EFFICIENT QUANTIZATION AND DENOISING USING LOCAL GRAPH FOURIER FRAMES

Philipp Reingruber and Gerald Matz

Institute of Telecommunications, Technische Universität Wien
Email: firstname.lastname@tuwien.ac.at

***Abstract*—In our recent work we have proposed a new class of graph signal expansion termed *local graph Fourier frames* (LGFF). LGFF have finite support in the vertex domain and hence entail computationally highly efficient signal analysis and synthesis algorithms. Furthermore, they are extremely flexible and can adapt to a multitude of graph signal types. In this paper, we discuss proof-of-concept approaches for the quantization and denoising of nonstationary graph processes based on LGFF. We furthermore propose an adaptation of the best basis algorithm to optimally choose the LGFF parameters. Our methods involve simple scalar processing in the LGFF domain and are shown to outperform existing approaches in spite of having a substantially lower complexity.**

I. Introduction

In recent years, the field graph signal processing (GSP) [1, 2] has evolved significantly as many theoretical and practical aspects have been studied [3–5]. GSP deals with data living on irregular domains represented by graphs. Since general graphs feature fewer symmetries compared to classical signal processing, many GSP algorithms suffer from high computational complexity and hence are limited to moderate graph sizes. This is particularly true for the graph Fourier transform (GFT), which is conceptually powerful but usually lacks a computationally attractive structure. Hence, various alternative graph signal transformations have been proposed [6–11], mostly exploiting analogies to conventional filter bank and wavelet constructions. All these proposals have in common that they are designed in the graph Fourier domain since the latter features a simple ordering like in the ordinary frequency domain. While some of the resulting filters and bases have been shown to be reasonably localized in the vertex domain, in general they are not sparse and do not have finite support.

In our recent work [12], To overcome the computational bottleneck in many GSP methods, we proposed a frame [13] whose elements have finite vertex-domain support named local graph Fourier frames (LGFF). LGFF amount to overcomplete bases that enable a local spectral analysis at low complexity. This paper showcases two potential practical applications of LGFF, namely graph signal compression and denoising. More specifically, our contributions are the following:

- We propose an LGFF-based transform coding scheme that outperforms GFT-based compression substantially in terms of performance and complexity.
- We devise a denoising scheme that builds on coefficient thresholding in the LGFF domain. While more efficient than GFT-based denoising, our method achieves superior noise suppression.
- To pick the optimal LGFF, we develop an adaptation of the best basis algorithm [14, 15] to graph partitions.
- We demonstrate via numerical experiments the pronounced performance advantage of LGFF-based quantization and denoising for signals that are not graph stationary (an assumption that is often contrived).

The paper is organization as follows. In Section II, we review the construction and properties of LGFF. Section III presents the proposed quantization and denoising methods. An adaption of the best basis algorithm is introduced in Section IV. In Section V we show numerical results and Section VI provides conclusions and an outlook.

II. LGFF

LGFF are a new approach to constructing orthogonal or overcomplete bases for graph signals. The construction of the basis $\mathcal{F}$ is similar in spirit to local Fourier bases [16] and lapped orthogonal transforms [17] known from harmonic analysis and conventional signal processing. The basis functions are well localized in the vertex domain and can be interpreted in terms of local Fourier analysis. These properties render LGFF intuitively meaningful and entail that the basis functions themselves and the associated analysis and synthesis operations can be computed in an efficient manner. We next review the key facts about LGFF, focusing on the special case of Parseval frames [13].

Consider a weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V} = \{1, \dots, N\}$ and edge set $\mathcal{E}$. The construct an LGFF, we partition the vertex set into disjoint sets $\mathcal{V}_k$, $k = 1, \dots, K$. The partitions can be obtained e.g. by suitable graph clustering methods [18, 19]. Let $\boldsymbol{\chi}_k$ denote the indicator function of $\mathcal{V}_k$, i.e., $\chi_k[n] = 1$ if $n \in \mathcal{V}_k$ and $\chi_k[n] = 0$ else. For each k, we design a non-negative window function $\mathbf{g}_k$ by smoothing (filtering, tapering) $\boldsymbol{\chi}_k$ using diffusion filters,

$$\mathbf{g}_k \odot \mathbf{g}_k = \sum_{l=0}^{L} \eta_l \widetilde{\mathbf{W}}^l \boldsymbol{\chi}_k.$$

Here, $\odot$ denotes the Hadamard product and $\widetilde{\mathbf{W}} = (\mathrm{Diag}\{\mathbf{W}\mathbf{1}\})^{-1}\mathbf{W}$ is the random walk weighted adjacency matrix of $\mathcal{G}$, with $\mathbf{W}$ denoting its weighted adjacency matrix. We impose the constraints $\eta_l \geq 0$ and $\sum_{l=0}^{L} \eta_l = 1$, which ensures that $\sum_{k=1}^{K} \mathbf{g}_k^2 = \mathbf{1}$. The window functions so designed feature a smooth roll-off and have a support that equals the L-hop neighborhood of $\mathcal{V}_k$,

$$\mathcal{S}_k = \{n\colon\ g_k[n] \neq 0\} = \mathcal{N}^L(\mathcal{V}_k).$$

Let by $\mathcal{G}_k$, $k = 1, \ldots, K$, be the subgraphs of $\mathcal{G}$ induced by $\mathcal{S}_k$ (these subgraphs usually overlap slightly) and denote by $\mathbf{U}_k = (\mathbf{u}_{kl})_{l=1,\ldots,N_k}$ the GFT basis associated with $\mathcal{G}_k$, where $N_k = |\mathcal{S}_k|$. The LGFF is then specified by $\mathbf{F} = (\mathbf{f}_{11}, \ldots, \mathbf{f}_{1N_1}, \mathbf{f}_{21}, \ldots, \mathbf{f}_{KN_K})$ with the basis vectors

$$\mathbf{f}_{kl} = \mathbf{g}_k \odot \mathbf{S}_k \mathbf{u}_{kl}, \qquad k = 1, \ldots, K, \;\; l = 1, \ldots, N_k,$$

where the matrices $\mathbf{S}_k \in \{0,1\}^{N \times N_k}$ map the length-N_k Laplacian eigenvectors $\mathbf{u}_{kl}$ of the subgraph $\mathcal{G}_k$ to length-N signals on $\mathcal{G}$ (zero padding). Since the support of $\mathbf{f}_{kl}$ is limited to $\mathcal{S}_k$ by construction, the LGFF basis vectors have finite support and thus typically very sparse. We show in [12] that $\mathbf{F}$ is indeed a parseval frame [13], $\mathbf{F}\mathbf{F}^T = \mathbf{I}$. In general the frame will be overcomplete, i.e., $M \geq N$, where $M = \sum_{k=1}^{K} N_k$ is the number of basis vectors. An orthogonal basis ($M = N$) can be obtained only without tapering ($\mathbf{g}_k = \boldsymbol{\chi}_k$).

While the Parseval frame is isometric, its basis functions are not normalized, i.e., the diagonal part $\mathbf{D}$ of the Gramian $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ is not equal to the identity matrix. For our purposes, it is preferable to use

$$\tilde{\mathbf{F}} = \mathbf{F}\mathbf{D}^{-1/2}, \quad \tilde{\mathbf{G}} = \mathbf{F}\mathbf{D}^{1/2}.$$

The frame $\tilde{\mathbf{F}}$ is normalized and $\tilde{\mathbf{G}}$ is the associated dual frame as can be seen from $\tilde{\mathbf{F}}\tilde{\mathbf{G}}^{\mathrm{T}} = \mathbf{F}\mathbf{D}^{-1/2}\mathbf{D}^{1/2}\mathbf{F}^{\mathrm{T}} = \mathbf{F}\mathbf{F}^{\mathrm{T}} = \mathbf{I}$.

III. Compression and Denoising

A. Transform-domain processing

In this section we explain how to use LGFF for the compression and denoising of a graph signal $\mathbf{x}$. In both cases our methods operate in the transform domain, i.e., they consist of an analysis stage,

$$\boldsymbol{\xi} = \tilde{\mathbf{G}}^{\mathrm{T}}\mathbf{x}, \quad \xi_{kl} = \|\mathbf{f}_{kl}\| \mathbf{f}_{kl}^{\mathrm{T}}\mathbf{x},$$

followed by scalar processing, $\hat{\xi}_{kl} = f_{kl}(\xi_{kl})$, and finally a synthesis stage,

$$\hat{\mathbf{x}} = \tilde{\mathbf{F}}\hat{\boldsymbol{\xi}} = \sum_{k=1}^{K} \sum_{l=1}^{N_k} \hat{\xi}_{kl} \frac{\mathbf{f}_{kl}}{\|\mathbf{f}_{kl}\|}$$

Note that for $f(\xi) = \xi$ we have $\hat{\mathbf{x}} = \mathbf{x}$. The key idea for both quantization and denoising in the transform domain is that the LGFF (approximately) decorrelates the signals of interest, thereby enabling scalar sample-by-sample processing without substantial performance penalty. At the same time, both analysis and synthesis can be done with $\sum_{k=1}^{K} N_k^2$ operations [12], which is typically much less than the N^2 operations required for a global (inverse) GFT.

B. Compression

With graph signal compression, our goal is to represent a noisefree signal $\mathbf{x} = \mathbf{s}$ with a small number of bits without causing too much distortion. We choose square distortion $d(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2$ as performance metric. With a focus on simplicity, we refrain from vector quantization and let the function $\hat{\xi}_{kl} = f_{kl}(\xi_{kl})$ represent a scalar quantizer (the quantization regions may be chosen differently for each coefficient ξ_{kl}). While the reproducer values $\tilde{\xi}_{kl}$ are used for signal reconstruction, binary representations of the reproducers are what is actually stored. In our experiments, we use either simple uniform quantizers or optimal quantizers designed via the Lloyd-Max algorithm [20]. The number of bits per coefficient is determined via the greedy bit allocation algorithm described in [21, Table 8.1]. The system thus obtained is similar in spirit to the JPEG standard for image compression, except that we use an LGFF instead of a DCT. Our method is orders of magnitude more efficient than the method proposed in [22], which is only feasible for graphs with a few tens of nodes.

C. Denoising

In the denoising problem, we consider a model where the signal $\mathbf{s}$ of interest is contaminated by i.i.d. zero-mean Gaussian noise with noise variance σ^2, i.e., $\mathbf{x} = \mathbf{s} + \mathbf{e}$. Here, the graph signal $\hat{\mathbf{x}}$ at the output of the LGFF synthesis stage serves as estimate of $\mathbf{s}$. As for the coefficient processing in the transform domain, we employ a simple hard thresholding operation (similar to [23, 24])

$$\hat{\xi} = f(\xi) = \begin{cases} 0, & |\xi| < \lambda, \\ \xi, & \text{else.} \end{cases}$$

This coefficient shrinkage zeros the small (probably noise-only) coefficients, thereby accomplishing the denoising. Since we use a normalized frame, the noise power is uniform in the transform domain and hence we can use the same threshold for all coefficients. In our experiments we optimized the threshold by means of a clairvoyant golden section search [25, Sec. 7.1] (yielding the threshold λ that maximizes the SNR in the estimate $\hat{\mathbf{x}}$).

IV. Best Basis Algorithm

In the previous section we assumed that the LGFF is predefined, which specifically assumes a given vertex partition $\{\mathcal{V}_1, \ldots, \mathcal{V}_K\}$. In this section we address the problem of determining an LGFF that is matched to a certain class of graph signals. Towards this end, we propose a graph adaptation of the best basis algorithm proposed for conventional signals in [14, 15]. The algorithm builds on a hierarchical dyadic graph partition $\mathcal{V}_k^{(i)} \subseteq \mathcal{V}$, $i = 0, \ldots, I$, $k = 1, \ldots, 2^i$, such that $\bigcup_{k=1}^{2^i} \mathcal{V}_k^{(i)} = \mathcal{V}$ and

$$\mathcal{V}_k^{(i)} = \mathcal{V}_{2k-1}^{(i+1)} \cup \mathcal{V}_{2k}^{(i+1)}. \tag{1}$$

One way to obtain such a hierarchical partition is via recursive binary spectral clustering (successive refinement). The subset $\mathcal{V}_k^{(i)}$ can be represented as node k in layer i of a binary tree whose root node is $\mathcal{V}_1^{(0)} = \mathcal{V}$. By this construction, every proper subtree (where each node is either a leaf or has two child nodes) defines a partition of $\mathcal{V}$. Each of these partitions can be obtained by starting at the finest partition at the deepest layer I and suitably combining vertex sets to obtain coarser partitions at higher layers. For example with $I = 2$ we could have $\{\mathcal{V}_1^{(1)}, \mathcal{V}_3^{(2)}, \mathcal{V}_4^{(2)}\} = \{\mathcal{V}_1^{(2)} \cup \mathcal{V}_2^{(2)}, \mathcal{V}_3^{(2)}, \mathcal{V}_4^{(2)}\}$. To each

valid partition we can associate an LGFF and the property (1) can be shown to translate to

$$\mathbf{g}_k^{(i)} \odot \mathbf{g}_k^{(i)} = \mathbf{g}_{2k-1}^{(i+1)} \odot \mathbf{g}_{2k-1}^{(i+1)} + \mathbf{g}_{2k}^{(i+1)} \odot \mathbf{g}_{2k}^{(i+1)},$$

which ensures that the coarsening operation indeed induces another valid LGFF.

Using a cost function $S(\mathbf{F})$ to assess the performance of an LGFF $\mathbf{F}$, the best basis algorithm proceeds according to a bottom-up strategy as follows:

1) Initialization at layer I: determine the finest partition $\{\mathcal{V}_1^{(I)}, \ldots, \mathcal{V}_{2^I}^{(I)}\}$ and the associated LGFF and compute the associated cost function;

2) Recursion at layer i: for each node k at layer i determine whether the LGFF associated with the coarsening (1) has a lower cost; if yes, replace $\mathcal{V}_{2k-1}^{(i+1)}$ and $\mathcal{V}_{2k}^{(i+1)}$ by $\mathcal{V}_k^{(i)}$; otherwise keep $\mathcal{V}_{2k-1}^{(i+1)}$ and $\mathcal{V}_{2k}^{(i+1)}$.

Once the iterations terminate, we obtain the LGFF achieving the minimum cost (best basis). In the context of quantization and denoising it is desirable to find a basis that achieves low correlation in the coefficient domain. Thus, we here use a cost function that effectively is a weighted norm of the off-diagonal elements of the coefficient covariance matrix $\mathbf{C}_\xi = \tilde{\mathbf{G}}^{\mathrm{T}} \mathbf{C}_x \tilde{\mathbf{G}}$. If adapted to varying subgraph sizes, the An alternative cost function is given by the coefficient entropy used in [14, 15].

V. Numerical results

In this section, we illustrate the performance of our proposed LGFF quantization and denoising schemes. We compare our results to a baseline method that uses the same optimized coefficient processing (quantization or thresholding) in the graph Fourier domain.

A. Simulation setup

We consider random geometric graphs with $N = 1024$ nodes. The graph signals were nonstationary and obtained as $\mathbf{s} = \tilde{\mathbf{F}}\tilde{\boldsymbol{\xi}}$ with uncorrelated Gaussian coefficients $\tilde{\boldsymbol{\xi}}$ and an untapered LGFF $\tilde{\mathbf{F}}$ ($L = 0$). The coefficients had an exponentially decaying power profile with bandwidth B_k within each subgraph but distinct total power across subgraphs. We used two underlying graph partitions: a coarse partition with $K = 4$ subgraphs (average subgraph size 256) and a fine partition with $K = 18$ subgraphs (average subgraph size 57). Note that finer graph partitions entail less stationary graph signals.

For the compression and denoising itself we used a dictionary of smooth tapered LGFF ($L = 1$). Out of this dictionary, the optimal LGFF was determined using the best basis algorithm from Section IV. The performance metrics shown have been obtained by averaging over $5N$ signal realizations and 10 graph realizations.

Before discussing the performance metrics, we show the measured average computational speedup achieved by the LGFF-based quantization and denoising approaches relative to the GFT-based methods (see Table I). Clearly, LGFF schemes are much faster than GFT schemes. The speedup for compression is generally larger than with denoising. This can be explained by the fact that hard thresholding in the GFT domain leaves only a small nonzero number of Fourier coefficients and hence a rather fast inverse GFT. It is also seen that the speedup is more pronounced for fine partitions; this is expected since many small local Fourier transforms are much faster than one global Fourier transform.

	Compression		
Partition	$R = 3$	$R = 5$	Denoising
Coarse	12.61	12.15	1.64
Fine	42.97	38.02	7.32

Table I: Speedup of LGFF-based approaches relative to GFT-based methods for various graph partitions and tasks.

B. Quantization

For signal compression, we chose the local signal bandwidths as $B_k = 0.8N_k$. We varied the rate, defined as the number of bits per nonzero signal coefficient, as $R \in \{1, ..., 8\}$ (this amounts to $R\sum_{k=1}^{K} B_k/N = 0.8R$ bits per vertex sample) and measured the associated distortion achieved by the various methods. The roll-off of the LGFF windows was chosen as $\eta_1 = 1-\eta_0 = \frac{1}{3}$. For the uniform quantizer, the support was chosen as the interval between $\pm 2\sqrt{E\{\xi_{kl}^2\}}$. As the ultimate performance benchmark, we use the rate-distortion function for Gaussian vector quantization [26, Sec. 10.3.3].

Fig. 1 shows the resulting rate-distortion trade-offs (instead of distortion we plot the reconstruction signal-to-noise ratio (SNR)) for both coarse (left plot) and fine (right plot) graph partitions. Clearly, LGFF-based compression outperforms GFT-based compression (in spite of its lower complexity), with the performance advantage being more pronounced for the fine partition.

Generally, uniform quantization tends to saturate at an SNR of 20 dB, probably due to outliers. Higher SNRs are achieved only with LGFF-based Lloyd-Max quantization. There is still a gap to the theoretical optimum since we do not perform vector quantization and the graph signals have large (local) bandwidths leading to non-sparse LGFF coefficients. For finer graph partitions, the GFT coefficients are increasingly correlated and hence even with optimal quantization its SNR is poorer than that of uniform LGFF-domain compression.

C. Denoising

For denoising, we used identical local bandwidths $B_k = B$ in our signal model. We chose $B = 10$ for the scenario with the coarse graph partition ($K = 4$) and $B = 3$ for the scenario with the fine graph partition ($K = 18$), resulting in 40 and 54 nonzero signal coefficients, respectively. The LGFF were constructed using a slightly steeper roll-off of $\eta_1 = 1-\eta_0 = \frac{1}{6}$.

In Fig. 2, we show plots of the average SNR improvement $E\{\|\mathbf{x} - \mathbf{s}\|^2\}/E\{\|\hat{\mathbf{x}} - \mathbf{s}\|^2\}$ in dB versus the input SNR $E\{\|\mathbf{s}\|^2\}/(N\sigma^2)$ for both LGFF-based and GFT-based denoising for the case of the coarse partition (left) and the fine partition (right). Again, the superiority of our LGFF

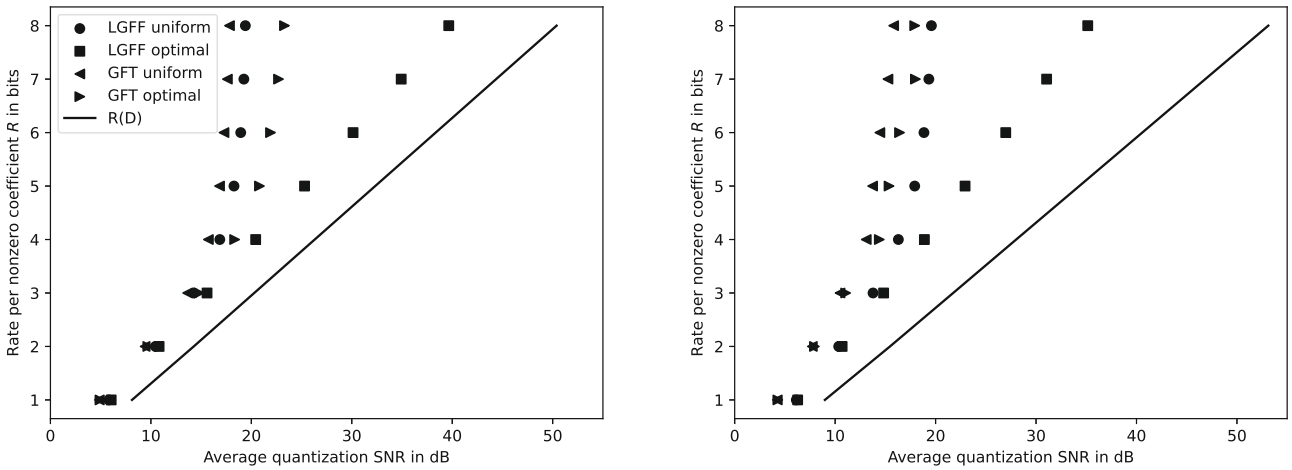

Figure 1: *Rate per nonzero coefficient during signal generation over average SNR for uniform and optimal scalar quantization in LGFF and GFT domain (left: coarse partition, right: fine partition).*

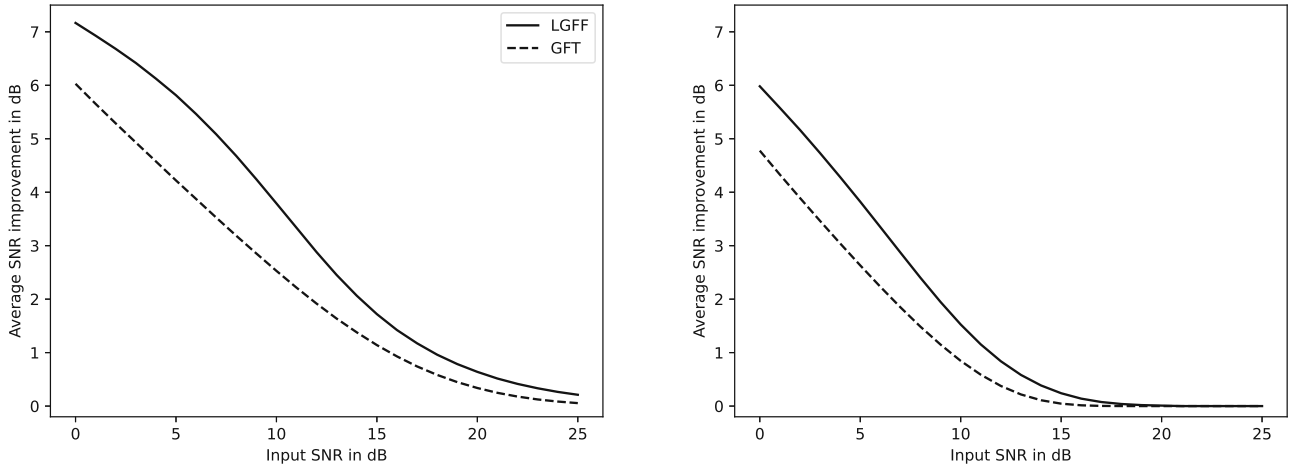

Figure 2: *Average denoising SNR improvement versus input SNR in LGFF and GFT domain. Left: coarse partition, local bandwidth $B = 10$; right: fine partition, local bandwidth $B = 3$.*

approach is evident, specifically at low-to-medium input SNR, notwithstanding its lower computational complexity.

Signals obtained with the finer graph partition are less stationary, thus deteriorating the GFT performance. However, in this case the window tapering also causes stronger oversampling in the LGFF, which in turn results in stronger residual correlation in the LGFF coefficients, thus also limiting the performance of the LGFF denoising approach.

VI. Conclusion

In this paper, we demonstrated the practical usefulness of local graph Fourier frames (LGFF) for two important graph signal processing tasks, namely compression and denoising. Our schemes consist of simple scalar coefficient processing sandwiched between efficient LGFF analysis and synthesis stages. We have seen that compared to global GFT-based processing our LGFF-based processing has superior performance while at the same time being substantially faster.

For realistic graph signals that violate the common stationarity assumption, the performance advantage of LGFF domain processing can be explained by the fact that the LGFF achieves a good decorrelation of the transform coefficients. This decorrelation can be promoted by using a best basis selection scheme that matches the LGFF to the signal properties.

In our future work, we plan to study the performance of LGFF-based GSP on real-world large-scale data sets, where computational efficiency is of paramount importance. Furthermore, we will investigate in more depth the efficient construction of LGFF and applications to other GSP tasks like and signal interpolation and detection.

REFERENCES

[1] 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," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[2] A. Sandryhaila and J. M. F. Moura, "Big Data Analysis with Signal Processing on Graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, Sep. 2014.

[3] T. Adali and A. O. (Eds.), "Applications of graph theory," *Proc. IEEE*, vol. 106, no. 5, 2018.

[4] L. Stanković, D. Mandic, M. Daković, M. Brajović, B. Scalzo, S. Li, and A. G. Constantinides, "Data analytics on graphs, Parts I-III," in *Foundations and Trends in Machine Learning*. NOW, 2020, vol. 13.

[5] A. Ortega, *Introduction to Graph Signal Processing*. Cambridge (UK): Cambridge Univ. Press, 2022.

[6] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[7] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2786–2799, June 2012.

[8] N. Tremblay and P. Borgnat, "Subgraph-based filterbanks for graph signals," *IEEE Trans.*, vol. 64, no. 15, pp. 3827–3840, 2016.

[9] B. de Loynes, F. Navarro, and B. Olivier, "Localized Fourier analysis for graph signal processing," *Applied and Computational Harmonic Analysis*, vol. 57, pp. 1–26, 2022.

[10] H. Behjat, U. Richter, D. Van De Ville, and L. Sörnmo, "Signal-adapted tight frames on graphs," *IEEE Trans. Signal Process.*, vol. 64, no. 22, pp. 6017–6029, 2016.

[11] D. I. Shuman, C. Wiesmeyr, N. Holighaus, and P. Vandergheynst, "Spectrum-adapted tight graph wavelet and vertex-frequency frames," *IEEE Trans. Signal Process.*, vol. 63, no. 16, pp. 4223–4235, 2015.

[12] P. Reingruber and G. Matz, "Tight Local Graph Fourier Frames with Finite Support," in *Proc. Asilomar Conf. Signals, Systems and Computers* (accepted for publication), 2024.

[13] O. Christensen, *An Introduction to Frames and Riesz Bases*. Boston (MA): Birkhäuser, 2003.

[14] R. R. Coifman and M. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 713–718, March 1992.

[15] M. V. Wickerhauser, "Fast approximate factor analysis," in *Curves and Surfaces in Computer Vision and Graphics II*, vol. 1610, 1992, pp. 23 – 32.

[16] R. R. Coifman and Y. Meyer, "Remarques sur l'analyse de Fourier à fenêtre," *C. R. Acad. Sci.*, vol. 312, no. 1, pp. 259–261, 1991.

[17] H. S. Malvar, "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 969–978, June 1990.

[18] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[19] X. Bresson, T. Laurent, D. Uminsky, and J. H. von Brecht, "Multiclass total variation clustering," in *Proc. Int. Conf. Neural Inf. Process. Systems (NIPS)*, Lake Tahoe (NV), Dec. 2013, pp. 1421–1429.

[20] J. Max, "Quantizing for minimum distortion," *IEEE Transactions on Information Theory*, vol. 6, no. 1, pp. 7–12, 1960.

[21] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston: Kluwer, 1992.

[22] P. Li, N. Shlezinger, H. Zhang, B. Wang, and Y. C. Eldar, "Graph signal compression by joint quantization and sampling," *IEEE Transactions on Signal Processing*, vol. 70, no. 9, pp. 4512–4527, Sep. 2022.

[23] P. Moulin, "Signal estimation using adapted tree structured bases and the MDL principle," in *Proc. IEEE-SP Int. Sympos. Time-Frequency Time-Scale Analysis*, Paris, France, June 1996, pp. 141–143.

[24] D. L. Donoho and I. M. Johnstone, "Minimax estimation via wavelet shrinkage," *Ann. Statist.*, vol. 26, no. 3, pp. 879–921, 1998.

[25] E. K. P. Chong and S. H. Żak, *An introduction to optimization*, 2nd ed. New York: Wiley, 2001.

[26] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New Jersey: Wiley, 2006.