Graph-Projected Signal Processing

Nicolas Grelier, Carlos Eduardo Rosar Kos Lassance, Elsa Dupraz and Vincent Gripon
IMT Atlantique, CNRS Lab-STICC
Brest, France

***Abstract*—In the past few years, Graph Signal Processing (GSP) has attracted a lot of interest for its aim at extending Fourier analysis to arbitrary discrete topologies described by graphs. Since it is essentially built upon analogies between classical temporal Fourier transforms and ring graphs spectrum, these extensions do not necessarily yield expected convolution and translation operators when adapted on regular multidimensional domains such as 2D grid graphs. In this paper we are interested in alternate definitions of Fourier transforms on graphs, obtained by projecting vertices to regular metric spaces on which the Fourier transform is already well defined. We compare our method with classical graph Fourier transform and demonstrate its interest for designing accurate convolutional neural networks on graph signals.**

***Index Terms*—Graph Signal Processing, Graph Embedding**

I. Introduction

Graph Signal Processing (GSP) is a mathematical framework that allows to define generalized discrete Fourier transform adapted to any topology described by a graph [8]. Formally, let us consider a weighted graph $G = \langle V, W\rangle$, where $V = \{v_1, \dots, v_n\}$ is the set of indexed vertices and $W \in \mathbb{R}^{n^2}$ is the symmetric adjacency matrix. We call Laplacian L associated with G the matrix $L = D - W$, where D is the diagonal strength matrix associated with G. Being symmetric and real valued, L can be written as $L = F\Lambda F^\top$ where F is orthonormal and Λ is diagonal. We call signal a vector $\mathbf{x} \in \mathbf{R}^n$ and Graph Fourier Transform (GFT) of $\mathbf{x}$ the vector $\hat{\mathbf{x}} \triangleq F^\top \mathbf{x}$. When the adjacency matrix W is circulant, columns of F can be chosen as usual Fourier modes. More generally, the rationale behind GSP is that columns of F are ad-hoc Fourier modes for the considered graph topology.

Based on these definitions, it is possible to propose convolutions and translations [10] on graphs. These definitions have for example been used in order to design Graph Convolutional Neural Networks [1]. When considering the toy example of a ring graph, these definitions usually match exactly the usual corresponding 1D regular operators. However, when considering regular graphs of higher intrinsic dimension, they diverge from their regular counterparts. In this work we are interested in defining GFTs that match exactly their regular counterparts when defined on regular grid graphs, as we believe such GFTs could improve the performance of machine learning routines (including Convolutional Neural Networks) defined on graph signals.

To this end, we propose the following methodology, taking as inputs a graph G and an integer d:

1) Project vertices of the graph to $\mathbb{Z}^d$, such that geodesic distances between vertices in the graph are close to Manhattan distance between their projections in $\mathbb{Z}^d$,
2) Define GFT on G as a particularization of the usual multidimensional GFT on $\mathbb{Z}^d$.

Obviously we expect this method to perform particularly well when facing approximations of (regular) grid graphs.

In this paper, we introduce the problem statement for step 1 and initial results in Section II. We prove this method guarantees that a 2D grid graph projection is exactly a rectangle in $\mathbb{Z}^2$, and that it is robust to minor changes in the graph structure in Section III. In Section IV we introduce an optimization method for step 1. In Section V we perform experiments and compare with other existing methods. Section VI is a conclusion.

II. Problem statement

Let us consider a weighted graph $G = \langle V, W\rangle$.

Definition 1. *We call **embedding** a function $\phi : V \to \mathbb{Z}^d$, where $d \in \mathbb{N}^*$.*

We are specifically interested in embeddings that preserve distances. Specifically, we define the cost $c_\alpha(\phi)$ of an embedding ϕ as the following quantity:

$$c_\alpha(\phi) \triangleq \sum_{v,v' \in V} \left|\alpha\|\phi(v) - \phi(v')\|_1 - d_G(v, v')\right| , \quad (1)$$

where d_G is the shortest path distance in G. In the remaining of this work, we denote $\delta(v, v') = |\alpha\|\phi(v) - \phi(v')\|_1 - d_G(v, v')|$.

Definition 2. *Given a fixed value of α, we call* optimal embedding *an embedding with minimum cost.*

Let us motivate the choices in this definition:

- First, we consider all pairs of vertices and not only edges. Consider for example a ring graph where each vertex has exactly two neighbors. Then there are plenty of embeddings that would minimize the cost if considering only edges, but only a few that minimize the sum over all pairs of vertices.
- Second, we use a sum and not a maximum. This is because small perturbations of grid graphs would lead to dramatic changes in embeddings minimizing the cost if using a maximum. Consider for instance a 2D grid graph in which an arbitrary edge is removed.
- Thirdly, we choose embedding in $\mathbb{Z}^d$ instead of in $\mathbb{R}^d$, as we want to particularize multidimensional *discrete* Fourier transforms.

- Fourthly, we use the Manhattan distance, as it is more naturally associated with $\mathbb{Z}^d$ than the Euclidean distance. It also ensures there exists natural embeddings for grid graphs with cost 0.
- Finally, α is a scaling factor.

Note that the question of finding suitable embeddings for graphs is not novel [4], [5]. But to our knowledge enforcing the embedding to be in $\mathbb{Z}^d$ is a main discriminative point compared to previous work. Even though Definitions 1 and 2 work for any d we are going to show the consistency of these definitions by considering the particular case $d = 2$.

Definition 3. *We call (2D)* grid graph *a graph whose vertices are of the form* $\{1, \dots, \ell\} \times \{1, \dots, h\}$ *where* $\ell, h \geq 3$*, and edges are added between vertices at Manhattan distance one from each other.*

Definition 4. *We call* natural embedding *of a grid graph the identity function.*

A first interesting result is related to optimal embeddings of grid graphs.

Theorem 1. *The natural embedding of a grid graph is its only optimal embedding for* $\alpha = 1$*, up to rotation, translation and symmetry.*

Proof: The proof is straightforward, as the cost of the natural embedding is clearly 0. Reciprocally, a cost of 0 forces any group of vertices $\{(x, y), (x, y'), (x', y), (x', y')\}$ to be projected to a translation, rotation and/or symmetry of the corresponding rectangle in $\mathbb{Z}^2$. Then any remaining vertex is uniquely defined from these four ones. ■

III. Robustness to small perturbation of grid graphs

In this section we are interested in showing that the optimal embedding is robust to small perturbations of grid graphs. Here, we always consider $\alpha = 1$. Let us first introduce quasi grid graphs as grid graphs with one missing edge:

Definition 5. *A* quasi grid graph *is a grid graph with one missing edge between vertices* (i, j) *and* $(i + 1, j)$*, with* $1 \leq i < \ell$ *and* $1 < j < \ell$*.*

We first prove results on two-slices graphs which is illustrated on Figure 1:

Definition 6. *We call* two-slices graph *a weighted graph with vertices of* $\{1, 2, 3\} \times \{1, h\}$*, where* $h \geq 2$*. It contains four weight-1 edges between vertices at Manhattan distance 1 from each other, and two weight-*h *edges between vertices on the corner. We call natural embedding of a two-slices graph the identity function on its vertices.*

First observe that the natural embedding of a two-slices graph has a cost of 4.

Lemma 1. *Consider an embedding* ϕ *of a two-slices graph for which* $\|\phi((2, 1)) - \phi((2, h))\|_1 \geq h + 4$*, then* $c_1(\phi) \geq 6$*.*

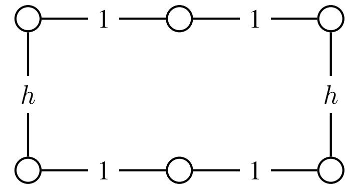

Figure 1. Depiction of a two-slices graph.

Proof. It holds that $c_1(\phi) \geq 2\delta((2, 1), (2, h)) + 2\delta((1, 1), (2, h)) + 2\delta((1, 1), (2, 1))$. In particular the first term of the sum is at least 4. Then we discuss two cases: a) $\delta((1, 1), (2, 1)) \geq 1$ in which case the result is immediate or b) $\delta((1, 1), (2, 1)) \leq 1$ in which case the Manhattan distance between $(1, 1)$ and $(1, h)$ is at least 3. □

Lemma 2. *Consider an embedding* ϕ *of a two-slices graph for which* $\|\phi((2, 1)) - \phi((2, h))\|_1 = h + 3$*, then* $c_1(\phi) \geq 8$*.*

Proof. Again, we look at the vertex $(1, 1)$. Either $\delta((1, 1), (2, 1)) \geq 1$ or $(1, 1)$ is at distance at least $h + 2$ from $(2, h)$. In both cases, we obtain $\delta((1, 1), (2, 1)) + \delta((1, 1), (2, h)) \geq 1$. We repeat the process for the other three vertices $(1, h), (3, h)$ and $(3, 1)$. □

Lemma 3. *Consider an embedding* ϕ *of a two-slices graph for which* $\|\phi((2, 1)) - \phi((2, h))\|_1 = h + 2$*, then* $c_1(\phi) \geq 8$*.*

Proof. Denote v one of the four border vertices ($(1, 1), (1, h), (3, 1)$ or $(3, h)$). Either a) $\delta(v, (2, 1)) = \delta(v, (2, h)) = 0$, or b) $\delta(v, (2, 1)) + \delta(v, (2, h)) \geq 2$. Indeed, either v is on a shortest (Manhattan) path between $(2, 1)$ and $(2, h)$, or it is not. If two of the border vertices verify b) the proof is finished. If only one verifies b), then one of the following holds: i) $\|\phi((1, 1)) - \phi((3, h))\|_1 = h$ or ii) $\|\phi((3, 1)) - \phi((1, h))\|_1 = h$. In both cases we conclude. Finally, if all border vertices are in case a), then both i) and ii) hold. □

Lemma 4. *Consider an embedding* ϕ *of a two-slices graph for which* $\|\phi((2, 1)) - \phi((2, h))\|_1 = h + 1$*, then* $c_1(\phi) \geq 8$*.*

Proof. The proof is omitted as it is similar to that of Lemma 2. □

Lemma 5. *The only embedding* ϕ *of a two-slices graph for which* $\|\phi((2, 1)) - \phi((2, h))\|_1 = h$ *and* $c_1(\phi) = 4$ *is the natural embedding, up to rotation, translation and symmetry.*

Proof. If $\phi((2, 1))$ and $\phi((2, h))$ are on the same line or column, the reasoning is very similar to that of Theorem 1.

We already have $\delta((2, 1), (2, h)) = 2$. So to achieve $c_1(\phi) = 4$, we must have $(1, 1)$ and $(1, h)$ not embedded on a shortest (Manhattan) path between the embedded versions of $(2, 1)$ and $(2, h)$. Their distance is thus $h + 2$. □

Lemma 6. *The only optimal embedding of a two-slices graph is the natural embedding, up to rotation, translation and symmetry.*

Proof. First observe that any embedding for which $\delta((2,1),(2,h)) \geq 3$ is not optimal. We conclude with Lemmas 1 to 5. □

Now we can derive the main result of the Section:

Theorem 2. *The only optimal embedding of a quasi grid graph is the natural embedding, up to rotation, translation and symmetry.*

Proof. Consider the missing edge in the quasi grid graph is at column j. Now we can discuss the sum in c_1. For couple of vertices one of which is not on column $j-1$, j or $j+1$, observe that the natural embedding has cost 0. For the remaining couple of vertices, we discuss on the rows: if both rows are above or both rows are below the missing edge, cost is 0. For the remaining cases, we use Lemma 6 to prove the natural embedding is the only one to minimize the cost. □

IV. Proposed optimization method

Using a bruteforce approach to find optimal embeddings is clearly impossible for graphs on reasonable order. In order to find a proxy to an optimal embedding, we propose to use the combination of gradient descent and barrier [7] methods. We start with a random projection in a hypercube of $\mathbb{R}^d$. The gradient descent aims at minimizing the weighted sum of the cost of the embedding and a penalty of how far away it is from $\mathbb{Z}^d$. The barrier method consists in smoothly modifying the scaling of the two parts of the cost by making use of two scaling parameters β and γ. As a result, at the first iterations the optimization is performed in $\mathbb{R}^d$ and we only aim at minimizing the cost, whereas in the last iterations we enforce the solution to be in $\mathbb{Z}^d$. In the experiments, we use $\beta = 10^{-6}$ and $\gamma = 1.03$.

An overview of the algorithm is:

Data: Graph G, dimension d, scaling factor α.
$\phi_0 \leftarrow$ random embedding
$\beta_0 \leftarrow \beta$
for i *in* *(1,...,K)* **do**
  $\beta_i \leftarrow \gamma\beta_{i-1}$
  $\phi_i \leftarrow GradMin_\phi\left(\frac{c_\alpha(\phi)+\beta_i\left[\sum_v d_1(\phi(v),\mathbb{Z}^d)\right]}{1+\beta_i}, \phi_{i-1}\right)$
end

Algorithm 1: Proposed method, $d_1(\phi(v),\mathbb{Z}^d)$ denotes the Manhattan distance between $\phi(v)$ and $\mathbb{Z}^d$, $GradMin_\phi(c(\phi),\phi_0)$ denotes a gradient descent algorithm with parameter ϕ starting from $\phi = \phi_0$ and aiming at minimizing $c(\phi)$.

A depiction of different steps in the algorithm is presented in Figure 2. On the first line we consider a 2D grid graph, whereas on the second line we represent a random geometric graph. As we can see, the embedding at the end of the first gradient descent is already the natural embedding for the grid graph. This is because of the choice of the Manhattan distance in the definition of c_α. To the contrary, the embedding obtained at the end of the first gradient descent is not aligned with $\mathbb{Z}^2$ in the case of the random geometric graph.

V. Experiments and results

In terms of experiments, we first compare costs of different methods for finding an embedding of grid graphs. We compare our proposed method with: a) a random embedding and b) an embedding obtained by considering the two eigenvectors associated with the smallest nonzero eigenvalues of the Laplacian of the graph. Note that for each method we choose the value of α that minimizes the cost. In the next experiment, we look at the effect of removing some of the edges in a grid graph with parameters $l, h = 10$ in terms of the distance between the obtained embedding and the natural one. Both results are summarized in Figure 4. Note that for random methods, we averaged the results on 100 tests. We observed very little deviation due to the initialization step, in the proposed method.

In Figure 3, we compare the eigenvectors/Fourier modes associated with the first nonzero eigenvalues/frequencies on a random geometric graph for a spectral definition based on the Laplacian and our proposed graph-projected approach. We can clearly see the interest of our proposed method in such a case where the underlying 2D structure is very important during the creation of the graph.

Finally, we compare our method against CNNs and graph CNNs in regular 2D grids and irregular 3D grid scenarios. In Table I we compare the performance of convolutional neural networks using our methodology with spectral solutions on the CIFAR-10 dataset an image dataset (2D grid). Our method is able to use the same architecture as a classic CNN, while [3] and [9] use a comparable but not equal architecture. Considering data augmentation, random crops are used for all methods but [9], which uses a method that approximates random crops using translations learned from the graph.

Table I
CIFAR-10 test set accuracy comparison table.

			Spectral	Nonspectral	
MLP [6]	CNN	Support	[3]	[9]	Proposed
78.62%	92.73%	Grid	84.41%	92.81%	92.73%
		Covariance graph	——	91.07%	89.25%

For the irregular 3D grid scenario we use the PINES dataset, which consists of fMRI scans on 182 subjects, during an emotional picture rating task [2]. We use the same methods from [9] to generate an irregular 3D grid. The final volumes used for classification contain 369 signals for each subject and rating. We used a shallow network for the classification task. The results are presented on Table II. We show that the proposed method is competitive with existing methods.

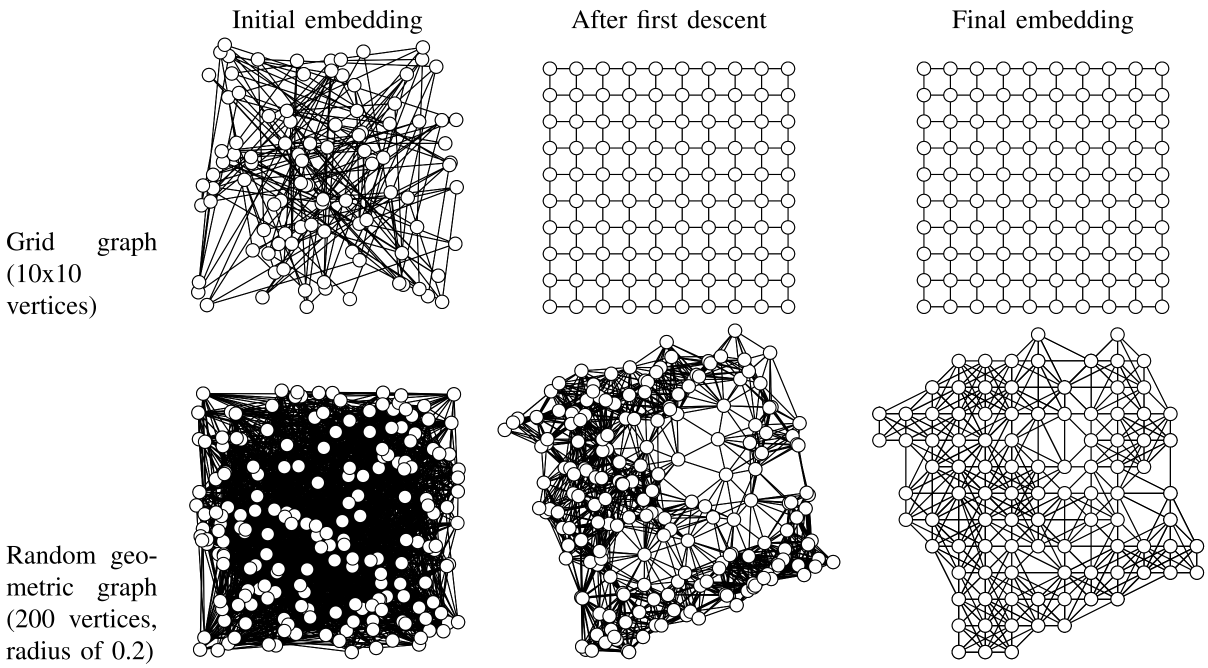

Figure 2. Illustration of successive steps in the proposed optimisation procedure (using Algorithm 1). Left column is initial random embedding, middle column is after the first gradient descent, and right column at the end of the process. A parameter of $\alpha = 1$ was used for the grid graph and $\alpha = 2$ for the random geometric graph.

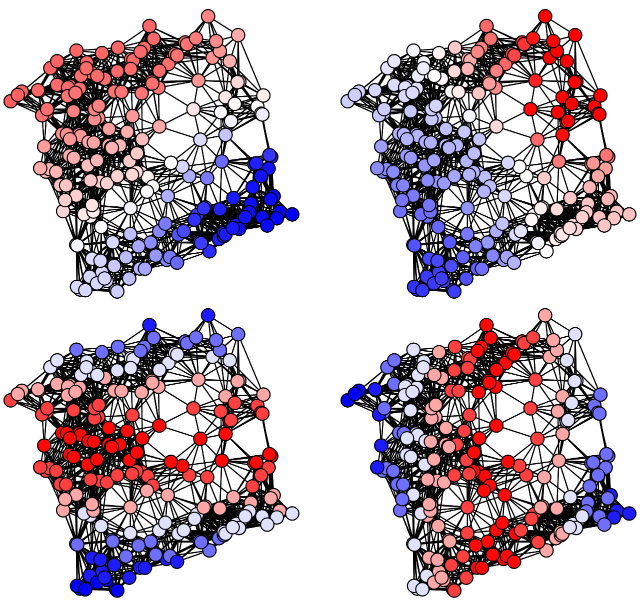

Figure 3. Depiction of the first eigenvectors/Fourier modes (associated with the lowest nonzero eigenvalues/frequencies) of a random geometric graph. First line corresponds to the classical graph signal processing definition (using the Laplacian), and second line to the proposed embedding.

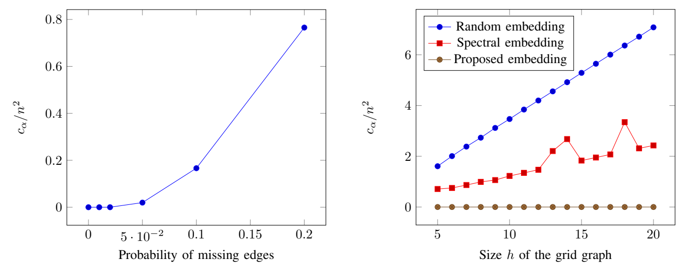

Figure 4. Comparison of embedding costs obtained by various methods (left) and evolution of the distance between a noisy grid graph embedding found using Algorithm 1 and its natural embedding (right).

Table II
PINES fMRI DATASET ACCURACY COMPARISON TABLE. RESULTS ARE PRESENTED USING THE MEAN OVER 10 TESTS. CNNS USE A 9X9X9 FILTER SIZE.

Graph	None		Neighborhood Graph		
Method	Dense	CNN	[3]	[9]	Proposed
Accuracy	82.62%	85.47%	82.80%	85.08%	84.78%

VI. Conclusion

We introduced an alternative definition of the Fourier transform on a graph. In the case of graphs obtained by a sampling of a regular metric space, we believe the proposed method can lead to better definition of Fourier modes and associated operators. This is supported by the experiments using convolutional neural networks.

In our experiments, finding Fourier modes for graphs containing up to a few thousands of vertices can be performed in a few minutes on a modern computer, thus requiring a time similar to that of finding eigenvectors and eigenvalues of the corresponding Laplacian matrix. It would be interesting to stress the interest of the proposed method for other graph signal uses. Also we consider running experiments on highly irregular graphs.

Acknowledgements

This work was funded in part with the support of the Brittany region.

References

[1] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[2] Luke J Chang, Peter J Gianaros, Stephen B Manuck, Anjali Krishnan, and Tor D Wager. A sensitive and specific neural signature for picture-induced negative affect. *PLoS biology*, 13(6):e1002180, 2015.

[3] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3837–3845, 2016.

[4] Stephen G. Kobourov. Force-directed drawing algorithms, 2004.

[5] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[6] Zhouhan Lin, Roland Memisevic, and Kishore Reddy Konda. How far can we go without convolution: Improving fully-connected networks. *CoRR*, abs/1511.02580, 2015.

[7] Walter Murray and Kien-Ming Ng. An algorithm for nonlinear optimization problems with binary variables. *Computational Optimization and Applications*, 47(2):257–288, 2010.

[8] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing. *arXiv preprint arXiv:1712.00468*, 2017.

[9] Carlos Eduardo Rosar Kos Lassance, Jean-Charles. Vialatte, and Vincent Gripon. Matching Convolutional Neural Networks without Priors about Data. *ArXiv e-prints*, abs/1802.09802, February 2018.

[10] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.