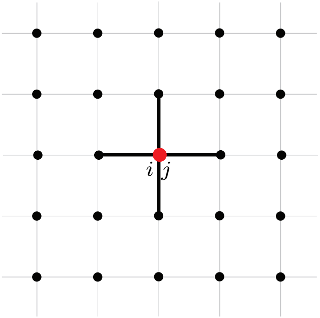
*We define Laplacian with negative sign

Discrete Laplacian (Euclidean)



One-dimensional

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



Two-dimensional

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

Physical application: heat equation

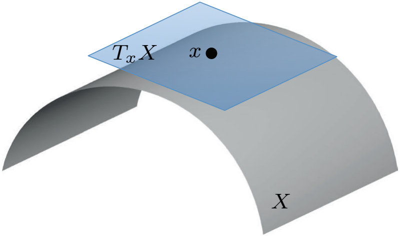
$$f_t = -c\Delta f$$

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^2/sec] $=$ thermal diffusivity constant (assumed $= 1$)

Riemannian geometry in one minute

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

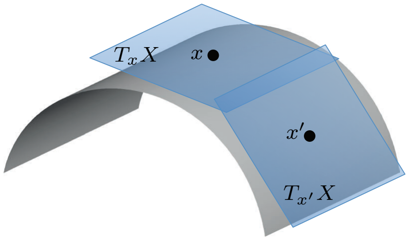


Riemannian geometry in one minute

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

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

depending smoothly on x



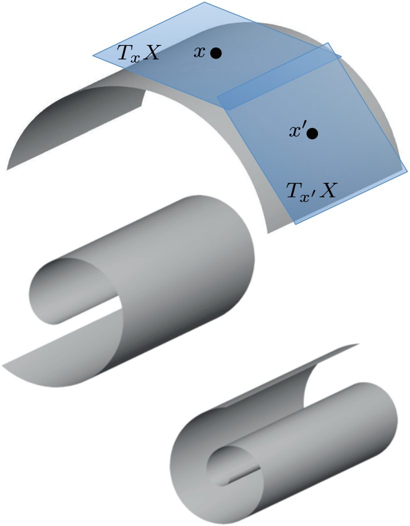
Riemannian geometry in one minute

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

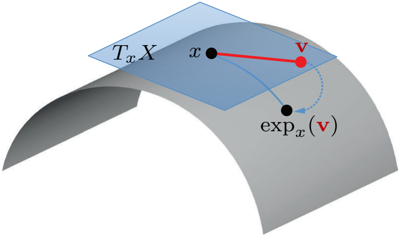
$$\langle \cdot, \cdot \rangle_{T_x X} : T_x X \times T_x X \to \mathbb{R}$$

depending smoothly on x

Isometry = metric-preserving shape deformation



Riemannian geometry in one minute



- Tangent plane T_xX = local Euclidean representation of manifold (surface) X around x

- Riemannian metric

  $$\langle \cdot, \cdot \rangle_{T_xX} : T_xX \times T_xX \to \mathbb{R}$$

  depending smoothly on x

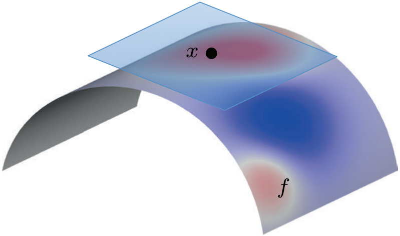
  Isometry = metric-preserving shape deformation

- Exponential map

  $$\exp_x : T_xX \to X$$

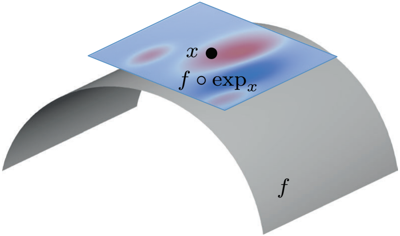
  'unit step along geodesic'

Laplace-Beltrami operator



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

Laplace-Beltrami operator

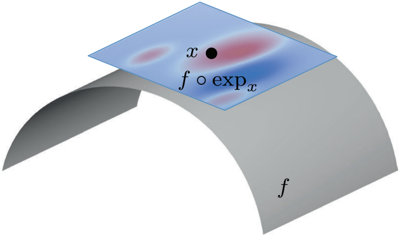


Smooth field $f \circ \exp_x : T_x X \to \mathbb{R}$

Laplace-Beltrami operator

- Intrinsic gradient

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



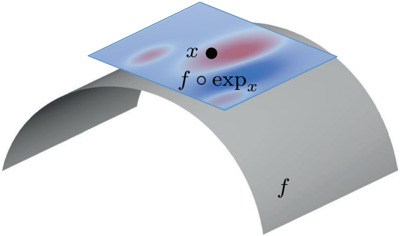
Laplace-Beltrami operator

- Intrinsic gradient

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

- Laplace-Beltrami operator

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



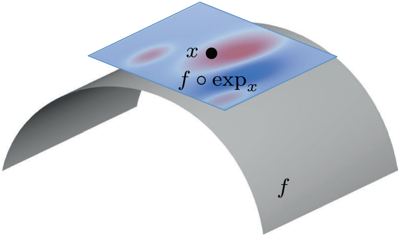
Laplace-Beltrami operator

- Intrinsic gradient

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

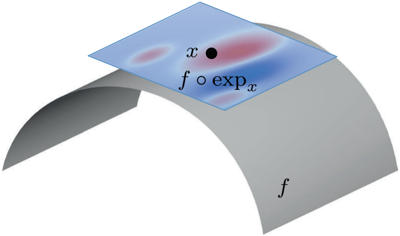
- Laplace-Beltrami operator

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



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

Laplace-Beltrami operator



- Intrinsic gradient

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

- 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

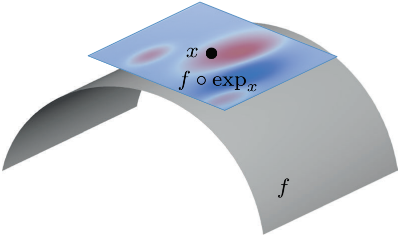
Laplace-Beltrami operator

- Intrinsic gradient

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

- 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 $\langle \Delta_X f, g \rangle_{L^2(X)} = \langle f, \Delta_X g \rangle_{L^2(X)}$

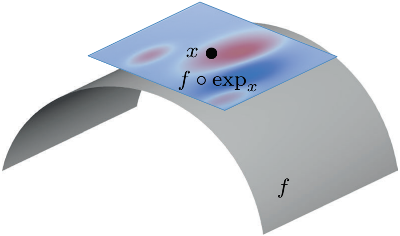
Laplace-Beltrami operator

- Intrinsic gradient

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

- 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

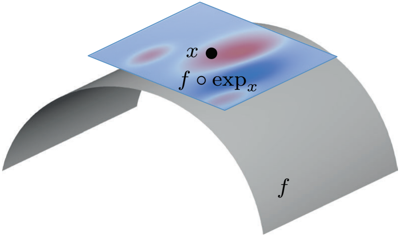
Laplace-Beltrami operator

- Intrinsic gradient

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

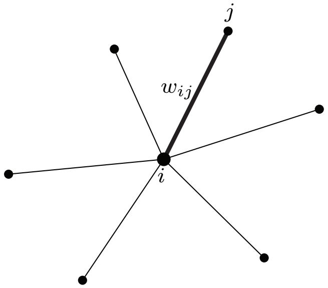
- Laplace-Beltrami operator

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



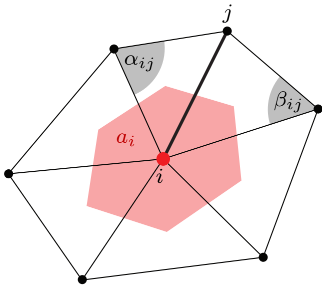
- Intrinsic (expressed solely in terms of the Riemannian metric)
- Isometry-invariant
- Self-adjoint $\langle \Delta_X f, g \rangle_{L^2(X)} = \langle f, \Delta_X g \rangle_{L^2(X)} \Rightarrow$ orthogonal eigenfunctions
- Positive semidefinite $\Rightarrow$ non-negative eigenvalues

Discrete Laplacian (non-Euclidean)



Undirected graph (V, E)

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



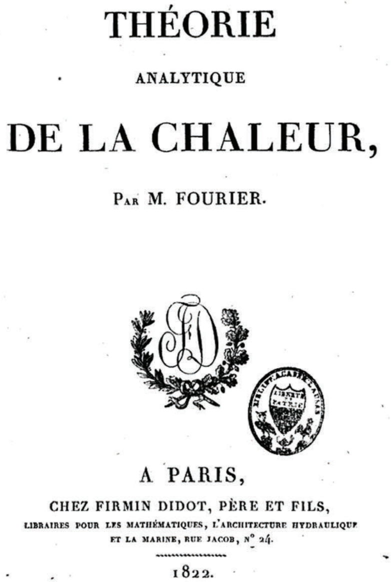
Triangular mesh (V, E, F)

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

$a_i =$ local area element

Tutte 1963; MacNeal 1949; Duffin 1959; Pinkall, Polthier 1993

Joseph Fourier (1768-1830)



THÉORIE
ANALYTIQUE
DE LA CHALEUR,

PAR M. FOURIER.

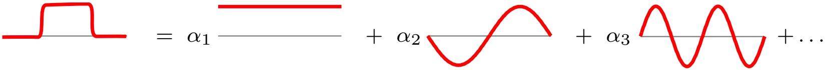
A PARIS,
CHEZ FIRMIN DIDOT, PÈRE ET FILS,
LIBRAIRES POUR LES MATHÉMATIQUES, L'ARCHITECTURE HYDRAULIQUE
ET LA MARINE, RUE JACOB, N° 24.

1822.

Fourier analysis (Euclidean spaces)

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

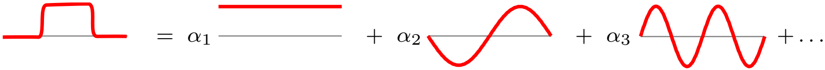
$$f(x) = \sum_{k \geq 0} \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\xi) e^{ik\xi} d\xi \; e^{-ikx}$$



Fourier analysis (Euclidean spaces)

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

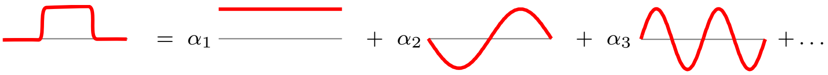
$$f(x) = \sum_{k \geq 0} \underbrace{\frac{1}{2\pi} \int_{-\pi}^{\pi} f(\xi) e^{ik\xi} d\xi}_{\hat{f}_k = \langle f, e^{-ikx} \rangle_{L^2([-\pi,\pi])}} \; e^{-ikx}$$



Fourier analysis (Euclidean spaces)

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

$$f(x) = \sum_{k \geq 0} \underbrace{\frac{1}{2\pi} \int_{-\pi}^{\pi} f(\xi) e^{ik\xi} d\xi}_{\hat{f}_k = \langle f, e^{-ikx} \rangle_{L^2([-\pi,\pi])}} e^{-ikx}$$

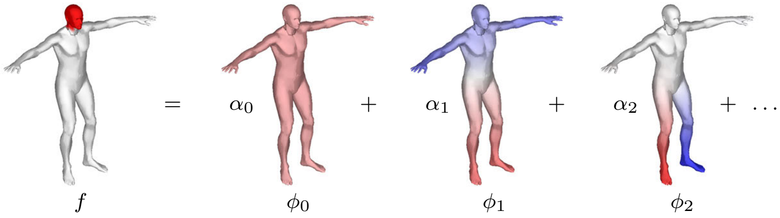


Fourier basis = Laplacian eigenfunctions: $\Delta e^{-ikx} = k^2 e^{-ikx}$

Fourier analysis (non-Euclidean spaces)

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

$$f(x) = \sum_{k \geq 0} \underbrace{\int_X f(\xi)\phi_k(\xi)d\xi}_{\hat{f}_k = \langle f, \phi_k \rangle_{L^2(X)}} \; \phi_k(x)$$



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

Convolution (Euclidean spaces)

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

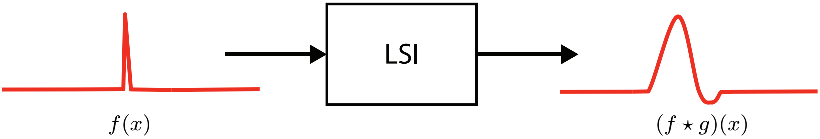
$$(f \star g)(x) = \int_{-\pi}^{\pi} f(\xi) g(x - \xi) d\xi$$

d'Alembert 1754

Convolution (Euclidean spaces)

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

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

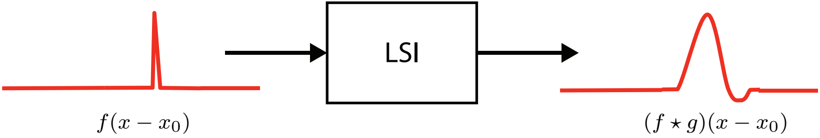


d'Alembert 1754

Convolution (Euclidean spaces)

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

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

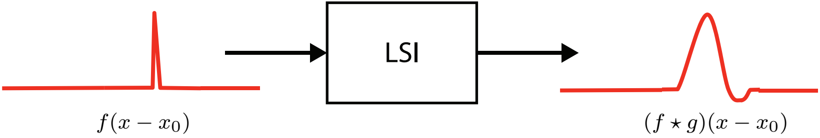


d'Alembert 1754

Convolution (Euclidean spaces)

Given two functions $f, g : [-\pi, \pi] \to \mathbb{R}$ 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

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

d'Alembert 1754; Borel 1899

Convolution (non-Euclidean spaces)

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

$$(f \star g)(x) \quad = \quad \sum_{k \geq 0} \langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)} \, \phi_k(x)$$

Shuman et al. 2013

Convolution (non-Euclidean spaces)

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

$$(f \star g)(x) \quad = \quad \sum_{k \geq 0} \underbrace{\langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)}}_{\text{product in the Fourier domain}} \phi_k(x)$$

Shuman et al. 2013

Convolution (non-Euclidean spaces)

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

$$(f \star g)(x) \quad = \quad \underbrace{\sum_{k \geq 0} \underbrace{\langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)}}_{\text{product in the Fourier domain}} \phi_k(x)}_{\text{inverse Fourier transform}}$$

Shuman et al. 2013

Convolution (non-Euclidean spaces)

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

$$(f \star g)(x) \quad = \quad \underbrace{\sum_{k \geq 0} \underbrace{\langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)}}_{\text{product in the Fourier domain}} \phi_k(x)}_{\text{inverse Fourier transform}}$$

- Not shift-invariant!

Shuman et al. 2013

Convolution (non-Euclidean spaces)

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

$$(f \star g)(x) = \underbrace{\sum_{k \geq 0} \underbrace{\langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)}}_{\text{product in the Fourier domain}} \phi_k(x)}_{\text{inverse Fourier transform}}$$

- Not shift-invariant!
- Represent filter in the Fourier domain

Shuman et al. 2013; Henaff, Bruna, LeCun 2015

Convolution (non-Euclidean spaces)

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

$$(f \star g)(x) \quad = \quad \underbrace{\sum_{k \geq 0} \underbrace{\langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)}}_{\text{product in the Fourier domain}} \phi_k(x)}_{\text{inverse Fourier transform}}$$

- Not shift-invariant!
- Represent filter in the Fourier domain
- Problem: Filter coefficients depend on basis $\{\phi_k\}_{k \geq 0}$

Shuman et al. 2013; Henaff, Bruna, LeCun 2015

Convolution (non-Euclidean spaces)

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

$$(f \star g)(x) \quad = \quad \underbrace{\sum_{k \geq 0} \underbrace{\langle f, \phi_k \rangle_{L^2(X)} \langle g, \phi_k \rangle_{L^2(X)}}_{\text{product in the Fourier domain}} \phi_k(x)}_{\text{inverse Fourier transform}}$$

- Not shift-invariant!
- Represent filter in the Fourier domain
- Problem: Filter coefficients depend on basis $\{\phi_k\}_{k \geq 0}$
  $\Rightarrow$ **does not generalize to other domains!**

Shuman et al. 2013; Henaff, Bruna, LeCun 2015

Convolution (non-Euclidean spaces)

Function f

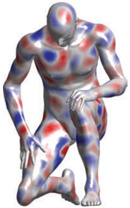
Filtered function $\tilde{f}$

Henaff, Bruna, LeCun 2015

Convolution (non-Euclidean spaces)

Function f

Filtered function f



Same filter
another shape

Henaff, Bruna, LeCun 2015

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

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

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

$$f(x,t) \quad = \quad e^{-t\Delta_X} f_0(x) = \sum_{k\geq 0} \langle f_0, \phi_k \rangle_{L^2(X)} e^{-t\lambda_k} \phi_k(x)$$

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

$$\begin{aligned} f(x,t) &= e^{-t\Delta_X} f_0(x) = \sum_{k\geq 0} \langle f_0, \phi_k \rangle_{L^2(X)} e^{-t\lambda_k} \phi_k(x) \\ &= \int_X f_0(\xi) \sum_{k\geq 0} e^{-t\lambda_k} \phi_k(x)\phi_k(\xi)\, d\xi \end{aligned}$$

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

$$\begin{aligned} f(x,t) &= e^{-t\Delta_X} f_0(x) = \sum_{k\geq 0} \langle f_0, \phi_k \rangle_{L^2(X)} e^{-t\lambda_k} \phi_k(x) \\ &= \int_X f_0(\xi) \underbrace{\sum_{k\geq 0} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi)}_{\text{heat kernel } h_t(x,\xi)} d\xi \end{aligned}$$

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

$$\begin{aligned} f(x,t) &= e^{-t\Delta_X} f_0(x) = \sum_{k\geq 0} \langle f_0, \phi_k \rangle_{L^2(X)} e^{-t\lambda_k} \phi_k(x) \\ &= \int_X f_0(\xi) \underbrace{\sum_{k\geq 0} e^{-t\lambda_k} \phi_k(x)\phi_k(\xi)}_{\text{heat kernel } h_t(x,\xi)} d\xi \end{aligned}$$

- "impulse response" to a delta-function at ξ

Heat diffusion on manifolds

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

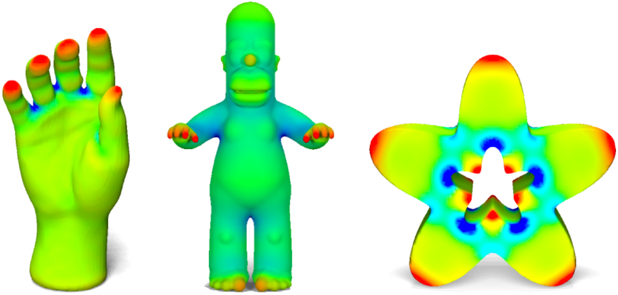
- $f(x,t)$ = amount of heat at point x at time t
- $f_0(x)$ = initial heat distribution

Solution of the heat equation expressed through the heat operator

$$\begin{aligned} f(x,t) &= e^{-t\Delta_X} f_0(x) = \sum_{k\geq 0} \langle f_0, \phi_k \rangle_{L^2(X)} e^{-t\lambda_k} \phi_k(x) \\ &= \int_X f_0(\xi) \underbrace{\sum_{k\geq 0} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi)}_{\text{heat kernel } h_t(x,\xi)} d\xi \end{aligned}$$

- "impulse response" to a delta-function at ξ
- "how much heat is transferred from point x to ξ in time t"

Autodiffusivity



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

Related to Gaussian curvature by virtue of the Taylor expansion

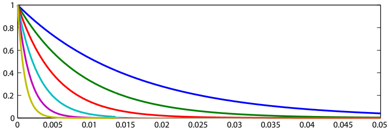
$$h_t(x,x) \approx \frac{1}{4\pi t} + \frac{K(x)}{12\pi} + \mathcal{O}(t)$$

Sun, Ovsjanikov, Guibas 2009

Spectral descriptors

$$\mathbf{f}(x) = \sum_{k \geq 0} \begin{pmatrix} \tau_1(\lambda_k) \\ \vdots \\ \tau_Q(\lambda_k) \end{pmatrix} \phi_k^2(x)$$

Heat Kernel Signature (HKS)

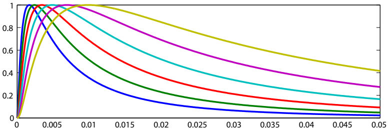


Low-pass filter bank

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

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

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

HKS descriptor robustness

HKS descriptor distance

Data: B et al. 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)

WKS descriptor robustness

WKS descriptor distance

Data: B et al. 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)

WKS descriptor robustness

WKS descriptor distance

Ambiguous to intrinsic symmetry!

Data: B et al. 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)

Convolution on manifolds

Convolution on manifolds

?

Homogeneous diffusion

$$f_t(x) = -c\Delta f(x)$$

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

Homogeneous diffusion

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

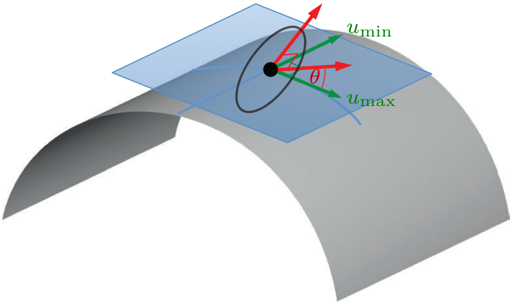
$c =$ thermal diffusivity constant describing heat conduction properties of the material (diffusion speed is equal everywhere)

Anisotropic diffusion

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

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

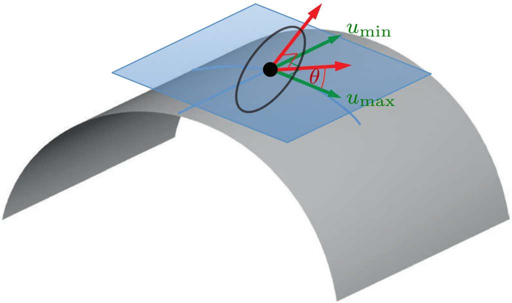
Anisotropic diffusion on manifolds



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

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

Anisotropic diffusion on manifolds



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

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

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

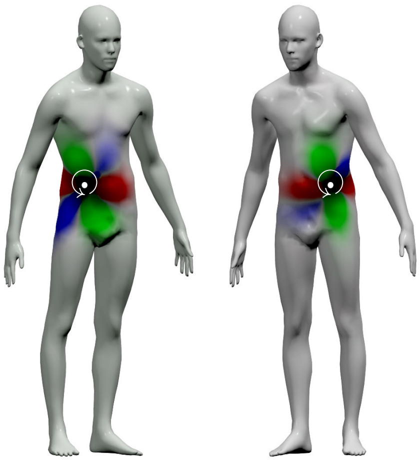
Anisotropic heat kernels

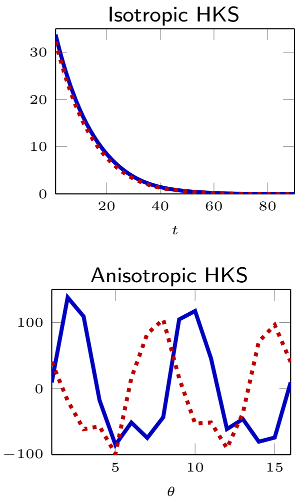
$$h_{\alpha\theta t}(x, \xi) = \sum_{k \geq 0} e^{-t\lambda_{\alpha\theta k}} \phi_{\alpha\theta k}(x) \phi_{\alpha\theta k}(\xi)$$

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

Boscaini, Masci, Rodolà, B, Cremers 2015

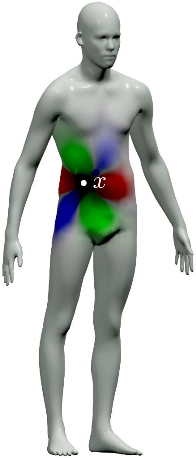
Intrinsic symmetry





Boscaini, Masci, Rodolà, B, Cremers 2015

Anisotropic diffusion patch operator



Given a function $f \in L^2(X)$, the patch operator

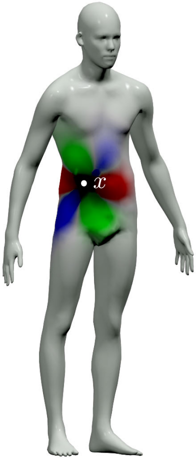
$$(D(x)f)(\theta, t) = \langle f, h_{\alpha\theta t}(x, \cdot) \rangle_{L^2(X)}$$

produces a local representation of f around point x

- θ = 'angular coordinate'
- t = 'radial coordinate'

Boscaini, Masci, Rodolà, B 2016

Anisotropic diffusion patch operator



Given a function $f \in L^2(X)$, the patch operator

$$(D(x)f)(\theta, t) = \langle f, h_{\alpha\theta t}(x, \cdot)\rangle_{L^2(X)}$$

produces a local representation of f around point x

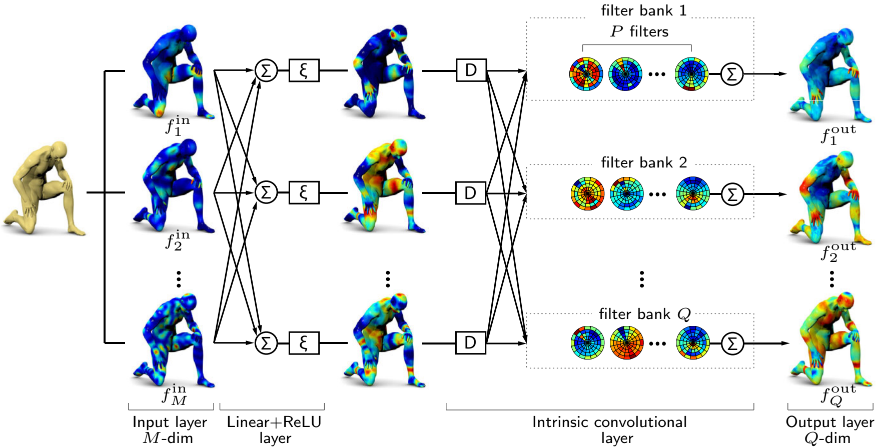
- θ = 'angular coordinate'
- t = 'radial coordinate'

Intrinsic convolution

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

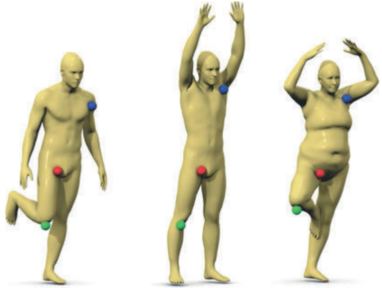
Boscaini, Masci, Rodolà, B 2016

Toy Anisotropic CNN (ACNN) architecture



Masci, Boscaini, B, Vandergheynst 2015; Boscaini, Masci, Rodolà, B 2016

Learning local descriptors

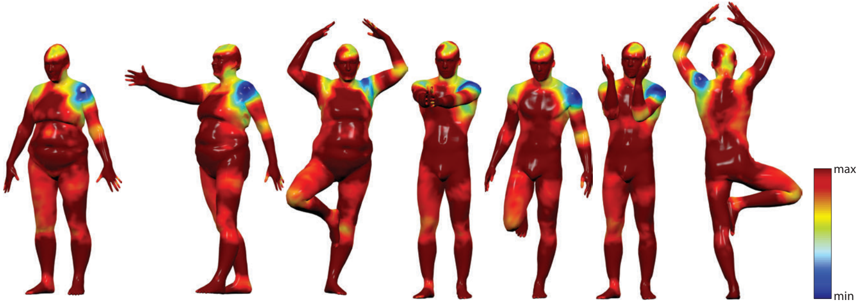


- As similar as possible on positives $\mathcal{T}^+$
- As dissimilar as possible on negatives $\mathcal{T}^-$
- Minimize siamese loss w.r.t. ShapeNet parameters $\mathbf{\Theta}$

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

Masci, Boscaini, B, Vandergheynst 2015; Boscaini, Masci, Rodolà, B 2016

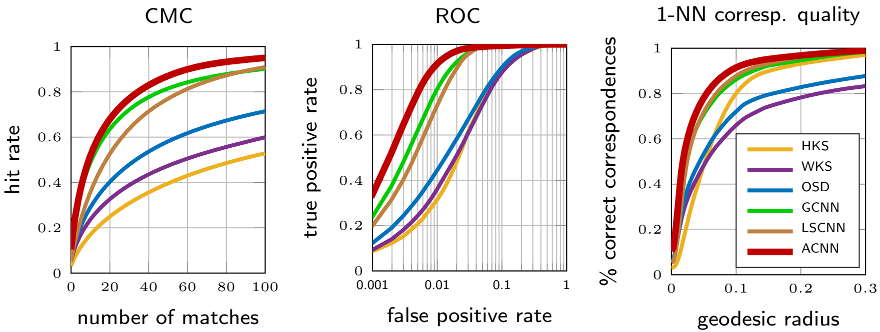
Learned descriptors robustness



Distance between descriptors produced by ACNN

Boscaini, Masci, Rodolà, B, Cremers 2015; Boscaini, Masci, Rodolà, B 2016

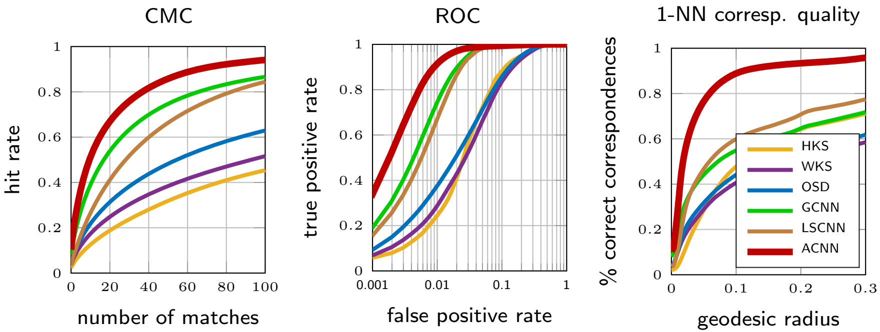
Learned descriptors performance



Descriptor performance using **symmetric** Princeton benchmark
(training and testing: disjoint subsets of FAUST)

Methods: Sun et al. 2009 (HKS); Aubry et al. 2011 (WKS); Litman, B 2014 (OSD); Masci, Boscaini, B, Vandergheynst 2015 (GCNN); Boscaini, Masci, Melzi, B, Castellani, Vandergheynst 2015 (LSCNN); Boscaini, Masci, Rodolà, B, Cremers 2015; Boscaini, Masci, Rodolà, B 2016 (ACNN); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

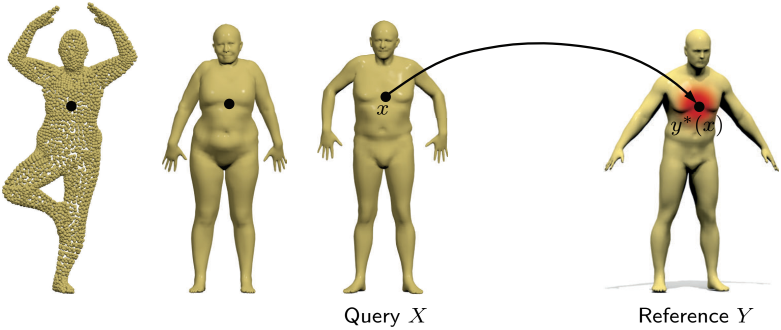
Learned descriptor performance



Descriptor performance using **asymmetric** Princeton benchmark
(training and testing: disjoint subsets of FAUST)

Methods: Sun et al. 2009 (HKS); Aubry et al. 2011 (WKS); Litman, B 2014 (OSD); Masci, Boscaini, B, Vandergheynst 2015 (GCNN); Boscaini, Masci, Melzi, B, Castellani, Vandergheynst 2015 (LSCNN); Boscaini, Masci, Rodolà, B, Cremers 2015; Boscaini, Masci, Rodolà, B 2016 (ACNN); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

Learning shape correspondence

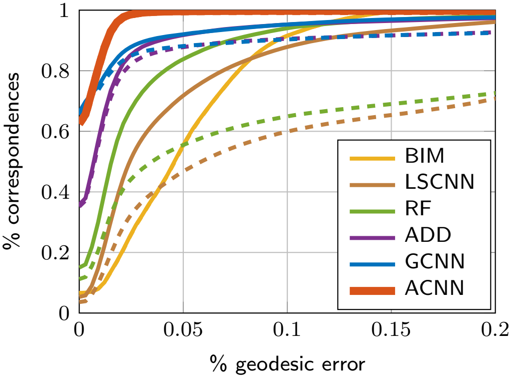


- Correspondence = labeling problem
- ShapeNet output $\mathbf{f}_{\mathbf{\Theta}}(x)$ = probability distribution on reference Y
- Minimize logistic regression cost w.r.t. ShapeNet parameters $\mathbf{\Theta}$

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

Rodolà et al. 2014; Masci, Boscaini, B, Vandergheynst 2015; Boscaini, Masci, Rodolà, B 2016

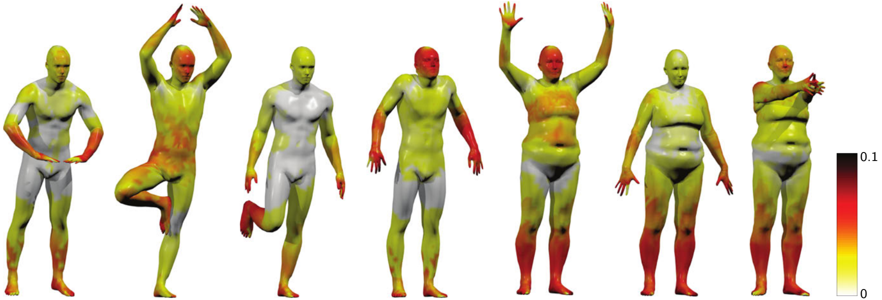
Correspondence performance



Symmetric (solid) and asymmetric (dotted) Princeton benchmark

Methods: Kim et al. 2011 (BIM); Boscaini, Masci, Melzi, B, Castellani, Vandergheynst 2015 (LSCNN); Rodolà et al. 2014 (RF); Boscaini, Masci, Rodolà, B, Cremers 2015 (ADD); Masci, Boscaini, B, Vandergheynst 2015 (GCNN); Boscaini, Masci, Rodolà, B 2016 (ACNN); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

Correspondence error: Blended Intrinsic Map



Pointwise geodesic error (in % of geodesic diameter)

Kim, Lipman, Funkhouser 2011

Correspondence error: ACNN

Pointwise geodesic error (in % of geodesic diameter)

Boscaini, Masci, Rodolà, B 2016

Learned correspondence visualization

Texture transferred from reference to query shapes
using correspondence learned by ACNN

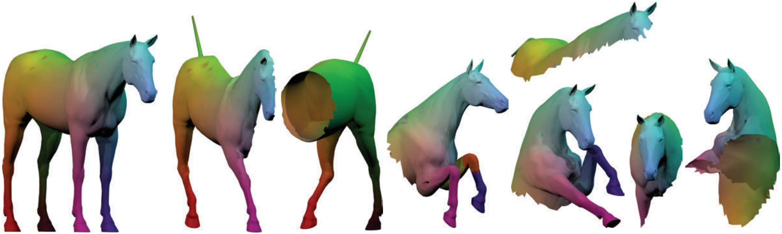
Boscaini, Masci, Rodolà, B 2016

Learned correspondence on point clouds

Colors transferred from reference to query shapes using correspondence learned with ACNN (similar colors encode corresponding points)

Boscaini, Masci, Rodolà, B, Cremers 2015

Learning partial correspondence



ACNN Correspondence

ACNN Correspondence error

Boscaini, Masci, Rodolà, B 2016

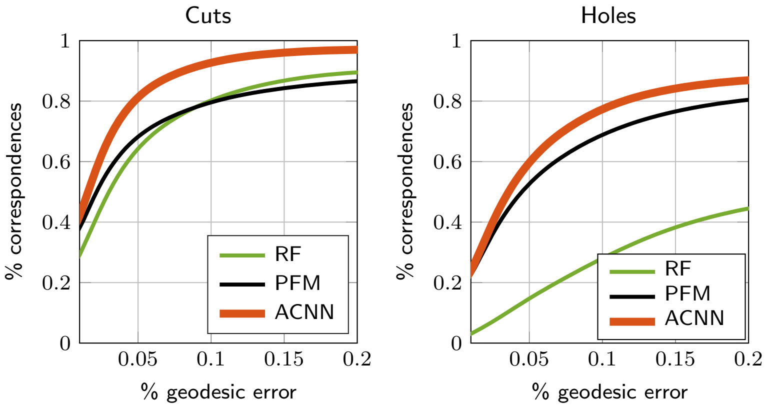
Learning partial correspondence

ACNN correspondence

ACNN correspondence error

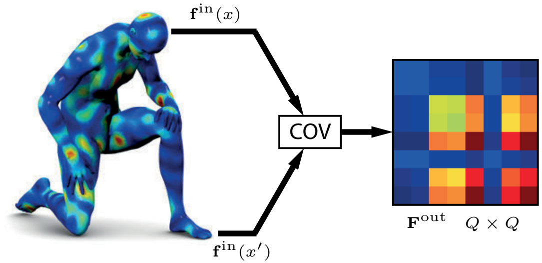
Boscaini, Masci, Rodolà, B 2016

Partial correspondence performance



Methods: Rodolà et al. 2014 (RF); Rodolà et al. 2015 (PFM); Boscaini, Masci, Rodolà, B 2016 (ACNN); data: Cosmo et al. 2016 (SHREC); benchmark: Kim et al. 2011

From local to global features: covariance layer

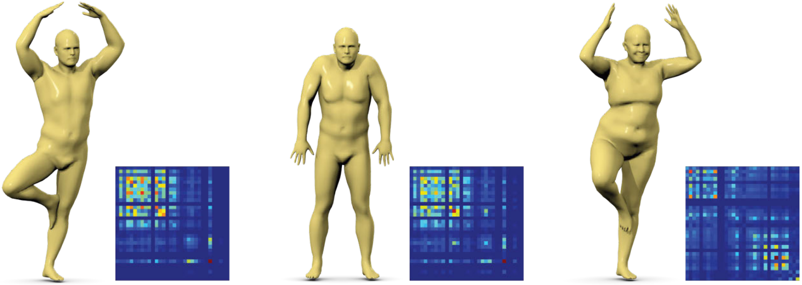


$$
\begin{aligned}
\mathbf{F}^{\mathrm{out}} &= \int_X (\mathbf{f}^{\mathrm{in}}(x) - \boldsymbol{\mu}_{\mathbf{f}^{\mathrm{in}}})(\mathbf{f}^{\mathrm{in}}(x) - \boldsymbol{\mu}_{\mathbf{f}^{\mathrm{in}}})^\top dx \\
\boldsymbol{\mu}_{\mathbf{f}^{\mathrm{in}}} &= \int_X \mathbf{f}_{\mathrm{in}}(x) dx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

- Aggregates local features into a global shape descriptor

Tuzel et al. 2006; Masci, Boscaini, B, Vandergheynst 2015

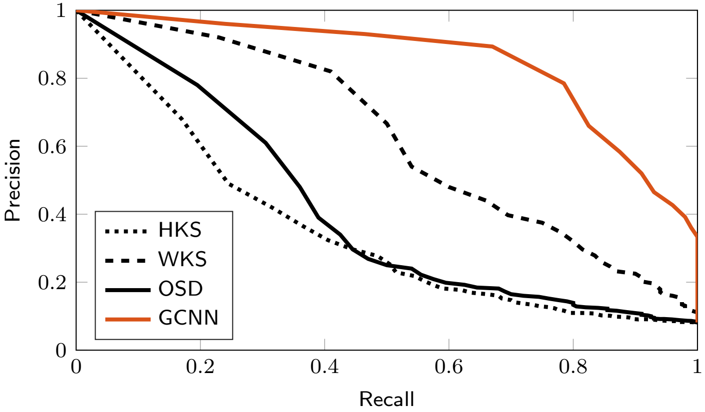
Learning shape similarity



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

Masci, Boscaini, B, Vandergheynst 2015

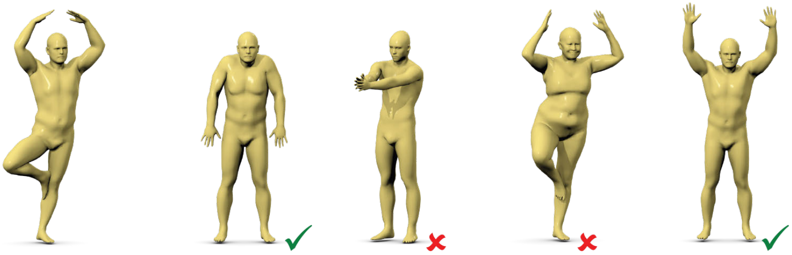
Learned retrieval performance



1-layer GCNN; Training and testing: FAUST

Masci, Boscaini, B, Vandergheynst 2015

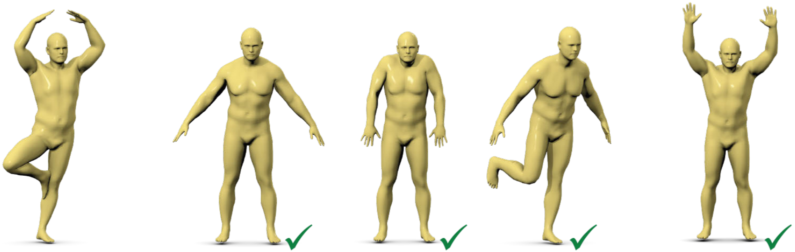
Retrieval examples: HKS



Shape retrieval using similarity computed with HKS

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

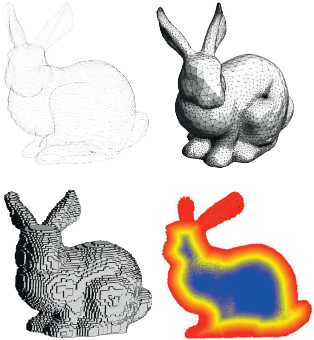
Retrieval examples: GCNN



Shape retrieval using similarity computed with GCNN

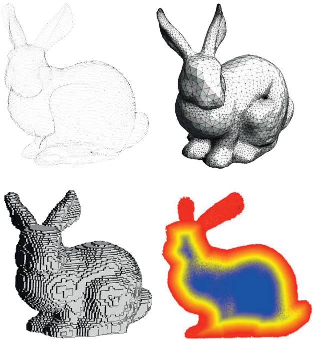
Masci, Boscaini, B, Vandergheynst 2015; data: Pickup et al. 2014

Geometric deep learning: next challenges



3D shapes

Geometric deep learning: next challenges



3D shapes

Graphs

Geometric deep learning: next challenges

Classical deep learning

- Euclidean
- 1D/2D grid structure
- Main ingredient: convolution
- Maps to GPU architecture

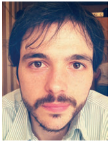
Geometric deep learning

- Non-Euclidean
- No grid structure
- No linear space structure
- No shift-invariance

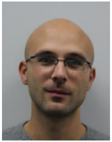
APACHE GIRAPH

signal collect

D. Boscaini

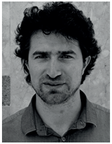


J. Masci

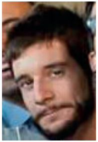


E. Rodolà

P. Vandergheynst



U. Castellani



S. Melzi

D. Cremers

USI

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

UNIVERSITÀ DEGLI STUDI DI VERONA

TUM

Supported by

erc

KTI/CTI
DIE FÖRDERAGENTUR FÜR INNOVATION
L'AGENCE POUR LA PROMOTION DE L'INNOVATION
L'AGENZIA PER LA PROMOZIONE DELL'INNOVAZIONE
THE INNOVATION PROMOTION AGENCY

HASLERSTIFTUNG

Google

NVIDIA

intel

Thank you!