Master SIF - REP (Part 9)
Advanced tools for Digital Image Processing II

Thomas Maugey
thomas.maugey@inria.fr

Université de Rennes

Inria

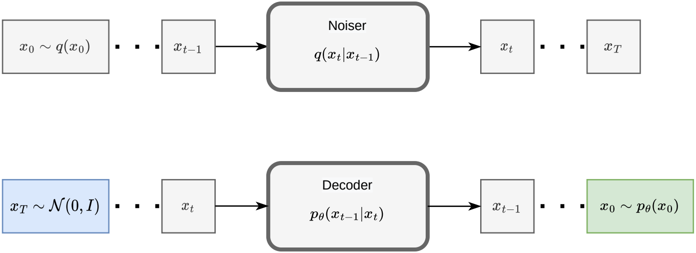
Fall 2023

Table of Contents

Generative diffusion models

General concept

Train a denoiser and use it to generate realistic images



[SONG, Jiaming, MENG, Chenlin, et ERMON, Stefano. Denoising Diffusion Implicit Models. In : International Conference on Learning Representations. 2020.]

Forward diffusion

Natural images follow a probability distribution, denoted by

$$x \sim q(x)$$

Forward diffusion process consists in adding noise to original data, gradually switching from q to a normal distribution:

$$\mathbf{x}_0 \sim q(x) \quad (1)$$

$$\mathbf{x}_t|\mathbf{x}_{t-1} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad (2)$$

$$\mathbf{x}_t|\mathbf{x}_0 \sim \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}_0; \sqrt{1-\alpha_t}\mathbf{I}) \quad (3)$$

With α_t going from 0 at 1 when t goes from T at 0. We thus obtain $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$.

Training

The role of the diffusion model is to be able to estimate $\mathbf{x}_{t-1}$ from the pair $(\mathbf{x}_t, t)$. We note the model ϵ_θ.

while *not converged* **do**

$\mathbf{x}_0 \sim q(\mathbf{x}_0)$;

$t \sim U(1, ..., T)$;

$\epsilon \sim \mathcal{N}(0, \mathbf{I})$;

`Gradient descent step on:`

$$\nabla_\theta \left\| \epsilon - \epsilon_\theta(\sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1 - \alpha_t}\epsilon, t) \right\|^2$$

end

Backward Sampling

Using the diffusion model ϵ_θ, we can predict $\mathbf{x}_{t-1}$:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\alpha_t}} + \sqrt{1-\alpha_{t-1}}\epsilon_\theta(\mathbf{x}_t, t) \tag{4}$$

We can estimate the current image at the step t that we note $x_{0|t}$ with:

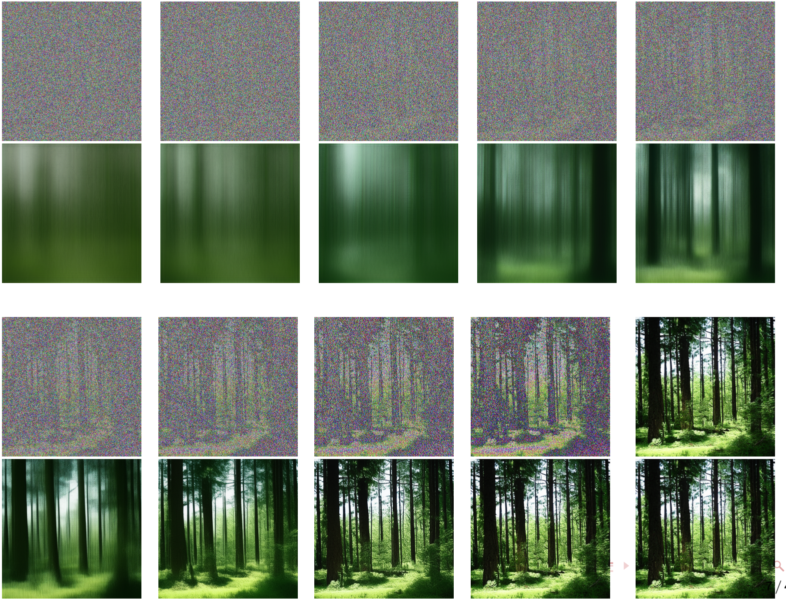
$$\mathbf{x}_{0|t}(\epsilon_\theta(\mathbf{x}_t, t)) = \frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\alpha_t}} \tag{5}$$

and thus we can rewrite:

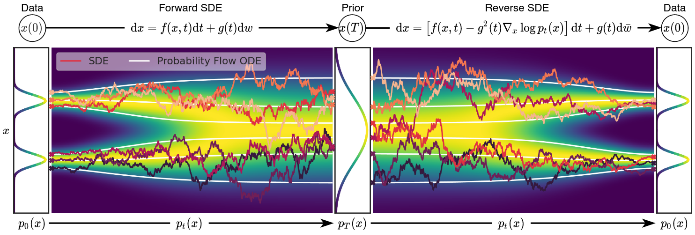
$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}x_{0|t}(\epsilon_\theta(\mathbf{x}_t, t)) + \sqrt{1-\alpha_{t-1}}\epsilon_\theta(\mathbf{x}_t, t) \tag{6}$$

$\mathbf{x}_{t-1}$ becomes a combination between, the estimated sample $x_{0|t}$ and what is commonly called the direction pointing to x_t

Illustration



Illustration



Guided diffusion models

Given a criterion σ, a diffusion model can be guided:

$$\tilde{\epsilon_\theta}(\mathbf{x}_t, t) = \epsilon_\theta(\mathbf{x}_t, t) + \lambda(t)\nabla_{\mathbf{x}_t}\mathcal{L}(\mathbf{x}_{0|t}(\epsilon_\theta(\mathbf{x}_t, t)), \sigma) \tag{7}$$

[ ]

Guided diffusion models

Given a criterion σ, a diffusion model can be guided:

$$\tilde{\epsilon_\theta}(\mathbf{x}_t, t) = \epsilon_\theta(\mathbf{x}_t, t) + \lambda(t)\nabla_{\mathbf{x}_t}\mathcal{L}(\mathbf{x}_{0|t}(\epsilon_\theta(\mathbf{x}_t, t)), \sigma) \quad (7)$$

Application to compression:

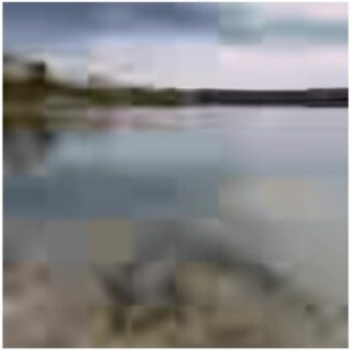
[T. Bordin, T. Maugey Semantic based image generative compression, IEEE MMSP 2023]

Guided diffusion models

Given a criterion σ, a diffusion model can be guided:

$$\tilde{\epsilon_\theta}(\mathbf{x}_t, t) = \epsilon_\theta(\mathbf{x}_t, t) + \lambda(t)\nabla_{\mathbf{x}_t}\mathcal{L}(\mathbf{x}_{0|t}(\epsilon_\theta(\mathbf{x}_t, t)), \sigma) \tag{7}$$

Application to compression:



[T. Bordin, T. Maugey Semantic based image generative compression, IEEE MMSP 2023]

Guided diffusion models

Given a criterion σ, a diffusion model can be guided:

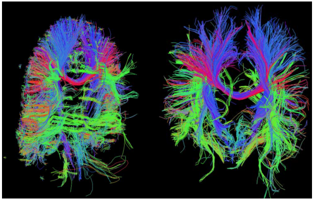
$$\tilde{\epsilon_\theta}(\mathbf{x}_t, t) = \epsilon_\theta(\mathbf{x}_t, t) + \lambda(t)\nabla_{\mathbf{x}_t}\mathcal{L}(\mathbf{x}_{0|t}(\epsilon_\theta(\mathbf{x}_t, t)), \sigma) \qquad (7)$$

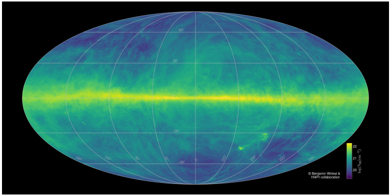
Application to compression:

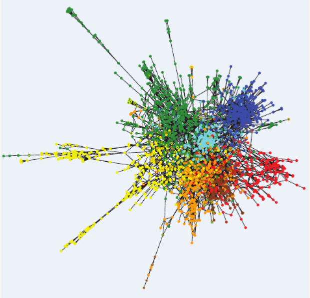
[T. Bordin, T. Maugey Semantic based image generative compression, IEEE MMSP 2023]

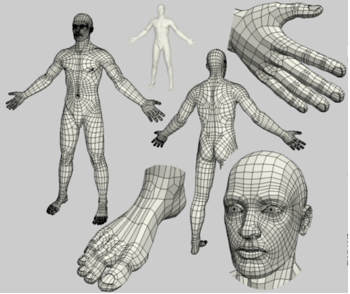
Table of Contents

Image lying on irregular domain



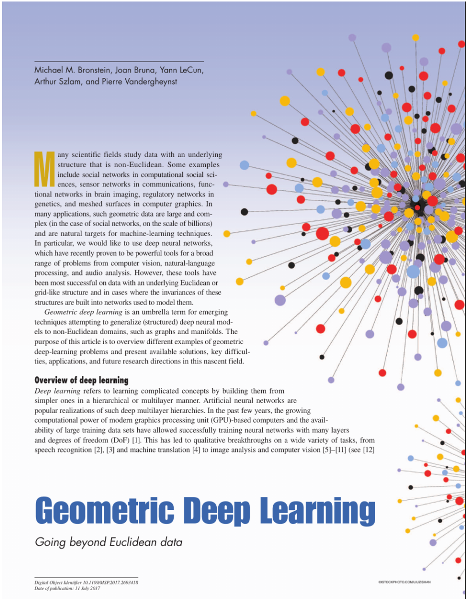






How to perform Deep learning on such data ?

A deep review of the field



Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst

Many scientific fields study data with an underlying structure that is non-Euclidean. Some examples include social networks in computational social sciences, sensor networks in communications, functional networks in brain imaging, regulatory networks in genetics, and meshed surfaces in computer graphics. In many applications, such geometric data are large and complex (in the case of social networks, on the scale of billions) and are natural targets for machine-learning techniques. In particular, we would like to use deep neural networks, which have recently proven to be powerful tools for a broad range of problems from computer vision, natural-language processing, and audio analysis. However, these tools have been most successful on data with an underlying Euclidean or grid-like structure and in cases where the invariances of these structures are built into networks used to model them.

Geometric deep learning is an umbrella term for emerging techniques attempting to generalize (structured) deep neural models to non-Euclidean domains, such as graphs and manifolds. The purpose of this article is to overview different examples of geometric deep-learning problems and present available solutions, key difficulties, applications, and future research directions in this nascent field.

Overview of deep learning

Deep learning refers to learning complicated concepts by building them from simpler ones in a hierarchical or multilayer manner. Artificial neural networks are popular realizations of such deep multilayer hierarchies. In the past few years, the growing computational power of modern graphics processing unit (GPU)-based computers and the availability of large training data sets have allowed successfully training neural networks with many layers and degrees of freedom (DoF) [1]. This has led to qualitative breakthroughs on a wide variety of tasks, from speech recognition [2], [3] and machine translation [4] to image analysis and computer vision [5]–[11] (see [12]

Geometric Deep Learning

Going beyond Euclidean data

Digital Object Identifier 10.1109/MSP.2017.2693418
Date of publication: 11 July 2017

CNN on graphs, what's the problem ?

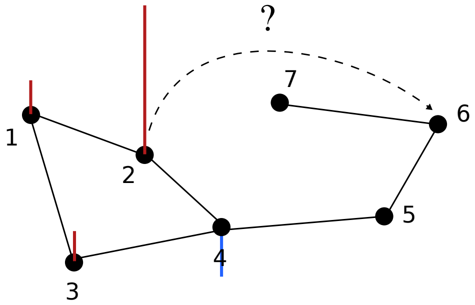
CNN $\rightarrow$ Convolution $\rightarrow$

CNN on graphs, what's the problem ?

CNN $\rightarrow$ Convolution $\rightarrow$ Translation

CNN on graphs, what's the problem ?

CNN → Convolution → Translation



CNN on graphs, what's the problem ?

CNN $\rightarrow$

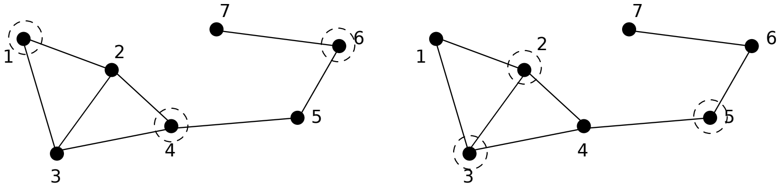
CNN on graphs, what's the problem ?

CNN $\rightarrow$ Pooling (downsampling)

CNN on graphs, what's the problem ?

CNN → Pooling (downsampling)

50% of the nodes ?



CNN on graphs, what's the problem ?

CNN $\rightarrow$

CNN on graphs, what's the problem ?

CNN $\rightarrow$ Fixed grid

CNN on graphs, what's the problem ?

CNN $\rightarrow$ Fixed grid

From left to right: a signal, frequency-domain edge detection, same detection applied when the topology slightly changes

[Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017). Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine, 34(4), 18-42.]

Table of Contents

Spectral definitions

Filtering

In classical signal processing:

In graph signal processing:

[D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83-98, May 2013.]

Filtering

In classical signal processing:

$$\hat{f}_{\text{out}}(\omega) = \hat{f}_{\text{in}}(\omega)\hat{h}(\omega)$$

In graph signal processing:

[D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83-98, May 2013.]

Filtering

In classical signal processing:

$$\hat{f}_{\mathrm{out}}(\omega) = \hat{f}_{\mathrm{in}}(\omega)\hat{h}(\omega)$$

In graph signal processing:

$$\hat{f}_{\mathrm{out}}(\lambda_l) = \hat{f}_{\mathrm{in}}(\lambda_l)\hat{h}(\lambda_l)$$

which gives

$$f_{\mathrm{out}}(n) = \sum_{l=0}^{N-1} \hat{f}_{\mathrm{in}}(\lambda_l)\hat{h}(\lambda_l)u_l(n)$$

it can also be written as

$$\mathbf{f}_{\mathrm{out}} = \hat{h}(\mathbf{L})\mathbf{f}_{\mathrm{in}}, \quad \text{with} \quad \hat{h}(\mathbf{L}) = \mathbf{U}\begin{bmatrix} \hat{h}(\lambda_0) & & 0 \\ & \ddots & \\ 0 & & \hat{h}(\lambda_{N-1}) \end{bmatrix}\mathbf{U}^\top$$

[D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83-98, May 2013.]

Convolution

In classical signal processing:

In graph signal processing:

[D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83-98, May 2013.]

Spectral definitions

Convolution

In classical signal processing:

$$(f * h)(t) = \int_{\mathbb{R}} f(\tau)h(t-\tau)d\tau$$

which can be written as

$$(f * h)(t) = \int_{\mathbb{R}} \hat{f}(\omega)\hat{h}(\omega)e^{2i\pi\omega t}d\omega$$

In graph signal processing:

[D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83-98, May 2013.]

Spectral definitions

Convolution

In classical signal processing:

$$(f * h)(t) = \int_{\mathbb{R}} f(\tau)h(t-\tau)d\tau$$

which can be written as

$$(f * h)(t) = \int_{\mathbb{R}} \hat{f}(\omega)\hat{h}(\omega)e^{2i\pi\omega t}d\omega$$

In graph signal processing:

$$(f * h)(n) := \sum_{l=0}^{N-1} \hat{f}(\lambda_l)\hat{h}(\lambda_l)u_l(n)$$

[D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83-98, May 2013.]

Translation

In classical signal processing:

In graph signal processing:

[D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83-98, May 2013.]

Translation

In classical signal processing:

$$(\mathcal{T}_\tau f)(t) = f(t-\tau)$$

which can be written as

$$(\mathcal{T}_\tau f)(t) = (f * \delta_\tau)(t)$$

In graph signal processing:

[D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83-98, May 2013.]

Spectral definitions

Translation

In classical signal processing:

$$(\mathcal{T}_\tau f)(t) = f(t-\tau)$$

which can be written as

$$(\mathcal{T}_\tau f)(t) = (f * \delta_\tau)(t)$$

In graph signal processing:

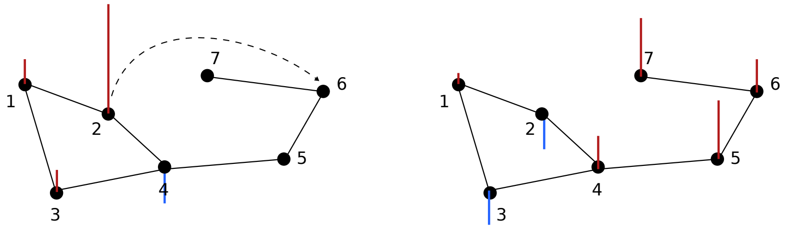
$$(\mathcal{T}_k f)(n) := \sqrt{N}(f * \delta_k)(n)$$

which becomes

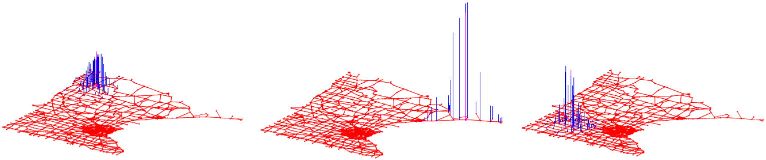
$$(\mathcal{T}_k f)(n) = \sqrt{N}\sum_{l=0}^{N-1} \hat{f}(\lambda_l)u_l(k)u_l(n)$$

[D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83-98, May 2013.]

Spectral definitions, a good solution?

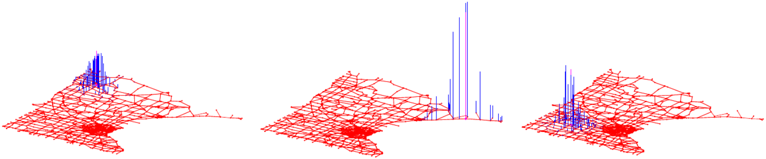


Spectral definitions, a good solution?



[D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83-98, May 2013.]

Spectral definitions, a good solution?



The spectrum is the same, however, the spatial shape is different. It can be a problem

[D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83-98, May 2013.]

Recall of Shannon-Nyquist theorem

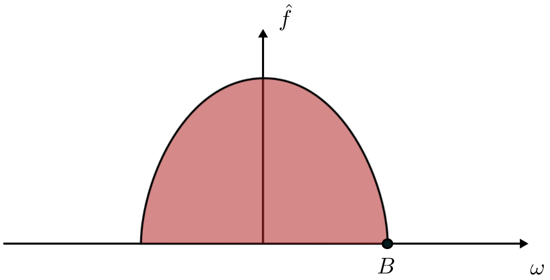
Let us consider a signal f that contains no frequencies higher than B:

$$\forall \omega, \quad \text{s.t. } |\omega| > B, \quad \text{then} \quad \hat{f}(\omega) = 0.$$

Recall of Shannon-Nyquist theorem

Let us consider a signal f that contains no frequencies higher than B:

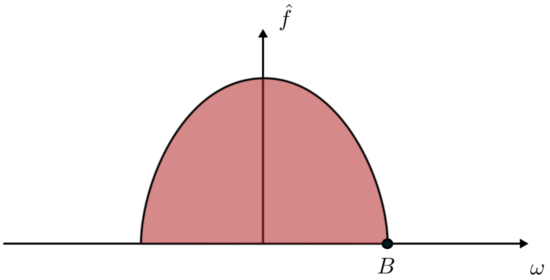
$$\forall \omega, \quad \text{s.t. } |\omega| > B, \quad \text{then} \quad \hat{f}(\omega) = 0.$$



Recall of Shannon-Nyquist theorem

Let us consider a signal f that contains no frequencies higher than B:

$$\forall \omega, \quad \text{s.t. } |\omega| > B, \quad \text{then} \quad \hat{f}(\omega) = 0.$$



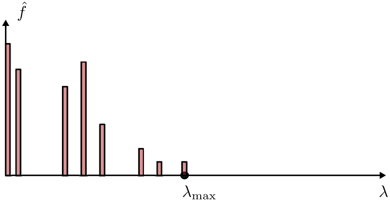
This signal can be sampled at a frequency of $2B$ and fully recovered.

Extension to graphs

Let us consider a signal $\mathbf{f}$ defined on a graph $\mathcal{G}$ that is bandlimited with a bandwidth $\lambda_{\max}$:

$$\forall \ \lambda_l > \lambda_{\max}, \quad \hat{f}(\lambda_l) = 0$$

The set of bandlimited signals $\lambda_{\max}$ with bandwidth is called the **Paley-Wiener space** $PW_{\lambda_{\max}}(\mathcal{G})$.



A **uniqueness set** for $PW_{\lambda_{\max}}(\mathcal{G})$ is a subset of vertices $\mathcal{S} \subset \mathcal{V}$ for which

$$\forall f, g \in PW_{\lambda_{\max}}(\mathcal{G}), \ f(\mathcal{S}) = g(\mathcal{S}) \Rightarrow f = g$$

The smallest **uniqueness set** for $PW_{\lambda_l}(\mathcal{G})$ has a size of l

Recovering the missing samples

Let $f \in PW_{\lambda_n}$, and $\mathcal{S}$ be a minumum uniqueness set.
Estimate the graph Fourier coefficients

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}(\mathcal{S}) \\ \mathbf{f}(\mathcal{S}^c) \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1(\mathcal{S}) & \dots & \mathbf{u}_N(\mathcal{S}) \\ \mathbf{u}_1(\mathcal{S}^c) & \dots & \mathbf{u}_N(\mathcal{S}^c) \end{bmatrix} \begin{bmatrix} \hat{\mathbf{f}}(1) \\ \vdots \\ \mathbf{f}(\hat{n}) \\ \mathbf{0} \end{bmatrix}$$

Thus

$$\begin{bmatrix} \hat{\mathbf{f}}(1) \\ \vdots \\ \hat{\mathbf{f}}(n) \end{bmatrix} = (\tilde{\mathbf{U}}_n(\mathcal{S}))^{-1}\mathbf{f}(\mathcal{S})$$

with $\tilde{\mathbf{U}}_n$ being the n first eigenvectors.
Finally,

$$\mathbf{f}(\mathcal{S}^c) = \tilde{\mathbf{U}}_n(\mathcal{S}^c) \begin{bmatrix} \hat{\mathbf{f}}(1) \\ \vdots \\ \hat{\mathbf{f}}(n) \end{bmatrix}$$

[Narang, S. K., Gadde, A., Sanou, E., and Ortega, A. (2013, December). Localized iterative methods for interpolation in graph structured data. In 2013 IEEE Global Conference on Signal and Information Processing (pp. 491-494). IEEE.]

Graph sampling

In the equation $\begin{bmatrix} \hat{\mathbf{f}}(1) \\ \vdots \\ \hat{\mathbf{f}}(n) \end{bmatrix} = (\tilde{\mathbf{U}}_n(\mathcal{S}))^{-1}\mathbf{f}(\mathcal{S})$, the matrix $\tilde{\mathbf{U}}_n(\mathcal{S})$ should be invertible.

Sampling algorithms consist in finding the set $\mathcal{S}$ for which, $\tilde{\mathbf{U}}_n(\mathcal{S})$ is invertible

Initialize: $\mathcal{S} \leftarrow \mathcal{V}_i$ where i is the index of any nonzero element offirst eigenvector

for $m = 2 \rightarrow n$ **do**

  Compute $\mathbf{x} = null(\tilde{\mathbf{U}}_m(\mathcal{S}))$

  Compute $\mathbf{b} = \tilde{\mathbf{U}}_m(\mathcal{S}^c)\mathbf{x}$

  $i \leftarrow argmax_i(|\mathbf{b}(i)|)$

  $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_c(i)$

end

[D. E. Tzamarias, P. Akyazi, and P. Frossard, \A novel method for sampling bandlimited graph signals," in Proceedings of EUSIPCO, no. CONF, 2018.]

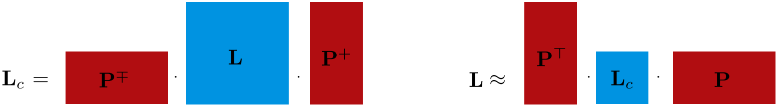
Graph Coarsening

Coarsening: From an initial graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{L}\}$ with N nodes and a signal $\mathbf{x}$, build a new coarsened graph $\mathcal{G}_c$ with N_c nodes:

$$\mathbf{x}_c = \mathbf{P}\mathbf{x}$$

$$\tilde{\mathbf{x}} = \mathbf{P}^{+}\mathbf{x}_c$$

where $\mathbf{P} \in \mathbb{R}^{N_c \times N}$ are matrices with more columns than rows and $\mathbf{P}^{+}$ the pseudo-inverse.



[A. Loukas and P. Vandergheynst, ''Spectrally approximating large graphs with smaller graphs,'' arXiv preprint arXiv:1802.07510, 2018]

Graph Coarsening

With the constraint that $\mathbf{L}_c$ is a graph Laplacian, we have:

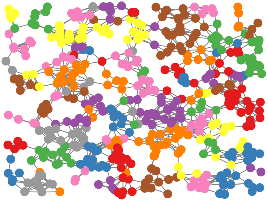
$$\mathbf{P}(r,i) = \begin{cases} \frac{1}{\|\mathcal{V}^{(r)}\|} & \text{if } v_i \in \mathcal{V}^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

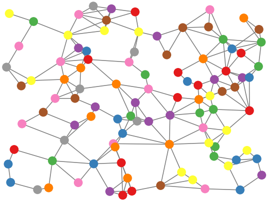
$$\mathbf{P}^{+}(i,r) = \begin{cases} 1 & \text{if } v_i \in \mathcal{V}^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Graph Coarsening

Results of the recursive coarsening available in

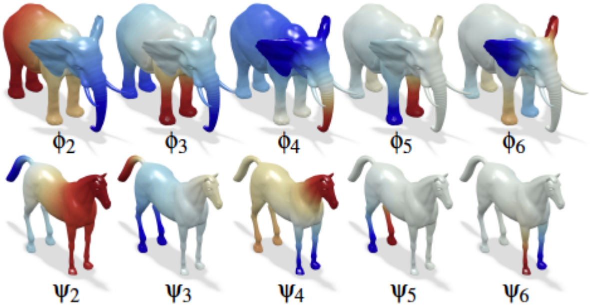
`https://github.com/loukasa/graph-coarsening`





Motivations

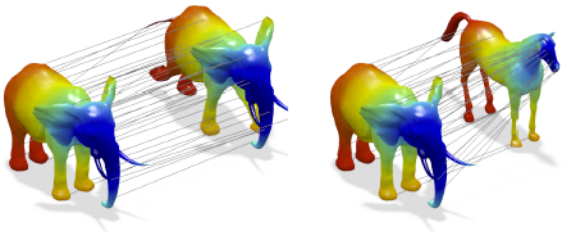
For different graph structure $\mathbf{L}_1$ and $\mathbf{L}_2$, the graph transform $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ may drastically vary



[Kovnatsky, A., Bronstein, M. M., Bronstein, A. M., Glashoff, K., and Kimmel, R. (2013, May). Coupled quasi-harmonic bases. In Computer Graphics Forum (Vol. 32, No. 2pt4, pp. 439-448). Oxford, UK: Blackwell Publishing Ltd.]

A priori step

Set correspondences



And register them in matrices $\mathbf{F}$ and $\mathbf{G}$

Joint Laplacian diagonalization

Solve the problem

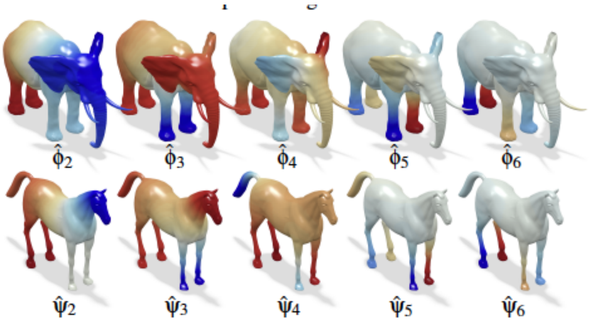
$$\min_{\hat{\mathbf{\Phi}},\hat{\mathbf{\Psi}}} = off(\hat{\mathbf{\Phi}}^\top \mathbf{L}_1 \hat{\mathbf{\Phi}}) + off(\hat{\mathbf{\Psi}}^\top \mathbf{L}_2 \hat{\mathbf{\Psi}}) + \mu ||\mathbf{F}^\top \hat{\mathbf{\Phi}} - \mathbf{G}^\top \hat{\mathbf{\Psi}}||$$

$$\text{s.t.} \quad \hat{\mathbf{\Phi}}^\top \hat{\mathbf{\Phi}} = \mathbf{I} \text{ and } \hat{\mathbf{\Psi}}^\top \hat{\mathbf{\Psi}} = \mathbf{I}$$

Solved with

- Rotation and permutation of the original eigenvectors
- Gradient descent

Results


$\hat{\phi}_2$
$\hat{\phi}_3$
$\hat{\phi}_4$
$\hat{\phi}_5$
$\hat{\phi}_6$
$\hat{\psi}_2$
$\hat{\psi}_3$
$\hat{\psi}_4$
$\hat{\psi}_5$
$\hat{\psi}_6$

Results

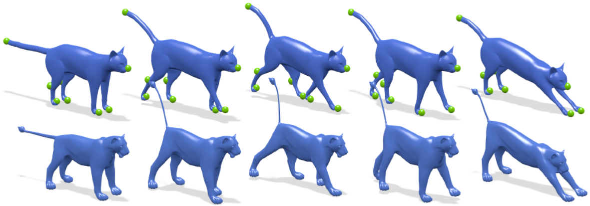
Moving object

Results

Two similar shapes

Results

Application for motion generation



[Rong, G., Cao, Y., and Guo, X. (2008). Spectral mesh deformation. The Visual Computer, 24, 787-796.]

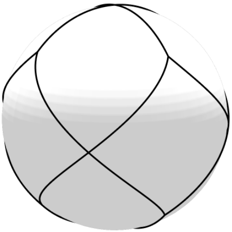
Results

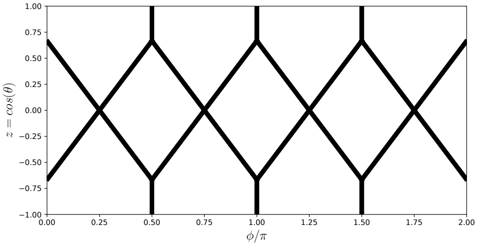
Texture transfer

[Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., and Bronstein, M. M. (2017). Geometric deep learning on graphs and manifolds using mixture model CNNs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5115-5124).]

HEALPix sampling

- The pixelization starts with partitioning the sphere into 12 equal-area regions (base resolution).

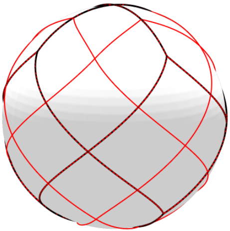


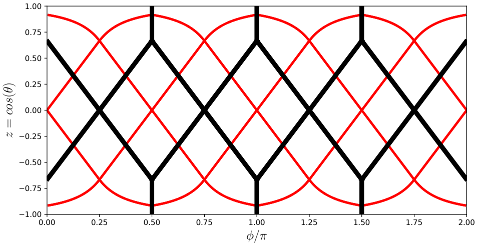


cylindrical projection

HEALPix sampling

- The pixelization starts with partitioning the sphere into 12 equal-area regions (base resolution).
- Finer pixelization is achieved by dividing each region into 4 equal-area regions.

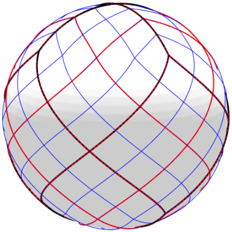


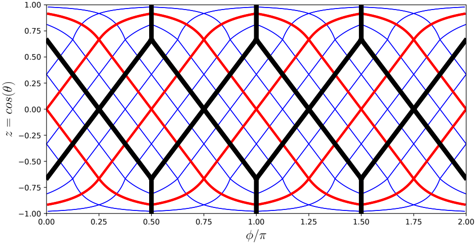


cylindrical projection

HEALPix sampling

- The pixelization starts with partitioning the sphere into 12 equal-area regions (base resolution).
- Finer pixelization is achieved by dividing each region into 4 equal-area regions.
- Hierarchical partitioning is repeated to reach the desired resolution.





cylindrical projection

All pixel centers are placed on rings of constant latitude.

HEALPix sampling

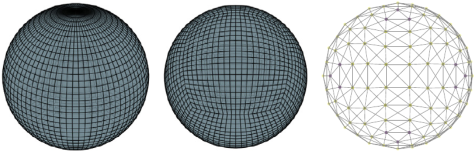


Fig. 6. Some pixelizations of the sphere. Left: the equirectangular grid, using equiangular spacing in a standard spherical-polar coordinate system. Middle: an equiangular cubed-sphere grid, as described in Ronchi et al. (1996). Right: graph built from a HEALPix pixelization of half the sphere ($N_{side} = 4$). By construction, each vertex has eight neighbors, except the highlighted ones which have only seven.[4]
Source: Left and middle figures are taken from Boomsma and Frellsen (2017)

[N. Perraudin, M. Defferrard, T. Kacprzak, R. Sgier, DeepSphere: Efficient spherical convolutional neural network with HEALPix sampling for cosmological applications, Astronomy and Computing, Volume 27, 2019, Pages 130-146, ISSN 2213-1337]

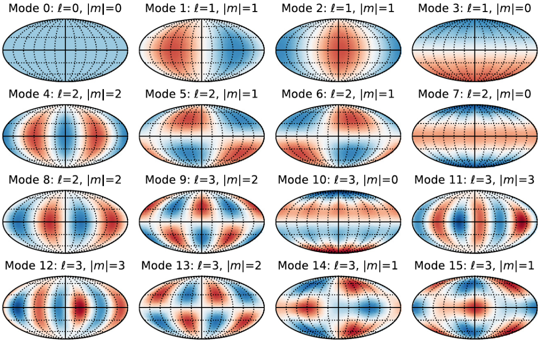
Eigenvectors on HEALPix sampling

- Remember that

$$\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$$

- The transformed of a signal $\mathbf{f}$ is given by

$$\hat{\mathbf{f}} = \mathbf{U}^\top\mathbf{f}$$



[N. Perraudin, M. Defferrard, T. Kacprzak, R. Sgier, DeepSphere: Efficient spherical convolutional neural network with HEALPix sampling for cosmological applications, Astronomy and Computing, Volume 27, 2019, Pages 130-146, ISSN 2213-1337]

Spatial convolution on the graph

Let $\boldsymbol{\alpha}$ be the convolution kernel. The *DeepSphere* convolution is:

$$\mathbf{x} \star \boldsymbol{\alpha} := \sum_{l=0}^{L} \alpha_l \mathbf{L}^l \mathbf{x},$$

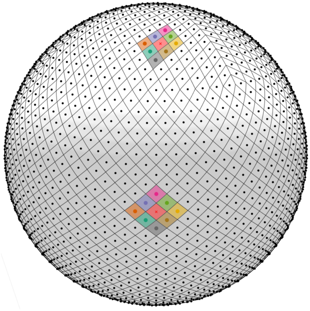
where L is the polynom degree. It controls the kernel widows size.
$\rightarrow$ Only one weight per neighborhood (isotropic filter).

Example of 1-hop:

$$(\mathbf{x} \star \boldsymbol{\alpha})(i) := \alpha_0 x_i + \alpha_1 . \left(\sum_{k=1}^{8} l_{i,\mathcal{N}_i(k)} x_{\mathcal{N}_i(k)} \right)$$

where $\mathcal{N}_i(k)$ is the k^{th} neighbor of pixel i.

Alternatives on the sphere

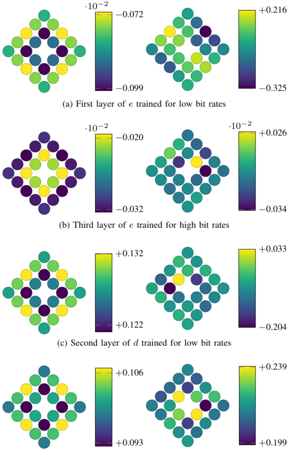


One weight per neighbor: expressive and anistropic filter

$$(\mathbf{x} \star \boldsymbol{\theta})(i) := \theta_0 \cdot x_i + \sum_{k=1}^{8} \theta_k \cdot x_{\mathcal{N}_i(k)} \cdot w_{\mathcal{N}_i(k),i}$$

[Mahmoudian Bidgoli, N., Azevedo, R. G. D. A., Maugey, T., Roumy, A., and Frossard, P. (2021). OSLO: On-the-Sphere Learning for Omnidirectional images and its application to 360-degree image compression. arXiv e-prints, arXiv-2107.]

Alternatives on the sphere



(a) First layer of e trained for low bit rates

(b) Third layer of e trained for high bit rates

(c) Second layer of d trained for low bit rates

(d) Forth layer of d trained for high bit rates

Fig. 12. Random selection of learned filters at different layers of e and s. The weight corresponding to the central pixels is not shown in the filter for better visibility. Left column represents filters learned with DeepSphere that results in isotropic filters. Right column represents the filters learned with OSLO solution.

[Mahmoudian Bidgoli, N., Azevedo, R. G. D. A., Maugey, T., Roumy, A., and Frossard, P. (2021). OSLO: On-the-Sphere Learning for Omnidirectional images and its application to 360-degree image compression. arXiv e-prints, arXiv-2107.]

Alternatives on the sphere

(a) 2408.43 KB (b) 936.65 KB

Fig. 13. Decompressed images in Mollweide projection. The zoomed versions of red, green, and blue rectangular regions are shown in the bottom row. (a) DeepSphere result stored in 2408.43 KB with 29.54 dB and 29.51 dB for SPSNR and WSPSNR respectively. (b) OSLO result stored in 936.65 KB with 40.91 dB and 41.65 dB for SPSNR and WSPSNR respectively.

[Mahmoudian Bidgoli, N., Azevedo, R. G. D. A., Maugey, T., Roumy, A., and Frossard, P. (2021). OSLO: On-the-Sphere Learning for Omnidirectional images and its application to 360-degree image compression. arXiv e-prints, arXiv-2107.]

Table of Contents

Paper study

Paper-1: Optimal transport methodology for climate modeling
Barré *et al.* 2020: *Averaging atmospheric gas concentration data using wasserstein barycenters*

Paper-2: Deep learning for waste sorting
Chen *et al.* 2020: *iWaste: video-based medical waste detection and classification.*

Paper-3: Deep super resolution for climate modeling
Vandal *et al.* 2017. *Deepsd: Generating high resolution climate change projections through single image super-resolution.*

Answer the following questions

- What is the societal challenge that is tackled ?
- What is the scientific problem that is tackled ?
- What is the contribution ?
- What is the obtained result ?

Image Processing & Ecology

Societal challenges:

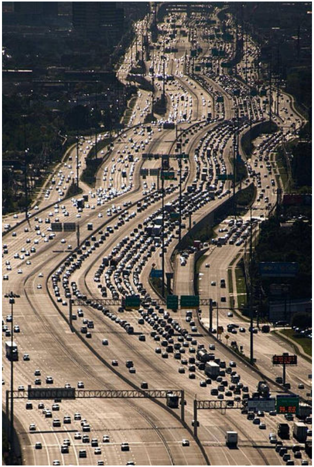
- Climate change measurement and modeling
- Energy Consumption reduction (automation, optimization)
- ...

Scientific challenges:

- Image super-resolution
- Image/Video compression
- Image understanding
- Data modeling
- ...

Some awareness: the rebound effect

Some awareness: the rebound effect



Katy Freeway, Houston (Wikipedia)

Jevon's paradox (or rebound effect):

As technological improvements increase the efficiency with which a resource is employed, the total consumption of that resource may increase rather than decrease. (Wikipedia)

- Direct *vs* indirect
- comes when the usage of the technology is not questioned
- limited by sensitization and by thinking the problem globally

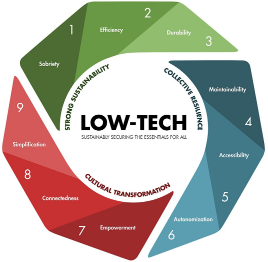
Some awareness: maintain *conviviality*

In ***Tools for Conviviality*** (Ivan Illich 1973):

Recent development brings:

- modernized poverty
- dependency
- out-of-control system

Instead we should *"invert the present deep structure of tools"* in order to *"give people tools that guarantee their right to work with independent efficiency"* → technology for emancipation



THE CRITERIA FOR ANY LOW-TECH INNOVATION APPROACH:

STRONG SUSTAINABILITY

1 **Sobriety**
Refocuses on the essentials and tends toward the technological optimum: lowest technological intensity and greatest simplicity ensuring needs be met with a high level of reliability

2 **Efficiency**
Minimizes the consumption of energy and resources, from extraction of raw materials through production, distribution and use to end of life

3 **Durability**
Presents maximum technical, functional, ecological as well as human viability in the short, medium and long term

COLLECTIVE RESILIENCE

4 **Maintainability**
Can be maintained and repaired by users themselves so far as possible, using standard parts and materials

5 **Accessibility**
Offers maximum ease of use

6 **Autonomization**
Is made from resources that are exploited and transformed as locally as possible

CULTURAL TRANSFORMATION

7 **Empowerment**
Facilitates appropriation by the greatest number, gives power to citizens and communities

8 **Connectedness**
Promotes the sharing of knowledge and know-how, cooperation, solidarity, social cohesion and links between communities

9 **Simplification**
Decomplexifies society at the socio-economic and organizational levels based on reflection about needs and vulnerabilities

Design: Arthur Keller and Emilien Bournigal