Title

How Can We Naturally Order and Organize Graph Laplacian Eigenvectors?

Author

Saito, Naoki

Publication Date

2018-06-13

DOI

10.1109/ssp.2018.8450808

HOW CAN WE NATURALLY ORDER AND ORGANIZE GRAPH LAPLACIAN EIGENVECTORS?

Naoki Saito

University of California, Davis
Department of Mathematics
Davis, CA 95616 USA

ABSTRACT

When attempting to develop wavelet transforms for graphs and networks, some researchers have used graph Laplacian eigenvalues and eigenvectors in place of the frequencies and complex exponentials in the Fourier theory for regular lattices in the Euclidean domains. This viewpoint, however, has a fundamental flaw: on a general graph, the Laplacian eigenvalues *cannot* be interpreted as the *frequencies* of the corresponding eigenvectors. In this paper, we discuss this important problem further and propose a new method to organize those eigenvectors by defining and measuring "natural" distances between eigenvectors using the Ramified Optimal Transport Theory followed by embedding them into a low-dimensional Euclidean domain. We demonstrate its effectiveness using a synthetic graph as well as a dendritic tree of a retinal ganglion cell of a mouse.

Index Terms— Graph Laplacian eigenvectors, ramified optimal transport, multidimensional scaling

1. INTRODUCTION

For the theory and practice of discrete wavelets on regular lattices in $\mathbb{R}^d$, Fourier analysis has played a significant role. Hence, when attempting to develop wavelet theory for graphs and networks, some researchers have used graph Laplacian eigenvalues and eigenvectors in place of the frequencies and complex exponentials, respectively; see, e.g., the Spectral Graph Wavelet Transform of Hammond et al. [1]. While tempting to do so, this viewpoint/strategy has several fundamental problems. One of them is the intricate relationship between the frequencies and the Laplacian eigenvalues. For undirected and unweighted paths (or cycles), the Laplacian eigenvectors are the discrete cosine (or Fourier) basis vectors and the corresponding eigenvalues are monotonically increasing functions of the frequency as discussed in [2, 3, 4, 5, 6] among others. Consequently on those simple graphs, one can precisely develop the classical wavelets using the Littlewood-Paley theory [7, Sec. 2.4]. However, as soon as a graph becomes even slightly more complicated (e.g., a discretized rectangle in 2D), the situation completely changes: we cannot view the eigenvalues as a simple monotonic function of frequency anymore [5, 6]. Hence, a fundamental question is how to *order* and *organize* Laplacian eigenvectors without using the eigenvalues, i.e., how to create a *dual* domain of a given graph. In this paper, we investigate this important problem further and propose a new method to order and organize those eigenvectors using the idea from the Ramified Optimal Transport (ROT) theory [8] to measure "natural" distances between eigenvectors followed by embedding them into a low-dimensional Euclidean domain.

This research was partially supported by the NSF grant DMS-1418779 and the ONR grant N00014-16-1-2255; the author also thanks Prof. Qinglan Xia of UC Davis for the fruitful discussion and the `Plots.jl` community for answering his many graphics questions.

2. PROBLEMS WITH ORDERING EIGENVECTORS ACCORDING TO THE CORRESPONDING EIGENVALUES

Let $G = G(V,E)$ be a graph with its vertex set V and edge set E. It is a well-known fact by now [2, 3, 4, 5, 6] that if the graph G is a path with n nodes, i.e., $G = P_n$, then the eigenvectors of the *graph Laplacian matrix* $L(P_n) := D(P_n) - A(P_n)$, where $D(P_n)$, $A(P_n)$ are its degree and adjacency matrices, respectively, are exactly the *DCT Type II* basis vectors (used for the JPEG standard) while those of the *symmetrically-normalized graph Laplacian matrix* $L_{\mathrm{sym}}(P_n) := D(P_n)^{-\frac{1}{2}} L(P_n) D(P_n)^{-\frac{1}{2}}$ are the *DCT Type I* basis vectors. In fact, the eigenpairs of $L(P_n)$ are:

$$\lambda_{k;n} := 4\sin^2\left(\frac{\pi k}{2n}\right),\ \ \phi_{k;n}[x] := a_{k;n}\cos\left(\frac{\pi k}{n}\left(x+\frac{1}{2}\right)\right), \quad (1)$$

where $k, x = 0 : n-1$, and $a_{k;n}$ is a normalization constant to have $\|\boldsymbol{\phi}_{k;n}\|_2 = 1$. It is clear that the eigenvalue is a monotonically increasing function of the *frequency*, which is the eigenvalue index k divided by 2 in this case.

As soon as a graph becomes even slightly more complicated than unweighted and undirected paths/cycles, however, the situation completely changes: we cannot view the eigenvalues as a simple monotonic function of frequency anymore. For example, consider a thin rectangle in $\mathbb{R}^2$, and suppose that this rectangle is discretized as $P_m \times P_n$ ($m > n > 1$). The Laplacian eigenpairs of this graph can be easily

derived from Eq. (1) as:

$$\lambda_k = \lambda_{(k_x,k_y)} := \lambda_{k_x;m} + \lambda_{k_y;n}$$
$$\phi_k[x,y] = \varphi_{k_x,k_y}[x,y] := \phi_{k_x;m}[x] \cdot \phi_{k_y;n}[y]$$

where $k = 0 : mn-1$; $x, k_x = 0 : m-1$; and $y, k_y = 0 : n-1$. As always, let $\{\lambda_k\}_{k=0:mn-1}$ be ordered in the nondecreasing manner. Fig. 1 shows the corresponding eigenvectors ordered in this manner (with $m = 7$, $n = 3$). Note that the layout of 3×7 grid of subplots is for the page saving purpose: the layout of 1×21 grid of subplots would be more natural if we use only the eigenvalue size for eigenvector ordering. In this case, the smallest eigenvalue is still $\lambda_0 = \lambda_{(0,0)} = 0$, and the corresponding eigenvector is constant. The

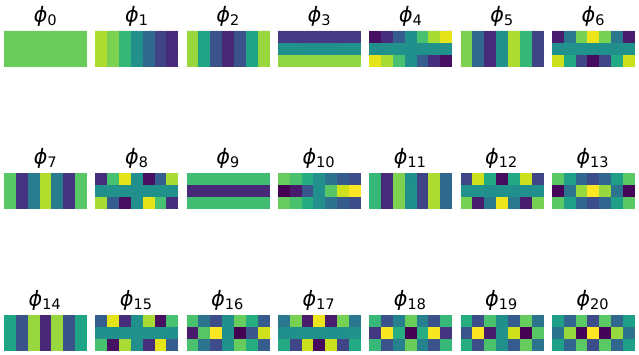


Fig. 1. Laplacian eigenvectors of $P_7 \times P_3$ sequentially ordered in terms of nondecreasing eigenvalues.

second smallest eigenvalue λ_1 is $\lambda_{(1,0)} = 4\sin^2(\pi/2m)$, since $\pi/2m < \pi/2n$, and its eigenvector has half oscillation (i.e., half period) in the x-direction. But, how about λ_2? Even for such a simple situation there are two possibilities for λ_2, depending on m and n. If $m > 2n$, then $\lambda_2 = \lambda_{(2,0)} < \lambda_{(0,1)}$. On the other hand, if $n < m < 2n$, then $\lambda_2 = \lambda_{(0,1)} < \lambda_{(2,0)}$. More generally, if $Kn < m < (K+1)n$ for some $K \in \mathbb{N}$, then $\lambda_k = \lambda_{(k,0)} = 4\sin^2(k\pi/2m)$ for $k = 0, \dots, K$. Yet we have $\lambda_{K+1} = \lambda_{(0,1)} = 4\sin^2(\pi/2n)$ and λ_{K+2} is equal to either $\lambda_{(K+1,0)} = 4\sin^2((K+1)\pi/2m)$ or $\lambda_{(1,1)} = 4[\sin^2(\pi/2m) + \sin^2(\pi/2n)]$ depending on m and n. In Fig. 1, one can see this behavior with $K = 2$. Clearly, the mapping between k and (k_x, k_y) is quite nontrivial. Notice that $\phi_{(k,0)}$ has $k/2$ oscillations in the x-direction whereas $\phi_{(0,1)}$ has only half oscillation in the y-direction. In other words, all of a sudden the eigenvalue of a completely different type of oscillation *sneaks into* the eigenvalue sequence. Hence, on a general graph, by simply looking at its Laplacian eigenvalue sequence $\{\lambda_k\}_{k=0,1,\dots}$, it is *almost impossible to organize the eigenvectors into physically meaningful dyadic blocks and follow the Littlewood-Paley approach* unless the underlying graph is of very simple nature, e.g., P_n or C_n (the cycle consisting of n nodes). Therefore, for complicated graphs, wavelet construction methods that rely on the Littlewood-Paley theory by viewing the Laplacian eigenvalues as the frequencies, such as the spectral graph wavelet transform [1], may face unexpected problems. In fact, not only the wavelet construction methods but also any procedures and applications having that viewpoint would become problematic on general graphs.

What we really want to do is to *organize* those eigenvectors as shown in Fig. 2 instead of Fig. 1. Then a natu-

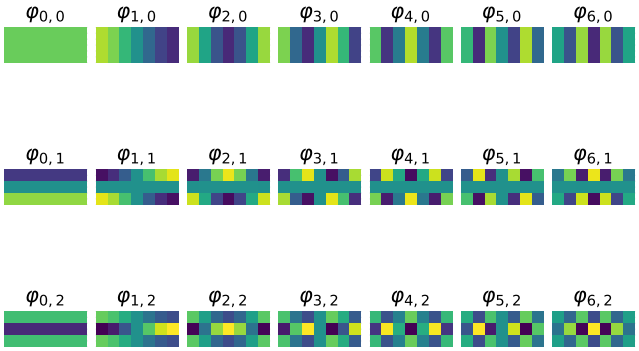


Fig. 2. Laplacian eigenvectors of $P_7 \times P_3$ ordered in terms of their natural horizontal/vertical frequencies.

ral question is: how can we *quantify the difference between the eigenvectors*? Note that the usual ℓ^2-distance does not work since $\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\|_2 = \sqrt{2}\delta_{ij}$ where δ_{ij} is the Kronecker delta. Here, we propose to derive a natural distance between eigenvectors using the ideas gained from the *ramified optimal transport theory* [8], i.e., we view the cost to "transport" one eigenvector to another eigenvector as the natural distance between two such eigenvectors.

3. A BRIEF REVIEW OF RAMIFIED OPTIMAL TRANSPORT THEORY

The Ramified Optimal Transport (ROT) theory [8] is a branch of more general optimal transport theory [9]: it studies transporting "mass" from one probability measure μ^+ to another μ^- along ramified transport paths with some specific transport cost functional, and has been used to analyze various *branching* structures, e.g., trees; veins on a leaf; cardiovascular systems; river channel networks; electrical grids; communication networks, to name a few. For simplicity, we only consider the discrete case: two probability mass functions (pmfs) in $\mathbb{R}^d$. Let $\boldsymbol{f} := \sum_{i=1}^k f_i \delta_{\boldsymbol{x}_i}$ and $\boldsymbol{g} := \sum_{j=1}^l g_j \delta_{\boldsymbol{y}_j}$ be two such pmfs on the points $\{\boldsymbol{x}_i\}_{i=1}^k, \{\boldsymbol{y}_j\}_{j=1}^l \subset \mathbb{R}^d$ with $\sum_{i=1}^k f_i = \sum_{j=1}^l g_j = 1$. Let $\mathrm{Path}(\boldsymbol{f}, \boldsymbol{g})$ be all possible transport paths from $\boldsymbol{f}$ to $\boldsymbol{g}$ without cycles, i.e., each $G \in \mathrm{Path}(\boldsymbol{f}, \boldsymbol{g})$ is a weighted acyclic directed graph with $\{\boldsymbol{x}_i\}_i \cup \{\boldsymbol{y}_j\}_j \subset V(G)$, whose edge weights ($> 0$) satisfy *the Kirchhoff law* at each node

$v \in V(G)$:

$$\sum_{\substack{e\in E(G)\\ e^-=v}} w(e) = \sum_{\substack{e\in E(G)\\ e^+=v}} w(e) + \begin{cases} f_i & \text{if } v = \boldsymbol{x}_i \text{ for } \exists i \in 1:k \\ -g_j & \text{if } v = \boldsymbol{y}_j \text{ for } \exists j \in 1:l \\ 0 & \text{otherwise.} \end{cases}$$

Define the cost of a transport path $G \in \mathrm{Path}(\boldsymbol{f}, \boldsymbol{g})$ as:

$$\boldsymbol{M}_\alpha(G) := \sum_{e\in E(G)} w(e)^\alpha \,\mathrm{length}(e), \quad \alpha \in [0,1].$$

Xia proved that the minimum transportation cost

$$d_\alpha(\boldsymbol{f}, \boldsymbol{g}) := \min_{G\in\mathrm{Path}(\boldsymbol{f},\boldsymbol{g})} \boldsymbol{M}_\alpha(G)$$

is a *metric* on the space of pmfs and of homogeneous of degree α; and moreover he derived numerical algorithms to generate the α-optimal path for a given pair $(\boldsymbol{f}, \boldsymbol{g})$; see [8] and the refereces therein for further information.

4. OUR PROPOSED METHOD

We propose the following:

Algorithm 4.1 (LapEigPort)**.**

Step 0: *Convert $\boldsymbol{\phi}_i$ to a pmf $\boldsymbol{p}_i$ over a graph G, $i = 0 : n-1$.*

Step 1: *Compute the cost to transport $\boldsymbol{p}_i$ to $\boldsymbol{p}_j$ optimally, for all $i, j = 0 : n-1$, which results in a "distance" matrix $D = (D_{ij}) \in \mathbb{R}^{n\times n}_{\geq 0}$.*

Step 2: *Use some embedding technique, e.g.,* Multidimensional Scaling *(MDS), to embed D into a low dimensional Euclidean space $\mathbb{R}^{n_0}$, $n_0 \ll n$ (e.g., $n_0 = 2$ or 3).*

Step 3: *Examine how eigenvectors are placed and organized in that embedding space $\mathbb{R}^{n_0}$.*

In Step 0, we set $\boldsymbol{p}_i := \left(\phi_i^2[0], \ldots, \phi_i^2[n-1]\right)^\mathsf{T}$ in this paper; note that $\|\boldsymbol{\phi}_i\|_2 = 1$, $i = 0 : n-1$. This is the most natural and obvious way to convert $\boldsymbol{\phi}_i$ to a pmf; however, this may have some drawbacks and is certainly not the only way to convert $\boldsymbol{\phi}_i$ to a pmf as we will discuss later.

The key step of Algorithm 4.1 is Step 1, which needs further explanation. Unlike the general ROT setting, a graph G is fixed and given in our case. On the other hand, our graph G is *undirected* while the ROT requires *directed* graphs. Hence, we turn an undirected graph G into the *bidirected* graph $\tilde{\tilde{G}}$, a special type of directed multigraphs, i.e., each edge of G joining two nodes, say, i and j, becomes two directed edges (i, j) and (j, i) in $\tilde{\tilde{G}}$. To do so, we first compute the *incidence matrix* $Q = \left[\boldsymbol{q}_1 | \cdots | \boldsymbol{q}_m\right] \in \mathbb{R}^{n\times m}$ of the undirected graph $G = G(V, E)$ with $n = |V|$, $m = |E|$. Here, $\boldsymbol{q}_k$ represents the endpoints of the kth edge e_k: if e_k joins nodes i and j, then $\boldsymbol{q}_k[l] = 1$ if $l = i$ or j; otherwise $\boldsymbol{q}_k[l] = 0$. Then we orient each edge of G in an artibrary manner to form a directed graph $\tilde{G}$ and its incidence matrix $\tilde{Q} = \left[\tilde{\boldsymbol{q}}_1 | \cdots | \tilde{\boldsymbol{q}}_m\right] \in \mathbb{R}^{n\times m}$ where $\tilde{\boldsymbol{q}}_k$ is defined as follows: if $e_k = (i, j)$ for some i, j, then $\tilde{\boldsymbol{q}}_k[i] = -1$, $\tilde{\boldsymbol{q}}_k[j] = 1$, and $\tilde{\boldsymbol{q}}_k[l] = 0$ for $l \neq i, j$. Finally, form the bidirected graph $\tilde{\tilde{G}}$ with $\tilde{\tilde{Q}} := \left[\tilde{Q} \mid -\tilde{Q}\right] \in \mathbb{R}^{n\times 2m}$.

Given $\tilde{\tilde{Q}}$, we solve the *balance equation* that forces the Kirchhoff law:

$$\tilde{\tilde{Q}}\boldsymbol{w}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i, \quad \boldsymbol{w}_{ij} \in \mathbb{R}^{2m}_{\geq 0}. \tag{2}$$

The weight vector $\boldsymbol{w}_{ij}$ describes the transportation plan from $\boldsymbol{p}_i$ to $\boldsymbol{p}_j$. Note that $\boldsymbol{w}_{ij}[k] = 0$, $\exists k \in [0, 2m)$ implies that the transportation plan does not use the corresponding directed edge represented by the kth column of $\tilde{\tilde{Q}}$. Let $\tilde{\tilde{G}}_{ij}$ be the bidirected graph $\tilde{\tilde{G}}$ with these edge weights; then $\tilde{\tilde{G}}_{ij} \in \mathrm{Path}(\boldsymbol{p}_i, \boldsymbol{p}_j)$. We caution here that Eq. (2) may have multiple solutions. Hence, we propose to solve the following *Linear Programming* (LP) problem (see, e.g., [10]):

$$\min_{\boldsymbol{w}_{ij}\in\mathbb{R}^{2m}} \|\boldsymbol{w}_{ij}\|_1 \;\text{ subj. to: } \begin{cases} \tilde{\tilde{Q}}\boldsymbol{w}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i; \\ \boldsymbol{w}_{ij}[l] \geq 0, l = 0 : 2m-1, \end{cases} \tag{3}$$

to obtain one of the *sparse* solutions of Eq. (2), which turned out to be much faster to compute than using the nonnegative least squares (NNLS) solver [11, Chap. 23], which tends to generate denser solutions.

Finally we fill the distance matrix $D = (D_{ij})$ by

$$D_{ij} = \boldsymbol{M}_\alpha(\tilde{\tilde{G}}_{ij}) = \sum_{e\in E(\tilde{\tilde{G}}_{ij})} w_{ij}(e)^\alpha \,\mathrm{length}(e), \quad \alpha \in [0,1].$$

Note that Eq. (3) may still have multiple solutions, and currently we are *not* examining all possible solutions of Eq. (3) to search $\arg\min_{\tilde{\tilde{G}}_{ij}\in\mathrm{Path}(\boldsymbol{p}_i,\boldsymbol{p}_j)} \boldsymbol{M}_\alpha(\tilde{\tilde{G}}_{ij})$. We plan to investigate how to handle such multiple solutions in LP, e.g., the method proposed in [12].

5. NUMERICAL RESULTS

In this section, we will demonstrate the effectiveness of our proposed method using the same 2D lattice graph we discussed earlier as well as a dendritic tree of a retinal ganglion cell (RGC) of a mouse. We note that we used the `JuMP` optimization package [13] written in `Julia` [14] in order to solve the optimization problem Eq. (3).

5.1. The 2D lattice graph $P_7 \times P_3$

Fig. 3 shows the embedding of these 21 eigenvectors into $\mathbb{R}^2$ using Algorithm 4.1 where we set $\alpha = 0.5$ and used the classical MDS [15, Chap. 12] for embedding. Of course, in general, when a graph is given, we cannot assume the best embedding dimension n_0 a priori. Here we simply used $n_0 = 2$ because the top two eigenvalues of the Gram matrix of the configurations (i.e., the outputs of the MDS) were

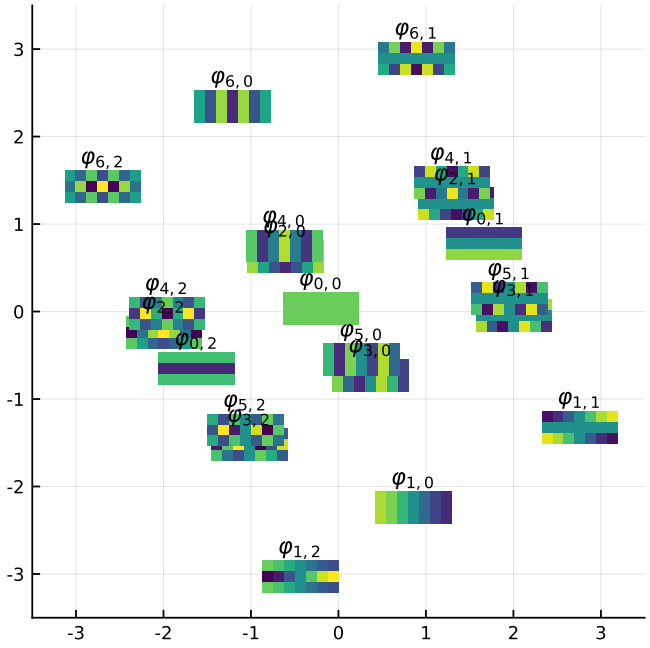


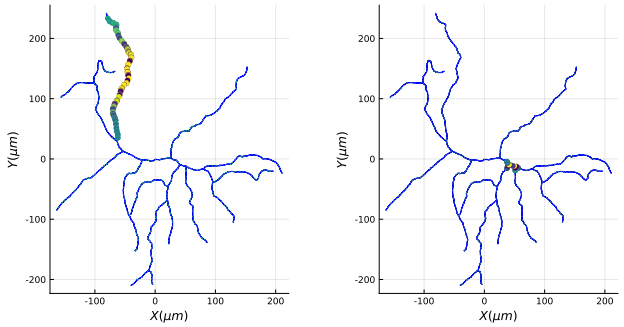
Fig. 3. Embedding of the Laplacian eigenvectors of $P_7 \times P_3$ into $\mathbb{R}^2$ using Algorithm 4.1 with $\alpha = 0.5$.

more than twice the third eigenvalue. Fig. 3 clearly reveals the two-dimensional organization of the eigenvectors, and is somewhat similar to a rotated version of Fig. 2, but the details are different. For example, the eigenvectors with even and odd oscillations are mapped in a symmetric manner around the "DC" vector $\boldsymbol{\varphi}_{0,0}$ located at the center of Fig. 3. This symmetry is due to the use of the energy of the eigenvector components in Step 0 of Algorithm 4.1: from Eq. (1), we have $\phi^2_{k;n}[x] + \phi^2_{n-k;n}[x] \equiv a^2_{k;n}$, $k = 1:n-1$, $x = 0:n-1$.

We also examined the case of $\alpha = 1$, which gave us more congested embedding around $\boldsymbol{\varphi}_{0,0}$.

5.2. The dendritic tree of an RGC of a mouse

For the details on the data acquisition and the conversion of dendritic trees of RGCs to graphs (in fact, literally "trees"), see [2]. As discussed in [3, 4] in greater detail, the graph Laplacian eigenvectors of these trees exhibit a very peculiar *phase transition phenomenon*: $\boldsymbol{\phi}_k$'s with $\lambda_k < 4$ oscillate *semi-globally* while those with $\lambda_k > 4$ are *concentrated* around junctions (i.e., nodes whose degrees are greater than 2). Fig. 4 shows 2D projections of two of the eigenvectors of a particular dendritic tree demonstrating this phenomenon. Fig. 5 shows the embedding of the Laplacian eigenvectors of this dendritic tree into $\mathbb{R}^3$ using Algorithm 4.1 with $\alpha = 0.5$. We used $n_0 = 3$ because of the size difference between the top three eigenvalues and the rest. These 3D point clouds form a upside-down bowl-like shape with three legs (or a shape partly joining two croissants). The magenta circle indicates the DC vector, $\boldsymbol{\phi}_0$,



(a) $\boldsymbol{\phi}_{1141}$ with $\lambda_{1141} = 3.9994$ (b) $\boldsymbol{\phi}_{1142}$ with $\lambda_{1142} = 4.3829$

Fig. 4. Laplacian eigenvectors before and after the phase transition. The yellow and purple circles indicate larger positive and negative components while those on thin blue lines are negligible. This tree has $n = 1154$ nodes in total.

while the cyan circle is the Fiedler vector, $\boldsymbol{\phi}_1$. The red circles located around the bottom are the eigenvectors $\boldsymbol{\phi}_k$, $k > 1141$, i.e., those localized around the junctions. The larger colored circles located in the lower right region are the eigenvectors supported on one of the branch indicated in Fig. 4(a). The gray scale colors of the other points indicate the magnitude of the corresponding eigenvalues.

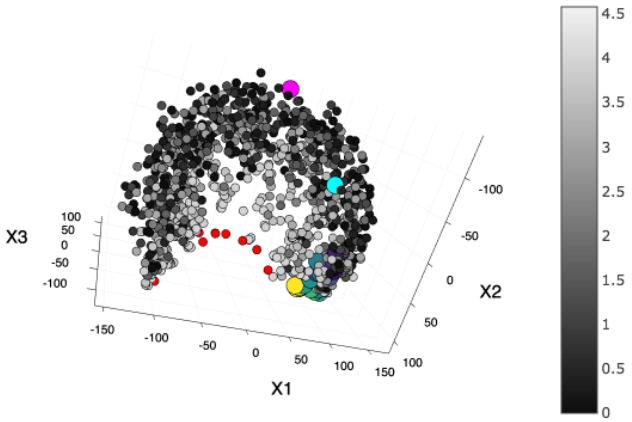


Fig. 5. Embedding of the Laplacian eigenvectors of the RGC tree into $\mathbb{R}^3$ using Algorithm 4.1 with $\alpha = 0.5$.

6. DISCUSSION

Although Algorithm 4.1 allows us to see interpretable organizations of the graph Laplacian eigenvectors, there remain several questions that need to be answered: 1) How can we choose optimal value of $\alpha \in [0,1]$? 2) Why the Fiedler vectors tend to be mapped far from the DC vectors? 3) How can we find the true cost-minimizing transportation path among Path$(\boldsymbol{p}_i, \boldsymbol{p}_j)$? 4) What other ways to turn $\boldsymbol{\phi}_i$ into $\boldsymbol{p}_i$ should we consider? 5) How can we develop the true Littlewood-Paley theory on graphs and most "natural" wavelets on graphs once the above embedding is done?

7. REFERENCES

[1] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harm. Anal.*, vol. 30, no. 2, pp. 129–150, 2011.

[2] N. Saito and E. Woei, "Analysis of neuronal dendrite patterns using eigenvalues of graph Laplacians," *JSIAM Letters*, vol. 1, pp. 13–16, 2009, Invited paper.

[3] N. Saito and E. Woei, "On the phase transition phenomenon of graph Laplacian eigenfunctions on trees," *RIMS Kôkyûroku*, vol. 1743, pp. 77–90, 2011.

[4] Y. Nakatsukasa, N. Saito, and E. Woei, "Mysteries around the graph Laplacian eigenvalue 4," *Linear Algebra Appl.*, vol. 438, no. 8, pp. 3231–3246, 2013.

[5] N. Saito, "Applied harmonic analysis on graphs and networks," *Bull. Jpn. Soc. Ind. Appl. Math.*, vol. 25, no. 3, pp. 6–15, 2015, in Japanese.

[6] J. Irion and N. Saito, "Applied and computational harmonic analysis on graphs and networks," in *Wavelets and Sparsity XVI, Proc. SPIE 9597*, M. Papadakis, V. K. Goyal, and D. Van De Ville, Eds., 2015, Paper # 95971F.

[7] S. Jaffard, Y. Meyer, and R. D. Ryan, *Wavelets: Tools for Science & Technology*, SIAM, Philadelphia, PA, 2001.

[8] Q. Xia, "Motivations, ideas and applications of ramified optimal transportation," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 49, no. 6, pp. 1791–1832, 2015, Special Issue – Optimal Transport.

[9] S. Kolouri, S. R. Park, M. Thorpe, D. Slepčev, and G. K. Rohde, "Optimal mass transport: Signal processing and machine-learning applications," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 43–59, 2017.

[10] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Inc., Upper Saddle River, NJ, 1982, Republished by Dover Publications, Inc. in 1998.

[11] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Inc., Upper Saddle River, NJ, 1974, Republished by SIAM in 1995.

[12] M. E. Dyer and L. G. Proll, "An algorithm for determining all extreme points of a convex polytope," *Math. Program.*, vol. 12, no. 1, pp. 81–96, 1977.

[13] I. Dunning, J. Huchette, and M. Lubin, "JuMP: A modeling language for mathematical optimization," *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017.

[14] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017.

[15] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, Springer, New York, 2nd edition, 2005.