Deep learning on geometric data

Michael Bronstein

University of Lugano

Intel Corporation

Oxford University, 12 October 2015
INVISION  A New Dimension to  Intel

(Acquired by Intel in 2012)
Different form factor computers featuring Intel RealSense 3D camera
Deluge of geometric data

3D sensors

Repositories

3D printers
Applications

Reconstruction

Recognition

Retrieval

Avatars

Virtual dressing

Gesture control

Images: Davison et al. 2011; Zafeiriou et al. 2012; Kim et al. 2013; Faceshift; Fashion3D; Minority report
Basic problems: shape similarity and correspondence
Basic problems: shape similarity and correspondence

Isometric

Non-isometric
Basic problems: shape similarity and correspondence

Isometric
Non-isometric
Different representation
3D feature descriptors

- **SIFT**\(^1\) / MeshHOG\(^2\)
- **MSER**\(^3\) / ShapeMSER\(^4\)
- (Intrinsic\(^6\)) Shape context\(^5\)
- Spin image\(^7\)
- Heat kernel signature\(^8\)

---

Task-specific features

Correspondence
Task-specific features

Correspondence

Similarity

Sivic, Zisserman 2003
Deep learning (r)evolution

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

Very Deep Convolutional Networks for Large-Scale Image Recognition

Karen Simonyan* & Andrew Zisserman+
Visual Geometry Group, Department of Engineering Science, University of Oxford
{karen,az}@robots.ox.ac.uk
3D shapes vs images

- Array of pixels
- Point cloud
- Voxels
- Mesh
- Level set
3D shapes vs images

\[ \frac{1}{2} + \frac{1}{2} = \]
3D shapes vs images

\[
\begin{align*}
\frac{1}{2} & \quad + \quad \frac{1}{2} \quad = \\
\frac{1}{2} & \quad + \quad \frac{1}{2} \quad = ?
\end{align*}
\]
Outline

- Background: Laplacians and spectral analysis on manifolds
- Spectral descriptors (heat- and wave-kernel signatures)
- Convolutional neural networks on manifolds
Outline

- Background: Laplacians and spectral analysis on manifolds
- Spectral descriptors (heat- and wave-kernel signatures)
- Convolutional neural networks on manifolds

Geodesic convolution

Windowed Fourier transform

Anisotropic diffusion
Laplacian in one minute

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

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

Divergence theorem:
\[
\int_V \text{div}(F) \, dV = \int_{\partial V} \langle F, \hat{n} \rangle \, dS
\]

\( \sum \text{sources} + \text{sinks} = \text{net flow} \)

Laplacian \( \Delta f(x) = -\text{div}(\nabla f(x)) \)

Smooth scalar field \( f \)
Laplacian in one minute

- **Gradient** $\nabla f(x) = \text{‘direction of the steepest increase of } f \text{ at } x'$

Smooth scalar field $f$
Laplacian in one minute

- **Gradient** $\nabla f(x) = \text{‘direction of the steepest increase of } f \text{ at } x\text{’}

- **Divergence** $\text{div}(F(x)) = \text{‘density of an outward flux of } F \text{ from an infinitesimal volume around } x\text{’}

Smooth vector field $F$
Laplacian in one minute

- **Gradient** $\nabla f(x) = \text{‘direction of the steepest increase of } f \text{ at } x$’

- **Divergence** $\text{div}(F(x)) = \text{‘density of an outward flux of } F \text{ from an infinitesimal volume around } x$’

**Divergence theorem:**

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

‘$\sum$ sources + sinks = net flow’

Smooth vector field $F$
Laplacian in one minute

- **Gradient** $\nabla f(x) = \text{‘direction of the steepest increase of } f \text{ at } x$’

- **Divergence** $\text{div}(F(x)) = \text{‘density of an outward flux of } F \text{ from an infinitesimal volume around } x\text{’}$

**Divergence theorem:**

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

‘$\sum$ sources + sinks = net flow’

- **Laplacian** $\Delta f(x) = -\text{div}(\nabla f(x))$

  ‘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*
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

\[ c \text{ [m}^2/\text{sec}] = \text{thermal diffusivity constant (assumed = 1)} \]
Riemannian geometry in one minute

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

*We assume manifolds without boundary for simplicity*
Riemannian geometry in one minute

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

- **Riemannian metric**

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

  depending smoothly on $x$

*We assume manifolds without boundary for simplicity*
Riemannian geometry in one minute

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

- **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

*We assume manifolds without boundary for simplicity*
Riemannian geometry in one minute

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

- **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

- **Exponential map**

  \[ \exp_x : T_x X \to X \]

*We assume manifolds without boundary for simplicity*
Riemannian geometry in one minute

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

- **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

- **Exponential map**
  \[
  \exp_x : T_x X \to X
  \]

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

*We assume manifolds without boundary for simplicity*
Laplace-Beltrami operator

\[ \nabla_X f(x) = \nabla (f \circ \exp_x)(0) \]

Taylor expansion

\[ (f \circ \exp_x)(v) \approx f(x) + \langle \nabla_X f(x), v \rangle_{T_xX} \]

Laplace-Beltrami operator

\[ \Delta_X f(x) = \Delta (f \circ \exp_x)(0) \]

Smooth field \( f : X \to \mathbb{R} \)

Isometry-invariant

Self-adjoint

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

Positive semidefinite \( \Rightarrow \) non-negative eigenvalues
Laplace-Beltrami operator

\[
\nabla_X f(x) = \nabla (f \circ \exp_x)(0)
\]

Taylor expansion

\[
(f \circ \exp_x)(v) \approx f(x) + \langle \nabla_X f(x), v \rangle_{T_xX}
\]

Laplace-Beltrami operator

\[
\Delta_X f(x) = \Delta (f \circ \exp_x)(0)
\]

Smooth field \( f \circ \exp_x : T_xX \to \mathbb{R} \)
**Intrinsic gradient**

\[ \nabla_X f(x) = \nabla (f \circ \exp_x)(0) \]

**Taylor expansion**

\[
(f \circ \exp_x)(v) \approx f(x) + \langle \nabla_X f(x), v \rangle_{T_x X}
\]
Laplace-Beltrami operator

- **Intrinsic gradient**
  \[ \nabla_X f(x) = \nabla(f \circ \exp_x)(0) \]

- **Taylor expansion**
  \[ (f \circ \exp_x)(v) \approx f(x) + \langle \nabla_X f(x), v \rangle_{T_x X} \]

- **Laplace-Beltrami operator**
  \[ \Delta_X f(x) = \Delta(f \circ \exp_x)(0) \]
Laplace-Beltrami operator

- **Intrinsic gradient**
  \[ \nabla_X f(x) = \nabla(f \circ \exp_x)(0) \]

  Taylor expansion
  \[ (f \circ \exp_x)(v) \approx f(x) + \langle \nabla_X f(x), v \rangle_{T_xX} \]

- **Laplace-Beltrami operator**
  \[ \Delta_X f(x) = \Delta(f \circ \exp_x)(0) \]

- **Intrinsic** (expressed solely in terms of the Riemannian metric)
Laplace-Beltrami operator

- **Intrinsic gradient**
  \[ \nabla_X f(x) = \nabla(f \circ \exp_x)(0) \]

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**
  \[ \Delta_X f(x) = \Delta(f \circ \exp_x)(0) \]

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

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

Positive semidefinite \( \Rightarrow \) non-negative eigenvalues
Laplace-Beltrami operator

- **Intrinsic gradient**
  \[
  \nabla_X f(x) = \nabla (f \circ \exp_x)(0)
  \]

- **Taylor expansion**
  \[
  (f \circ \exp_x)(v) \approx f(x) + \langle \nabla_X f(x), v \rangle_{T_x X}
  \]

- **Laplace-Beltrami operator**
  \[
  \Delta_X f(x) = \Delta (f \circ \exp_x)(0)
  \]

- **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**
  \[ \nabla_X f(x) = \nabla(f \circ \exp_x)(0) \]

- **Taylor expansion**
  \[
  (f \circ \exp_x)(v) \approx f(x) + \langle \nabla_X f(x), v \rangle_{T_x X}
  \]

- **Laplace-Beltrami operator**
  \[ \Delta_X f(x) = \Delta(f \circ \exp_x)(0) \]

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

- **Intrinsic gradient**
  \[ \nabla_X f(x) = \nabla(f \circ \exp_x)(0) \]
  Taylor expansion
  \[ (f \circ \exp_x)(v) \approx f(x) + \langle \nabla_X f(x), v \rangle_{T_x X} \]

- **Laplace-Beltrami operator**
  \[ \Delta_X f(x) = \Delta(f \circ \exp_x)(0) \]

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

One-dimensional

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

Two-dimensional

\[(\Delta f)_{ij} \approx 4f_{ij} - f_{i-1,j} - f_{i+1,j} - f_{i,j-1} - f_{i,j+1}\]
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 \((V, E, F)\)

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

\(a_i = \text{local area element}\)

Tutte 1963; MacNeal 1949; Duffin 1959; Pinkall, Polthier 1993
A function $f : [-\pi, \pi] \to \mathbb{R}$ can be written as Fourier series

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

$$= \alpha_1 + \alpha_2 + \alpha_3 + \ldots$$
A function $f : [-\pi, \pi] \rightarrow \mathbb{R}$ can be written as Fourier series

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

$$\hat{f}(\omega) = \langle f, e^{-i\omega x} \rangle_{L^2([-\pi, \pi])}$$

$$= \alpha_1 \quad + \quad \alpha_2 \quad + \quad \alpha_3 \quad + \ldots$$
A function $f : [-\pi, \pi] \to \mathbb{R}$ can be written as Fourier series

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

$$\hat{f}(\omega) = \langle f, e^{-i\omega x} \rangle_{L^2([-\pi, \pi])}$$

$$= \alpha_1 + \alpha_2 + \alpha_3 + \ldots$$

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

$$f(x) = \sum_{k \geq 1} \int_X f(\xi) \phi_k(\xi) d\xi \phi_k(x)$$

$$\hat{f}_k = \langle f, \phi_k \rangle_{L^2(X)}$$

$$f = \alpha_1 + \alpha_2 + \alpha_3 + \ldots$$

Fourier basis = Laplacian eigenfunctions: $\Delta_X \phi_k(x) = \lambda_k \phi_k(x)$
Heat diffusion on manifolds

\[
\begin{aligned}
  f_t(x, t) &= -\Delta_X f(x, t) \\
  f(x, 0) &= f_0(x)
\end{aligned}
\]

- \( f(x, t) \) = amount of heat at point \( x \) at time \( t \)
- \( f_0(x) \) = initial heat distribution
Heat diffusion on manifolds

\[
\begin{align*}
\frac{\partial f(x, t)}{\partial t} &= -\Delta_X f(x, t) \\
f(x, 0) &= f_0(x)
\end{align*}
\]

- \( f(x, t) \) = 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

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

\[
= \int_X f_0(\xi) \sum_{k \geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(\xi) \, d\xi
\]
Heat diffusion on manifolds

\[ \begin{align*}
  f_t(x, t) &= -\Delta_X f(x, t) \\
  f(x, 0) &= f_0(x)
\end{align*} \]

- \( f(x, t) \) = 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

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

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

heat kernel \( h_t(x, \xi) \)
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**

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

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

- “impulse response” to a delta-function at \( \xi \)
- heat kernel \( h_t(x, \xi) \)
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

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

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

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

\[ f(x) = \sum_{k \geq 1} \left( \begin{array}{c} \tau_1(\lambda_k) \\ \vdots \\ \tau_Q(\lambda_k) \end{array} \right) \Phi_k^2(x) \]

Heat Kernel Signature (HKS)

Wave Kernel Signature (WKS)

Low-pass filter bank

\[ \tau_i(\lambda) = \exp(-\lambda t_i) \]

Heat autodiffusivity

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
Optimal spectral descriptors

\[ f_\tau(x) = \sum_{k \geq 1} \left( \begin{array}{c} \tau_1(\lambda_k) \\ \vdots \\ \tau_Q(\lambda_k) \end{array} \right) \phi_k^2(x) \]

parametrized by frequency responses \( \tau(\lambda) = (\tau_1(\lambda), \ldots, \tau_Q(\lambda))^T \)
Optimal spectral descriptors

\[ f_A(x) = \sum_{k \geq 1} A \begin{pmatrix} \beta_1(\lambda_k) \\ \vdots \\ \beta_M(\lambda_k) \end{pmatrix} \phi_k^2(x) \]

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

Litman, Bronstein 2014
Optimal spectral descriptors

\[ f_A(x) = A \sum_{k \geq 1} \left( \begin{array}{c} \beta_1(\lambda_k) \\ \vdots \\ \beta_M(\lambda_k) \end{array} \right) \phi_k^2(x) \]

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

Litman, Bronstein 2014
Optimal spectral descriptors

\[ f_A(x) = A \sum_{k \geq 1} \left( \begin{array}{c} \beta_1(\lambda_k) \\ \vdots \\ \beta_M(\lambda_k) \end{array} \right) \phi_k^2(x) = A g(x) \]

parametrized by linear combination coefficients \( A \) of geometry vectors
\[ g(x) = (g_1(x), \ldots, g_M(x))^\top \]

Litman, Bronstein 2014
Optimal spectral descriptors

\[ f_A(x) = A \sum_{k \geq 1} \begin{pmatrix} \beta_1(\lambda_k) \\ \vdots \\ \beta_M(\lambda_k) \end{pmatrix} \phi_k^2(x) = Ag(x) \]

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

- Optimal \( A \) 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

\[
\mathbf{f}_A(x) = A \sum_{k \geq 1} \begin{pmatrix}
\beta_1(\lambda_k) \\
\vdots \\
\beta_M(\lambda_k)
\end{pmatrix} \phi^2_k(x) = A \mathbf{g}(x)
\]

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

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

Litman, Bronstein 2014
Optimal spectral descriptors

\[ f_A(x) = A \sum_{k \geq 1} \begin{pmatrix} \beta_1(\lambda_k) \\ \vdots \\ \beta_M(\lambda_k) \end{pmatrix} \phi_k^2(x) = Ag(x) \]

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

- Optimal \( A \) 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!

Litman, Bronstein 2014
Convolutional neural networks

- Combination of convolution and pooling layers

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

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
Convolution
Geodesic convolution

Windowed Fourier transform

Anisotropic diffusion
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 $\theta$

Kokkinos, B², Litman 2012
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

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

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

Kokkinos, B², Litman 2012
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
  \[
  \Omega(x) : B_{\rho_0}(x) \to [0, \rho_0] \times [0, 2\pi)
  \]
  from manifold to local coordinates $(\rho, \theta)$ around $x$

- Patch operator applied to $f \in L^2(X)$
  \[
  (D(x)f)(\rho, \theta) = (f \circ \Omega^{-1}(x))(\rho, \theta)
  \]

Kokkinos, B², Litman 2012
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

\[ v_\rho(x, \xi) \propto e^{-\left( \sum_{\xi}^\rho (x, \xi) - \rho \right)^2 / \sigma_\rho^2} \]

Angular weight

\[ v_\theta(x, \xi) \propto e^{-\sum_{\xi}^\theta (\Gamma(x, \theta), \xi) / \sigma_\theta^2} \]

Kokkinos, B², Litman 2012
Geodesic convolution

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

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

Masci, Boscaiini, B, Vandergheynst 2015
**Geodesic convolution**

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

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

---

Masci, Boscahin, B, Vandergheynst 2015
Geodesic convolution

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

\[
(f * a)(x) = \sum_{\theta, r} (D(x)f)(r, \theta) \ a(\theta + \Delta \theta, r)
\]

- Angular coordinate origin is arbitrary = rotation ambiguity!

Masci, Boscaini, B, Vandergheynst 2015
**Geodesic convolution**

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

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

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

Masci, Boscaini, B, Vandergheynst 2015
Geodesic Convolution layer

\[ f^\text{out}_{\Delta \theta, q}(x) = \sum_{p=1}^{P} (f_p \ast a_{\Delta \theta, qp})(x), \quad q = 1, \ldots, Q \]

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

Masci, Boscaini, B, Vandergheynst 2015
Geodesic Convolution layer

\[ f_{q_{out}}(x) = \max_{\Delta \theta} \sum_{p=1}^{P} (f_p \star a_{\Delta \theta, qp})(x), \quad q = 1, \ldots, Q \]

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

Masci, Boscaini, B, Vandergheynst 2015
Toy ShapeNet architecture

Masci, Boscaini, B, 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 $\Theta$

$$
\ell(\Theta) = (1 - \gamma) \sum_{(x,x^+) \in \mathcal{T}^+} \| f_\Theta(x) - f_\Theta(x^+) \|
+ \gamma \sum_{(x,x^-) \in \mathcal{T}^-} \max\{\mu - \| f_\Theta(x) - f_\Theta(x^-) \|, 0\}
$$

Masci, Boscaini, B, Vanderghyeynst 2015
ShapeNet descriptor robustness

HKS descriptor distance

Masci, Boscaini, B, Vanderghynst 2015; data: B et al. 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)
ShapeNet descriptor robustness

Masci, Boscaini, B, Vanderheynst 2015; data: B et al. 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)
ShapeNet descriptor robustness

Masci, Boscaini, B, Vanderheynst 2015; data: B et al. 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)
ShapeNet descriptor robustness

Masci, Boscaioni, B, Vandergheynst 2015; data: B et al. 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)
ShapeNet 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, B, Vanderheynst 2015 (ShapeNet); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011
ShapeNet 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, B, Vandergheynst 2015 (ShapeNet); data: Bogo et al. 2014 (FAUST); B et al. 2008 (TOSCA); benchmark: Kim et al. 2011
Learning shape correspondence with ShapeNet

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

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

Rodolà et al. 2014; Masci, Boscaini, B, Vanderghynst 2015
ShapeNet correspondence performance

3-layer ShapeNet; Training and testing: FAUST

Masci, Boscaini, B, Vandergheynst 2015; Rodolà et al. 2014; Kim et al. 11
ShapeNet correspondence examples

Correspondence found using ShapeNet
(similar colors encode corresponding points)

Boscaini, Masci, Rodolà, B, Cremers 2015
From local to global features: covariance layer

\[
F_{\text{out}} = \int_X (f_{\text{in}}(x) - \mu_{f_{\text{in}}})(f_{\text{in}}(x) - \mu_{f_{\text{in}}})^\top dx
\]

\[
\mu_{f_{\text{in}}} = \int_X f_{\text{in}}(x) dx
\]

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

Tuzel et al. 2006; Masci, Boscaini, B, Vandergheynst 2015
Global shape descriptor using covariance layer in ShapeNet $F_\Theta$

As similar as possible on positives $\mathcal{T}^+$

As dissimilar as possible on negatives $\mathcal{T}^-$

Minimize siamese loss w.r.t. ShapeNet parameters $\Theta$

Rodolà et al. 2014; Masci, Boscaini, B, Vandergheynst 2015
ShapeNet retrieval performance

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: ShapeNet

Shape retrieval using similarity computed with ShapeNet

Masci, Boscai, B, Vandergeynst 2015; data: Pickup et al. 2014
Geodesic convolution

Windowed Fourier transform

Anisotropic diffusion
Uncertainty principle

Spatial localization $\times$ Frequency localization $= \text{const}$
Uncertainty principle

Spatial localization $\times$ Frequency localization $= \text{const}$
Uncertainty principle

Spatial localization $\times$ Frequency localization $= \text{const}$
Windowed Fourier transform (WFT)

Localize Fourier transform in a window

\[
\text{WFT}(f(x))(\xi, \omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) w(x - \xi) e^{-i\omega x} \, dx
\]
Windowed Fourier transform (WFT)

Localize Fourier transform in a window

\[
\text{WFT}(f(x))(\xi, \omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) w(x - \xi) e^{-i\omega x} \, dx
\]
Windowed Fourier transform (WFT)

Localize Fourier transform in a window

\[
\text{WFT}(f(x))(\xi, \omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) w(x - \xi) e^{-i\omega x} \, dx
\]
Windowed Fourier transform (WFT)

Localize Fourier transform in a window

$$\text{WFT}(f(x))(\xi, \omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) w(x - \xi) e^{-i\omega x} \, dx$$
Windowed Fourier transform (WFT)

Localize Fourier transform in a window

\[
\text{WFT}(f(x))(\xi, \omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) w(x - \xi) e^{-i\omega x} \, dx
\]

where \( w(x) \) is the window function.

Graphical representation showing the effect of different windows on the Fourier transform.
Windowed Fourier transform (WFT)

Localize Fourier transform in a window

\[
\text{WFT}(f(x))(\xi, \omega) = \langle f, g_{\xi,\omega} \rangle_{L^2([\pi, \pi])}
\]
Windowed Fourier transform on manifolds

**Translation**: convolution with delta

\[(T_{x'} f)(x) = (f * \delta_{x'})(x)\]

Shuman et al. 2014
**Windowed Fourier transform on manifolds**

**Translation**: convolution with delta

\[
(T_{x'} f)(x) = (f \ast \delta_{x'})(x) = \sum_{k \geq 1} \langle f, \phi_k \rangle_{L^2(X)} \langle \delta_{x'}, \phi_k \rangle_{L^2(X)} \phi_k(x)
\]

Shuman et al. 2014
Windowed Fourier transform on manifolds

**Translation**: convolution with delta

\[
(T_{x'} f)(x) = (f \ast \delta_{x'})(x) = \sum_{k \geq 1} \langle f, \phi_k \rangle_{L^2(X)} \langle \delta_{x'}, \phi_k \rangle_{L^2(X)} \phi_k(x)
\]

\[
= \sum_{k \geq 1} \hat{f}_k \phi_k(x') \phi_k(x)
\]

Shuman et al. 2014
**Windowed Fourier transform on manifolds**

**Translation**: convolution with delta

\[
(T_{x'} f)(x) = (f * \delta_{x'})(x) = \sum_{k \geq 1} \langle f, \phi_k \rangle_{L^2(X)} \langle \delta_{x'}, \phi_k \rangle_{L^2(X)} \phi_k(x)
\]

\[
= \sum_{k \geq 1} \hat{f}_k \phi_k(x') \phi_k(x)
\]

**Modulation**: multiplication by basis function

\[
(M_k f)(x) = f(x) \phi_k(x)
\]

Shuman et al. 2014
**Translation:** convolution with delta

\[(T_{x'} f)(x) = (f \ast \delta_{x'})(x) = \sum_{k \geq 1} \langle f, \phi_k \rangle_{L^2(X)} \langle \delta_{x'}, \phi_k \rangle_{L^2(X)} \phi_k(x)\]

\[= \sum_{k \geq 1} \hat{f}_k \phi_k(x') \phi_k(x)\]

**Modulation:** multiplication by basis function

\[(M_k f)(x) = f(x) \phi_k(x)\]

**Windowed Fourier transform:**

\[(S f)_{x,k} = \langle f, M_k T_x g \rangle_{L^2(X)} = \sum_{l \geq 1} \hat{g}_l \phi_l(x) \langle f, \phi_l \phi_k \rangle_{L^2(X)}\]

Shuman et al. 2014
Examples of WFT atoms $g_{x,k}$ for different windows $\hat{g}$

$$g_{x',k}(x) = (M_k T_{x'} g)(x) = \phi_k(x) \sum_{l \geq 1} \hat{g}_l \phi_l(x') \phi_l(x)$$

Boscaini, Masci, Melzi, B, Castellani, Vandergheyden 2015
Learning WFT windows

\[ f_q^{\text{out}}(x) = \sum_{p=1}^{P} \sum_{k=1}^{K} a_{qpk} |(S f_p^{\text{in}})_{x,k}|, \quad q = 1, \ldots, Q \]

where for each input dimension WFT uses a different window

\[ (S f_p^{\text{in}})_{x,k} = \sum_{l \geq 1} \gamma_p(\lambda_l) \phi_l(x) \langle f_p^{\text{in}}, \phi_l \phi_k \rangle_{L^2(X)}, \quad p = 1, \ldots, P \]

- Learn window for each input dimension (coefficients \( b_{pm} \))

Boscaini, Masci, Melzi, B, Castellani, Vandergheynst 2015
Learning WFT windows

\[ f_{q}^{\text{out}}(x) = \sum_{p=1}^{P} \sum_{k=1}^{K} a_{qpk} |(S f_{p}^{\text{in}})_{x,k}|, \quad q = 1, \ldots, Q \]

where for each input dimension WFT uses a different window

\[ (S f_{p}^{\text{in}})_{x,k} = \sum_{l \geq 1} \gamma_{p}(\lambda_{l}) \left\langle \phi_{l}(x), f_{p}^{\text{in}}, \phi_{l}\phi_{k} \right\rangle_{L^{2}(X)}, \quad p = 1, \ldots, P \]

- Learn window for each input dimension (coefficients \( b_{pm} \))
- Learn bank of filters for each WFT (coefficients \( a_{qpk} \))
Example of learned WFT windows

WFT windows $\gamma_p(\lambda)$ learned on FAUST dataset

Boscaini, Masci, Melzi, B, Castellani, Vandergheynst 2015
Localized Spectral CNN (LSCNN) descriptors

Example of learned LSCNN shape descriptors

Boscaini, Masci, Melzi, B, Castellani, Vandergheynst 2015
LSCNN descriptor robustness: point clouds

LSCNN descriptor distance

Boscaini, Masci, Melzi, B, Castellani, Vandergheynst 2015
LSCNN 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, B, Vanderheynst 2015 (ShapeNet); Boscaini, Masci, Melzi, B, Castellani, Vanderheynst 2015 (NN, LSCNN); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011
Geodesic convolution

Windowed Fourier transform

Anisotropic diffusion
Homogeneous diffusion

\[ f_t(x) = -\text{div}(c \nabla f(x)) \]

\( c = \text{thermal diffusivity constant} \) describing heat conduction properties of the material (diffusion speed is equal everywhere)
Anisotropic diffusion

\[ f_t(x) = - \text{div}(A(x) \nabla f(x)) \]

\( A(x) = \text{heat conductivity tensor} \) describing heat conduction properties of the material (diffusion speed is position + direction dependent)
Anisotropic diffusion on manifolds

\[ f_t(x) = -\text{div}_X \left( R_\theta \begin{pmatrix} \alpha \\ 1 \end{pmatrix} R_\theta^T \nabla_X f(x) \right) \]

Andreux et al. 2014; Boscaini, Masci, Rodolà, B, Cremers 2015
Anisotropic diffusion on manifolds

\[ f_t(x) = -\text{div}_X \left( D_{\alpha\theta}(x) \nabla_X f(x) \right) \]

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

Andreux et al. 2014; Boscaini, Masci, Rodolà, B, 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 \( \alpha \)

Boscaini, Masci, Rodolà, B, Cremers 2015
Isotropic vs Anisotropic HKS

Boscaini, Masci, Rodolà, B, Cremers 2015
Anisotropic Diffusion Descriptor (ADD) robustness

Anisotropic Diffusion Descriptor (ADD) distance

Boscaini, Masci, Rodolà, B, Cremers 2015
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, B, Vandergheynst 2015 (ShapeNet); Boscaini, Masci, Melzi, B, Castellani, Vandergheynst 2015 (LSCNN); Boscaini, Masci, Rodolà, B, 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, B, Vandergheynst 2015 (ShapeNet); Boscaini, Masci, Melzi, B, Castellani, Vandergheynst 2015 (LSCNN); Boscaini, Masci, Rodolà, B, Cremers 2015 (ADD); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011
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!