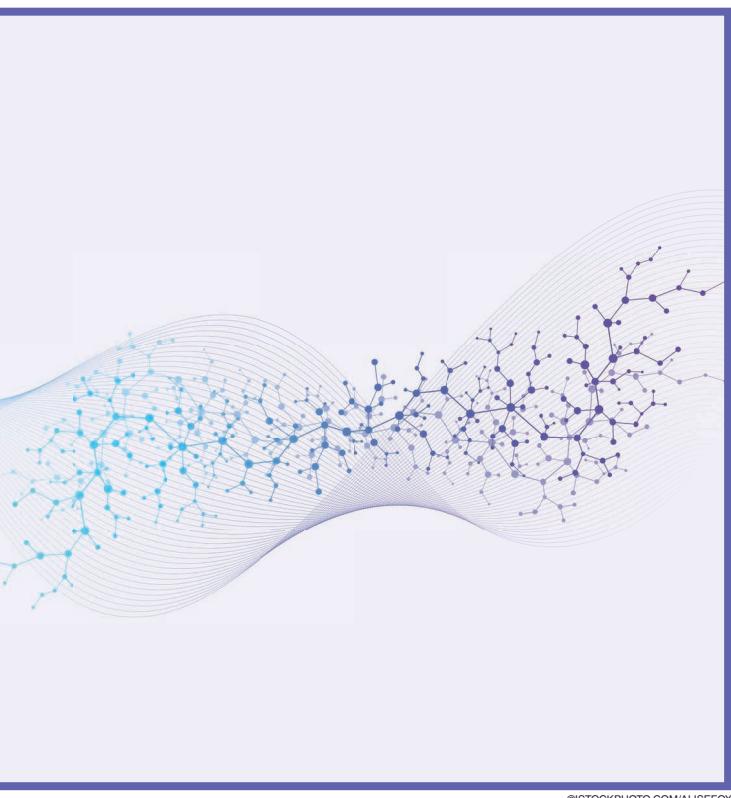


Daniel L. Lau, Gonzalo R. Arce, Alejandro Parada-Mayorga,  
Daniela Dapena, and Karelia Pena-Pena

# Blue-Noise Sampling of Graph and Multigraph Signals

*Dithering on non-Euclidean domains*



With the surge in the volumes and dimensions of data defined in non-Euclidean spaces, graph signal processing (GSP) techniques are emerging as important tools in our understanding of these domains [1]. A fundamental problem for GSP is to determine which nodes play the most important role; so, graph signal sampling and recovery thus become essential [2]. In general, most of the current sampling methods are based on graph spectral decompositions where the graph Fourier transform (GFT) plays a central role [2]. Although adequate in many cases, they are not applicable when the graphs are large and where spectral decompositions are computationally difficult [3]. After years of beautiful and useful theoretical insights developed in this problem, the interest has now centered on finding more efficient methods for the computation of good sampling sets. Looking to the spatial domain for inspiration, substantial research has been performed that looks at the use of spatial point processes to define stochastic sampling grids with a particular interest at point processes that generate “blue noise.”

The term *blue noise* was first coined by Ulichney [4] in the field of halftoning to describe a homogeneous distribution of printed dots used by inkjet printers to reproduce shades of gray that are spaced as far apart as possible, having a power spectra dominated by high-frequency energy where blue is the high-frequency component of white light [5]. Minimizing low-frequency or red energy creates dot patterns that are least visible to the human visual system, and by understanding the relationship between interdot spatial characteristics to desirable spectral properties, an algorithm developer can focus on optimizing these spatial characteristics instead of spectral. And because these arrangements also define good stochastic sampling patterns as well as sampling sets on graphs [6], [7], it is expected that vertex-domain sampling algorithms exist for graphs that are computationally efficient and applicable to large graphs where Fourier-based methods are impractical.

Concepts similar to blue noise in relation to graph sampling include the early work of Stadler [8], who developed spectral measures to quantify the correlation between the values of

Digital Object Identifier 10.1109/MSP.2020.3014070  
Date of current version: 28 October 2020

signals on the vertex domain for some families of graphs. Also included is the work of Pesenson [9], who defined sampling patterns based on the covering of a manifold by means of open sets of the same size, while de Silva and Ghrist [10] used a similar concept of disk coverings to capture homological properties with graphs. Tremblay [11] proposed graph sampling with a point process having a repulsion property that minimized the possibility of nearest neighboring samples existing within a predefined range, all based on the approach proposed by Puy et al. [12] who introduced blue-noise-like random sampling strategies based on the local graph properties.

Lacking a single, unifying framework for good graph sampling with low computational complexity, it is the goal of this article to translate the concept of blue noise for spatial-domain sampling onto graphs such that the attributes of the algorithms of halftoning can be justifiably employed in the vertex domain as an alternative to methods that require the calculation of eigenvalues and eigenvectors. A fascinating aspect of this article is that it draws from various disciplines within signal processing, spectral graph theory, and from digital halftoning. Additionally, the concept of admissible partitions proposed in [13], although theoretical, put the seeds for the development of diffusion sampling methods based on blue-noise sampling principles. As error diffusion is one of the most successful and computationally efficient blue-noise halftoning algorithms in the spatial domain [14], we expect it to become a standard technique for graphs.

Beyond unidirectional, connected graphs, this article also considers multigraphs, where signals can be defined and processed exploiting concepts from GSP but where a unique set of challenges arises from the intrinsic characteristics of multigraphs. Without a clear definition of frequency, sampling algorithms proposed for simple graphs like those of Anis et al. [15] do not have multigraph variants. Yet even though there is no clear understanding of blue noise in the spectral domain, the vertex-domain definition of blue noise as a process that minimizes clustering of sampling nodes is more easily extended to nonsimple graphs, such as multigraphs. Within the context of digital printing, we hypothesize that multigraphs are similar to color halftones composed of cyan, magenta, yellow, and black inks where the individual halftones observe blue-noise properties as well as the composite color halftoned when viewed in the monochrome, luminance space [16]. As such, this article demonstrates a series of modifications to our graph algorithms to also sample multigraphs where we extend the idea of jointly blue noise to multigraph sampling.

## Background and notations

Before introducing the details about blue-noise sampling, we note that the graphs considered in this discussion are undirected, weighted, connected, and simple. They are represented by  $G = (V(G), E(G))$ , where  $V(G)$  is the set of nodes and  $E(G)$  the set of edges. The weights associated to the edges in  $E(G)$

are stored in the symmetric matrix  $\mathbf{W}$ , being  $\mathbf{W}(u, v) \geq 0$ , the weight associated to the edge connecting the nodes  $u$  and  $v$ . The diagonal matrix  $\mathbf{D}$  stores the degrees of the nodes in the graph, and its entries are given by  $\mathbf{D}(u, u) = \sum_{v \in V(G)} \mathbf{W}(u, v)$ . The values of  $\mathbf{D}$  can be used to characterize any graph  $G$  by its volume as  $\text{vol}(G) = \sum_{u \in V(G)} \mathbf{D}(u, u)$ . On any graph, several graph shift operators can be considered to exploit different properties of the graph and to define a GSP framework [1], [17]. For our discussion, we use the combinatorial graph Laplacian  $\mathbf{L} = \mathbf{D} - \mathbf{W}$ , which is a symmetric-positive semidefinite matrix whose eigenvalues are represented as  $0 = \mu_1 \leq \mu_2 \leq \dots \leq \mu_N$ ,  $N = |V(G)|$ . A path between the nodes  $u_1$  and  $u_q$  is given by a sequence of nodes  $\{u_i\}_{i=1}^q$  such that  $\mathbf{W}(u_i, u_{i+1}) \neq 0$  for all  $1 \leq i \leq q-1$ , and its length is given by  $\sum_{i=1}^{q-1} |\mathbf{W}(u_i, u_{i+1})|$ . Then, the distance (geodesic) between  $u_1$  and  $u_q$  is defined as the minimum length of all the paths between  $u_1$  and  $u_q$ ; we store the values of these distances in the matrix  $\Gamma$ , where  $\Gamma(u, v)$  is the distance between the nodes  $u$  and  $v$ .

A real-valued signal  $\mathbf{x}$  on the graph  $G$  is defined as a function  $\mathbf{x} : V(G) \rightarrow \mathbb{R}$ , which is associated to a vector  $\mathbf{x} \in \mathbb{R}^N$ . The component  $\mathbf{x}(v)$  represents the value of the signal on the node  $v \in V(G)$ . The support of

$\mathbf{x}$ , defined as the set of nodes where  $\mathbf{x}$  is different from zero, is indicated by  $\text{supp}(\mathbf{x})$ , and the restriction of  $\mathbf{x}$  on a subset of nodes  $S$  is represented by  $\mathbf{x}(S)$ . Considering the spectral decomposition of the Laplacian as  $\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^\top$ , the GFT of a graph signal  $\mathbf{x}$  on  $G$  is defined as  $\hat{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}$  [1]. The bandwidth of  $\mathbf{x}$  is defined in terms of the nonzero components of  $\hat{\mathbf{x}}$ . In particular, the bandwidth of  $\mathbf{x}$  on the spectral axes is given by  $\omega \in \mathbb{R}_+$  if  $\hat{\mathbf{x}} \in PW_\omega(G) = \text{span}\{\mathbf{U}_k : \mu_k \leq \omega\}$ , where  $PW_\omega(G)$  is the so-called Paley–Wiener space of bandwidth  $\omega$  [18], and  $\mathbf{U}_k$  is the matrix whose columns are the first  $k$  columns of  $\mathbf{U}$ . An alternative representation of the bandwidth is given by the largest integer  $k$  such that  $\mu_k \leq \omega$ . When considering the realizations  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q$  of a random signal  $\mathbf{x}$ , the power spectrum of  $\mathbf{x}$  is computed as  $\mathbf{p}(\ell) = (N/q) \sum_{i=1}^q \hat{\mathbf{x}}_i(\ell)^2 / \|\hat{\mathbf{x}}_i\|^2$ .

The notion of bandwidth on a graph is based on the fact that any signal on the graph can be represented by means of its nonzero coefficients in the GFT, which states a minimum number of values of the signal for its unique representation. In this context, we consider the sampling of a signal  $\mathbf{x}$  on the graph  $G$  by selecting the components of  $\mathbf{x}$  on a subset of nodes  $S = \{s_1, \dots, s_m\} \subset V(G)$ . These values are represented by  $\mathbf{x}(S) = \mathbf{M}\mathbf{x}$ , where  $\mathbf{M}$  is a binary sampling matrix. A reconstructed version of  $\mathbf{x}$  can be obtained from  $\mathbf{x}(S)$  as

$$\mathbf{x}_{\text{rec}} = \underset{z \in \text{span}(\mathbf{U}_k)}{\text{argmin}} \|\mathbf{M}z - \mathbf{x}(S)\|_2^2 = \mathbf{U}_k(\mathbf{M}\mathbf{U}_k)^\dagger \mathbf{x}(S), \quad (1)$$

where  $(\mathbf{M}\mathbf{U}_k)^\dagger$  is the Moore–Penrose pseudoinverse of  $\mathbf{M}\mathbf{U}_k$  [6], [15]. The closeness between  $\mathbf{x}$  and  $\mathbf{x}_{\text{rec}}$  is directly dependent on the choice of the sampling set  $S$ ; therefore, the central challenges in sampling are two. First, given a fixed value of the bandwidth of the signals, it is desired to find the subset of

sampling nodes  $S$  such that the reconstruction in (1) provides the minimum error, and second, given a fixed number of sampling nodes, it is desired to find the subset of nodes  $S$ , which extends or maximizes the bandwidth of the signals that can be represented in a unique way on  $S$ .

In [18], some quantities have been defined to characterize the goodness of sampling sets, relying on the concept of removable sets. In particular, any subset of nodes  $S \subset V(G)$  is said to be a  $\Lambda$ -removable set if

$$\| \mathbf{x} \|_2 \leq (1/\Lambda) \| \mathbf{Lx} \|_2 \quad \forall \mathbf{x} \in L_2(S), \quad (2)$$

where  $L_2(S)$  is the set of all signals  $\mathbf{x}$ , with support in  $S \subset V(G)$  and finite  $\ell_2$  norm. The highest-value  $\Lambda$  for which (2) holds, is represented as  $\Lambda_S$ . This can be associated to the quality of a given sampling set. Indeed, as stated in [18], if for a set  $S \subset V(G)$ , it happens that  $S^c = V(G) \setminus S$  is a  $\Lambda_{S^c}$ -removable set, then all signals in  $PW_\omega(G)$  are completely determined by their values in  $S$  whenever  $0 < \omega < \Lambda_{S^c}$ . As such,  $\Lambda_{S^c}$  provides a measure for the quality of the set  $S$  as a sampling set. In [13], other parameters connected to  $\Lambda_{S^c}$  are introduced and a detailed discussion about them can be found in [6], where it is also shown that sets of nodes with large values of  $\Lambda_{S^c}$  are desired not only for the uniqueness of the representation but also for a high-quality reconstruction.

Different approaches have been considered in the literature for the search of optimal sampling sets [6], [15], [17]. Most of these approaches rely on the minimization or maximization of a cost function that is directly or indirectly related to the values of  $\Lambda_{S^c}$  as a consequence of the fact that finding the subset of nodes that provides the maximum value of  $\Lambda_{S^c}$  is a combinatorial NP-hard problem. The different cost functions are based on the minimization of the error in the reconstruction [6] achieved by means of greedy algorithms that add one node at a time. A detailed discussion of the recent methods can be found in [6] considering different types of sampling schemes and optional reconstruction methods. Illustrated in “Example 1” is one such sampling method from [15] that relies on the systematic calculation of the first eigenvector and eigenvalue of a Laplacian matrix indexed by the nodes in the complement set of a temporary sampling set  $S$ . In particular, denoting by  $\mu_1(\mathbf{L}_{S^c, S^c})$  the first eigenvalue and  $\mathbf{u}_1$  its associated eigenvector of the matrix obtained from  $\mathbf{L}$  deleting the rows and columns indexed by  $S$ , an optimal sampling set  $S$  is obtained as  $S = S \cup \{v\}$ , where  $v$  corresponds to the index of the component of  $\mathbf{u}_1$  with maximum absolute value. This process starts with  $S = \emptyset$  and repeats the desired number of sampling points.

Also illustrated in “Example 1” are the sampling methods proposed in [12] and [11]. In [12],  $m$  samples are drawn independently according to the sampling distribution  $\mathbf{p}^* \in \mathbb{R}^N$ , which minimizes the graph-weighted coherence given by  $\mathbf{p}_i^* = \|\mathbf{U}_k^T \mathbf{\epsilon}_i\|_2 / \sum_{i=1}^N \|\mathbf{U}_k^T \mathbf{\epsilon}_i\|_2$ , where  $\mathbf{\epsilon}_i$  is the  $N$ -dimensional Kronecker column vector centered at  $i$ . Tremblay et al. in [19],

on the other hand, introduced a way to find the optimal distribution  $\mathbf{p}^*$  based on determinantal point processes, considering sampling sets of size  $m = k$  when  $\mathbf{U}_k$  is available. An efficient implementation of this sampling technique was later developed in [11]. The recovery of the signal for the methods in [11] and [12] is given by

$$\mathbf{x}_{\text{rec}} = \underset{\mathbf{z} \in \text{span}(\mathbf{U}_k)}{\text{argmin}} \| \mathbf{P}^{-1/2} (\mathbf{Mz} - \mathbf{x}(S)) \|_2^2, \quad (3)$$

where  $\mathbf{P} = \text{diag}(\mathbf{p}^*)$ . Note that (3) can be considered a particular instantiation of (1), where the effective sampling matrix is  $\mathbf{P}^{-1/2} \mathbf{M}$ .

### Blue-noise graph sampling

As an alternative to deriving eigenvectors and values, blue-noise sampling on graphs provides a framework for the efficient generation of sampling sets relying on the simplicity of generating vertex-domain blue-noise-like patterns using algorithms originally derived for halftoning. Although blue-noise sampling can be described by means of simple and intuitive ideas, its connection with formal theoretical results about graph sampling [6] provides a solid ground for the development of simple, efficient, and fast algorithms with consequences in applications where large networks are considered. As various studies have looked at how to convert a continuous-tone image into a blue-noise halftone, we similarly are interested in algorithms for building blue-noise sampling patterns on graphs.

As stated previously, Ulichney [4] demonstrated that the best halftoning algorithms are the ones that maximize the high or blue-noise frequencies or, conversely, minimize the low or red-noise frequencies; however, doing so is much easier, computationally, in the spatial domain assuming that we know the spatial properties corresponding to minimizing a pattern’s redness. Likewise, Parada-Mayorga et al. [6] show that good graph sampling is defined by binary graph signals whose GFT also minimizes *redness*, which they define as a weighted measure of the signal’s total energy according to

$$R_s = \frac{1}{\| \hat{s} \|_2^2} \sum_{\ell=2}^N \frac{\hat{s}^2(\ell)}{\mu_\ell} = \frac{1}{m} \sum_{\ell=2}^N \frac{\hat{s}^2(\ell)}{\mu_\ell}, \quad (4)$$

where  $\hat{s}$  is the GFT of  $s$ , and the square of each coefficient is weighted by the inverse of the corresponding eigenvalue/frequency. In this way, the lowest frequency components have much greater weight than do the highest. In the vertex domain, these binary signals have intersampling node-spatial properties, which are characterized as a homogeneous distribution of sampling nodes spaced as far apart as possible while also uniformly spanning the area of the graph at a given density. To illustrate this, “Example 2” compares well-formed blue noise with uncorrelated white-noise patterns generated by associating to each node a Bernoulli random variable with a probability of success of  $p = m/N$ .

## Example 1

The void-and-cluster (VAC) and error-diffusion methods have the advantage that spectral decompositions are not required, whereas the other used methods do. The patterns generated by adaptive VACs are similar to those generated using spectral decomposition (see Figure S1).

To compare the performance of various sampling methods, a numerical experiment using the 2,642-node Minnesota graph was generated. One hundred band-limited signals with bandwidths of  $0.05|V(G)|$  were generated having additive Gaussian noise with a signal-to-noise ratio of 20. An example of these signals is depicted in Figure S2(a). The signals were then sampled and reconstructed using different methods according to the sampling nature; deterministic methods used (1) for the reconstruction, while probabilistic

methods (Puy et al. [12] and Tremblay [11]) used (3). The mean-squared errors (MSEs) between the reconstructed and original signals were compared for all the methods [see Figure S2(b)].

A VAC with  $\alpha = 1.5$  has an MSE that quickly converges to the sampling patterns of Anis et al. [15] and Tremblay et al. [11] and performs substantially better than random sampling. A VAC with  $\alpha = 0$  produces a uniform spatial distribution, independent of the underlying graph structure; in the case of the Minnesota graph, this causes a poor performance. Even though error diffusion does not outperform a VAC with  $\alpha = 1.5$ , it offers lower computational complexity, and its performance is comparable to the method proposed by Puy et al. [12].

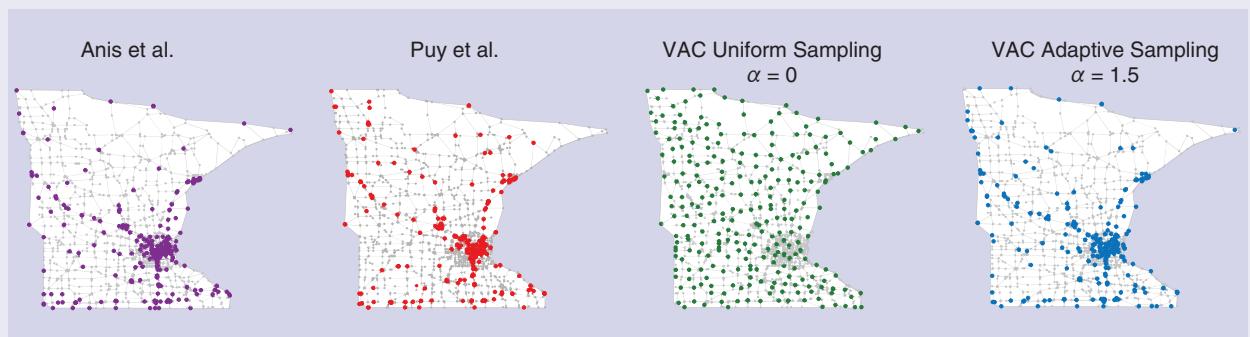


FIGURE S1. Different sampling strategies on the Minnesota graph with a density of 10%.

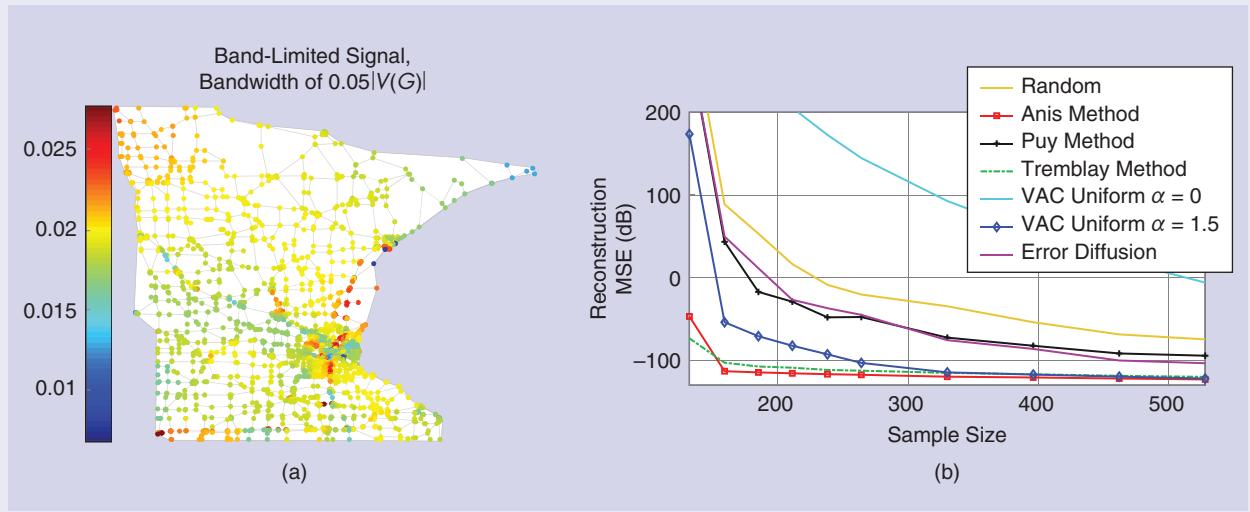


FIGURE S2. Example of (a) a band-limited signal and (b) reconstruction errors for different sampling methods.

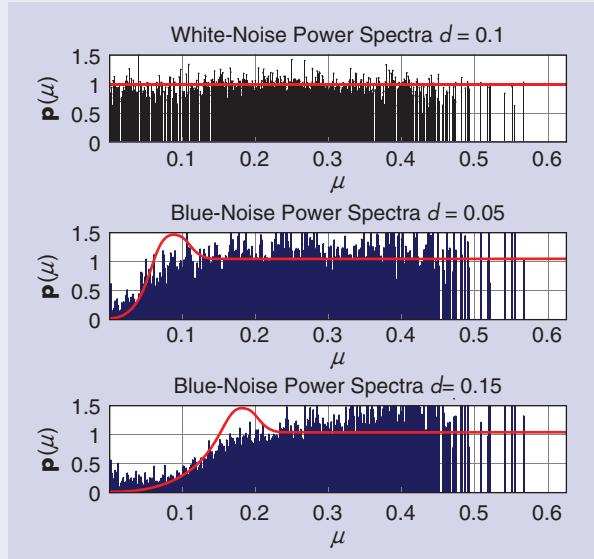
Like halftones, white-noise graph signals have a flat Fourier response for all frequencies, while blue noise shows a substantial reduction in red energy near  $\mu = 0$ . It is this reduction in redness that makes blue noise beneficial to graph signal sampling. Specifically, Parada-Mayorga et al. [6] proved

that the redness of any sampling pattern, with support in a set  $S \subset V(G)$ , is related to the value of  $\Lambda_{S^c}$  according to

$$\Lambda_{S^c} > C_\delta \left( \frac{R_s}{\text{vol}(G)R_s - m(1 - \frac{m}{N})^2} \right)^{\frac{2}{\delta}}, \quad (5)$$

## Example 2

In Figure S3, the red lines represent the ideal spectral shape defined by Ulichney [4] for digital halftones



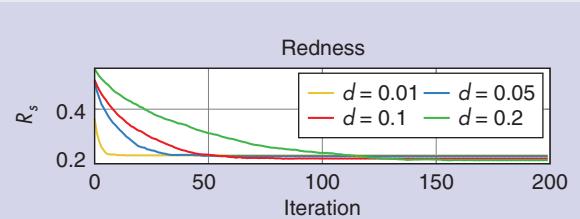
**FIGURE S3.** Graph power spectra on a sensor network generated by random and void-and-cluster sampling patterns with horizontal axes  $\mu_k$  representing the normalized eigenvalues (frequency), while  $p(\mu_k)$  are the intensities of the power spectrum calculated as the squared average of the  $k$ th coefficient of the graph Fourier transform.

where  $\delta$  is the isoperimetric dimension of the graph, and  $C_\delta$  is a constant that depends only on  $\delta$ . Note that low values of  $R_s$  are desired because, as  $R_s$  decreases, the right-hand side of (5) increases, providing the guarantee that the minimum possible value of  $\Lambda_{S^c}$  increases as well. The proof of (5) relies on the use of the Dirichlet inequality  $\mu_D(S) > C_\delta(1/\text{vol}(S)^{2/\delta})$ , where  $\mu_D(S)$  is the first eigenvalue of the combinatorial Laplacian on the induced subgraph by  $S \subset V(G)$ , and on the fact that  $\text{vol}(S) \geq m^2(1 - m/N)^2 / (\sum_{\ell=2}^N \hat{s}(\ell)^2 / \mu_\ell)$  [6]. Additionally, we note that (5) is consistent with an intuitive interpretation of the relationship between  $R_s$  and the correlation between sampling nodes. We recall that as pointed out in [8], the correlation length of a signal  $x$  on a  $K$ -regular graph is given by  $\ell_x = K / \|\hat{x}\|_2^2 \sum_{\ell=2}^N (\hat{x}(\ell)^2 / \mu_\ell)$  and, therefore, the redness of a sampling pattern can be seen as directly linked to how correlated the sampling nodes in a sampling pattern are. And this correlation is intuitively expected to decrease when the distance between sampling nodes is increased, which at the same time would lead to a reduction of the redness.

Now, as ideal blue-noise sampling patterns inherit the low values of  $R_s$ , it is our goal to establish a spatial-domain statistic on which we can develop vertex-domain optimization algorithms sans eigenvectors. Translating Ulichney [4] from halftoning to graphs, blue-noise graph signals can be

transcribed to the graph Fourier transform. The graph power spectra of random or white-noise sampling is characterized by a flat shape across all of the frequencies, while blue-noise patterns  $d = 0.05$  and  $0.15$  are characterized by the suppression of low-frequency components that becomes more pronounced for patterns with higher density.

Note that the location of the peaks in the ideal frequency response (Figure S3) and the pair correlation (Figure 2) move in opposite directions, but a closed-form expression can be derived for only specific families of graphs. Patterns with higher densities require more iterations to arrive at their stable form, while patterns with lower densities are stabilized after just a few iterations (see Figure S4).



**FIGURE S4.** An example of how the redness ( $R_s$ ) of the patterns, computed using (4), decreases as the number of iterations in the void-and-cluster algorithm increases.

described as a set of fixed radius disks covering the graph such that the area of the disk is equal to the area of the graph divided by the number of sampling nodes [6]. The radius of these disks is then the average distance (geodesic) between the nearest neighboring sample nodes indicated by  $\lambda_b$  and referred to as the *principal wavelength* of blue noise [14]. Although the relationship between  $\lambda_b$  and the gray level of a binary halftone is deterministic and proportional to the square of the density, it is not so on graphs, which may vary greatly in the way nodes are connected. Hence, the principal wavelength of an ideal blue-noise sampling pattern can be related to the number of sampling nodes according to  $d = 1/\mathbb{E}\{N(\lambda_b)\}$ , where  $\mathbb{E}\{N(\lambda_b)\}$  is the expected number of 1s on an open ball of radius  $\lambda_b$ , and  $d = \|s\|_0/N$  is the density of the sampling pattern. Because  $\mathbb{E}\{N(\lambda_b)\}$  is graph dependent, so is  $\lambda_b$ . The value of  $\lambda_b$  can be computed experimentally considering the histograms of the number of nodes inside a given radius for a given node.

It is important to point out that, for a particular graph, the values of  $\mathbb{E}\{N(\lambda_b)\}$  may vary dramatically from one region to the other if the local properties of the graph are not homogeneous. This leads to a dilemma, as to have a consistent relationship between  $d$  and  $\lambda_b$ , the spacing between sampling nodes should be modified accordingly. For the remainder of this

section, we consider graphs with this local homogeneous structure measured by the local isoperimetric dimension. In these types of graphs, equal spacing leads to an optimal reduction of the redness which, as it is shown in [6], is a desired attribute of good sampling patterns. Note that when this property is not satisfied, a partition of  $V(G)$  can be built such that each element in the partition has the same local properties.

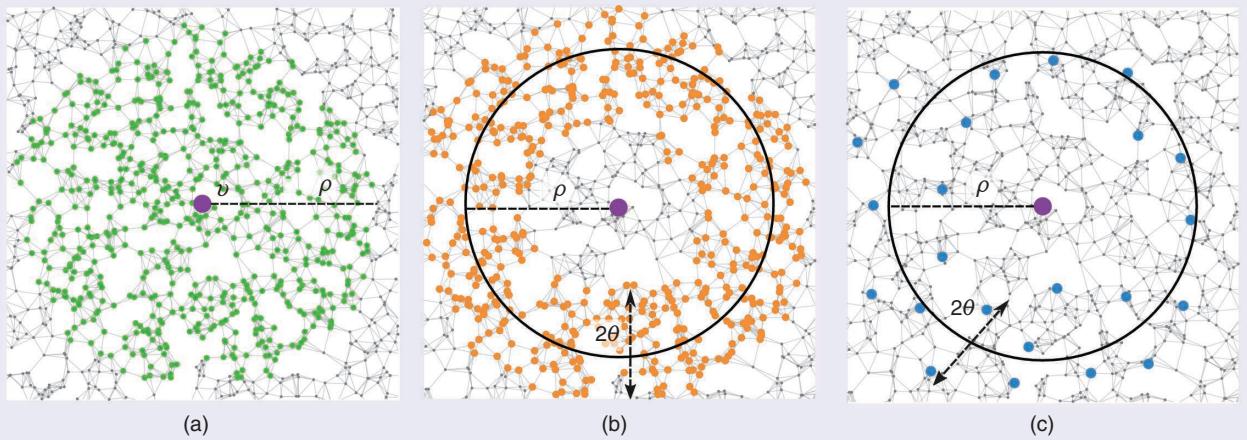
Assuming a local homogeneous graph structure, we expect the average distance between sampling nodes,  $\lambda_b$ , to have some variation with too large a variation having a noisy or uncorrelated (white) point distribution, while too small a variation has a periodic or rigid sampling arrangement. Statistically, we can quantify the variation in  $\lambda_b$  using the pair correlation to measure the increased/decreased likelihood of a sampled node occurring at a given distance from another sampled node. In [6], the graph pair correlation is introduced to analyze the vertex-domain characteristics of sampling patterns following the same principles used by Lau et al. [20] in traditional halftoning to provide a clear description of the distance between the closest points in a sampling pattern. Its definition relies on the use of concentric ring sets (annulus) given by  $B_\theta(v, \rho) = \{u \in V(G) : \rho - \theta \leq |\gamma_{v,u}| \leq \rho + \theta\}$ , where  $\gamma_{v,u}$  is the shortest path between the nodes  $v$  and  $u$ . Note that  $B_\theta(v, \rho)$  can be obtained as the set difference of two open balls centered at  $v$  with a radius of  $\rho - \theta$  and  $\rho + \theta$ , respectively. With this notion, the sample pair correlation of a sampling  $s$  is given by

$$\mathcal{R}_s(\rho) = \frac{\frac{1}{m} \sum_{i=1}^m \|s(B_\theta(s_i, \rho))\|_0}{\frac{1}{N} \sum_{v \in V(G)} \|s(B_\theta(v, \rho))\|_0}, \quad (6)$$

where  $m = \|s\|_0$  is the size of the support of  $s$ , i.e.,  $S = \text{supp}(s) = \{s_1, s_2, \dots, s_m\}$ , and  $\|s(B_\theta(s_i, \rho))\|_0$  is the number of elements in the support of  $s$  on  $B_\theta(s_i, \rho)$ .

To help illustrate the meaning of the pair correlation, Figure 1 shows in green all of the nodes within a shortest path distance of an arbitrary  $\rho$  from a sample node of interest for a random sensor network. A close inspection of the figure shows that  $\rho$  is measured based on the weights of the connections and not the Euclidean distance, as the disk of green has a rough edge. Shown in orange are all of the nodes that are within  $\pm\theta$  of  $\rho$  from the node of interest; while the sample nodes in blue are within the set of orange nodes. So the pair correlation is the ratio of the number of blue sample nodes over the total number of orange nodes. Thus, the numerator is the average number of elements in  $S$  enclosed on a ring centered in another element  $s_i \in S$ , while the denominator is the average number of elements in  $S$  enclosed on a ring centered in any node  $v \in V(G)$ . The value of  $\theta$  is selected according to the distribution of nonzero values of  $\mathbf{W}$  being the average of these values of the typical selection [6]. When a random pattern is considered via  $q$  of its realizations  $s_1, \dots, s_q$ , the pair correlation is calculated as  $\mathcal{R}(\rho) = 1/q \sum_{r=1}^q \mathcal{R}_{s_r}(\rho)$ , where the values of  $\rho$  for which  $\mathcal{R}(\rho)$  reaches a peak are an indication of a frequent occurrence of a given distance between sampling nodes, while the valleys indicate a suppression at these inter-point distances.

Presented in Figure 2(a) and (b) are the pair correlations for a white- and blue-noise process, respectively, where, for white noise or random sampling, the expected number of selected nodes stays constant for all values of  $\rho$ , with the pair correlation exhibiting a flat shape. Given the nature of blue-noise sampling to minimize clustering and equally space sampling nodes, we expect the pair correlation of a well-formed blue-noise sampling pattern to match that depicted in Figure 2(b), which exhibits the following properties: 1) given a sampling node, no other sampling nodes lie within a path of length  $\rho < \lambda_b$ ; 2) for  $\rho > \lambda_b$ , the expected number of sampling nodes per unit area (on the geodesic domain) tends to a



**FIGURE 1.** How to calculate the pair correlation for a given set of sampling nodes where (a) the green nodes are all nodes within a radius  $\rho$  of a given sample nodes, (b) the orange nodes are within  $\pm\theta$  of  $\rho$  from the node of interest, and (c) the blue nodes are sample nodes within the set of orange nodes.

constant value; and 3) the internode distance with the highest occurrence is  $\lambda_b$ .

As stated previously, as the variance in the distance between the nearest sampling nodes increases, we expect to see the valley in  $\mathcal{R}_s(\rho)$  for  $\rho < \lambda_b$  to fill in, while the peak at  $\rho = \lambda_b$  is expected to dissipate until it becomes flat for all  $\rho$ , which is characteristic of uncorrelated or white-noise sampling patterns. From Figure 2(b), we see the characteristic peak when  $\rho$  is equal to  $\lambda_b$  with a noticeable suppression of samples nodes at a distance less than  $\lambda_b$  apart. In [11], Tremblay referred to this suppression as a *repulsion property*. We summarize this result by saying that ideal blue-noise sampling patterns on graphs have their sampling nodes spread as far apart as possible from each other.

### Void and cluster

To demonstrate that good blue-noise sampling patterns can be constructed entirely in the vertex domain, Parada-Mayorga et al. [6] proposed a blue-noise generator for graphs inspired by Ulichney's void-and-cluster (VAC) algorithm [21]. For halftoning, VAC modifies the concentration of printed pixels moving dots from regions with high concentration to regions with low concentration. For graphs, VAC relies on the use of geodesic distances between sampling nodes on the graph, maximizing for each sampling node the sum of distances with respect to other sampling nodes. Specifically, VAC uses the Gaussian kernel,  $\mathbf{K}(u, v) = \exp(-\Gamma(u, v)^2/\sigma)$ , to map the geodesic distance  $\Gamma(u, v)$  between nodes  $u$  and  $v$  into a new set of values that are directly used to measure the density of sampling nodes in a given region. The values of  $\mathbf{K}(u, v)$  are close to 1 and 0 for the small and large values of  $\Gamma(u, v)$ , respectively.

Initially, we set the signal  $s$  to a random binary signal where the signal's support,  $\text{supp}(s)$ , becomes the set of sampling nodes where  $s$  is equal to 1. The density of the sampling nodes,  $\text{supp}(s)$ , is calculated as  $\mathbf{c}(\text{supp}(s)) = \sum \mathbf{K}(\text{supp}(s), \text{supp}(s))$ , while the density of the remaining nonsampling nodes,  $\text{supp}(s)^c$ , is calculated as  $\mathbf{c}(\text{supp}(s)^c) = \sum \mathbf{K}(\text{supp}(s), \text{supp}(s)^c) - \tau$ , where  $\tau \geq N$  and  $\sum \mathbf{K}(A, B) = \sum_{a_i \in A, b_j \in B} \mathbf{K}(a_i, b_j)$  with  $a_i \in A$ ,  $b_j \in B$ . The new location of the nodes in  $s$  is done according to  $s(\text{argmax}_i \{\mathbf{c}(i)\}) = 0$  and  $s(\text{argmin}_i \{\mathbf{c}(i)\}) = 1$ . That is,  $s(\text{argmax}_i \{\mathbf{c}(i)\})$  is the sampling node with the highest corresponding local density. We make this node a non-sampling node by setting its signal value to 0. Likewise,  $s(\text{argmin}_i \{\mathbf{c}(i)\})$  is the nonsampling node having the least or sparsest local density, which we make into a sampling node by setting its signal value to 1. Repeating these steps until the same two nodes flip back and forth leads to a sampling pattern like the one depicted in Figure S1 (in green) in "Example 1," which shows a uniform and homogeneous spreading of the samples.

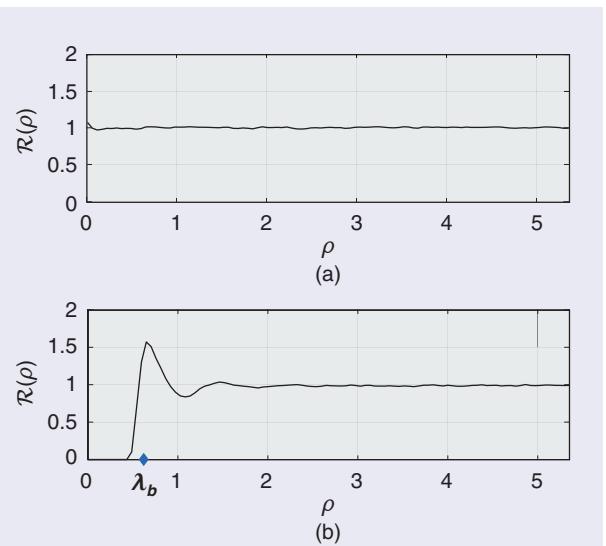
As proposed, the uniformity of the sampling pattern obtained by means of VAC does not consider the density of the connections around each node. As a consequence, the richness of the information associated to a given sampling node might be poor. This density can be included considering the mapping  $\mathbf{W}(u, v) \mapsto \mathbf{W}(u, v) \max \{\rho(u), \rho(v)\}^\alpha$ , where

$\rho(u) = N(u)/\mathbf{D}(u, u)$ ,  $N(u)$  is the number of neighbors of  $u$ , and  $\alpha \in \mathbb{R}^+$ . In this way, the weights will be increased in regions of the graph where we have nodes with a large number of connections, allowing the nodes to be further away than they would be with no modification of the weights. At the same time, those regions with a low density of connections will have low concentrations of sampling nodes, as the weights will be decreased, resulting in patterns like Figure S1 (in blue) in "Example 1."

### Error diffusion

Although VAC produces good blue-noise patterns, its computational complexity,  $\mathcal{O}(N^2)$ , makes it especially expensive for large graphs. So, we propose a method of error diffusion [7] that performs well with only  $\mathcal{O}(N(2 + \bar{d}_{\text{deg}}))$  complexity, where  $\bar{d}_{\text{deg}}$  is the average degree of the nodes in the graph. In digital halftoning, error diffusion has been extensively studied and improved for generating blue-noise-like patterns. The basic premise is to process signal samples serially along a predefined raster path, quantizing each sample to its nearest binary value and propagating the quantization error into the local neighborhood of yet-to-be-processed samples. How exactly the quantization error is divided into neighboring samples is determined by a user-defined error filter kernel, with different kernels producing different halftone textures—some more blue than others.

For the sampling of graph signals, we follow the same principle where graph edges take over the role of the error filter kernel to define how much the quantization error from a given node gets propagated to neighboring nodes. And just as with images, error diffusion on the graph requires a predefined raster path, with some paths leading to better blue-noise sample sets than others. We note that the idea of using diffusion for the generation of sampling patterns is connected to the notion of admissible partitions proposed in [13]. Then, although error

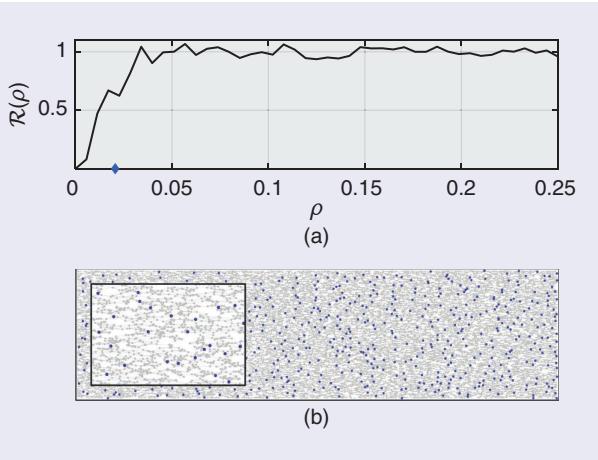


**FIGURE 2.** Plots of the pair correlation for (a) white-noise and (b) blue-noise graph-sampling sets.

diffusion was developed originally for halftoning applications, it can be considered on graphs as an attempt to generate the optimal sequence of admissible partitions that would lead to an optimal sampling set.

In this article, error diffusion on the graph starts by defining a constant signal  $\mathbf{x} = (m/N)\mathbf{1}_{N \times 1}$  or  $\mathbf{x}$  with amplitude  $d = m/N$  along with a raster path defining the order on which we will visit each node of the graph and a quantization threshold  $t_{\text{th}}$  defining when a scalar sample is closer to 0 or 1. In graphs, the raster path is defined using the labeling of the nodes assigned when the graph is generated. This labeling is not unique [1], but for our discussion, we consider graphs generated by the GSP toolbox with the labels it provides for the construction of the raster paths. Then, starting with the first node in the raster, an error is generated as  $e_p = s(1) - \mathbf{x}(1)$ , where  $s(1) = 1$  if  $\mathbf{x}(1) > t_{\text{th}}$  or  $s(1) = 0$  if  $\mathbf{x}(1) < t_{\text{th}}$ . The value of  $e_p$  is then diffused to the neighbors of this first node,  $\mathcal{N}(1)$ , as  $e(i) = \mathbf{W}(1, i)e_p / (\sum_{v \in \mathcal{N}(1)} W(v, i))$ ,  $i \in \mathcal{N}(1)$ . This process is repeated iteratively following the ordering given by the raster in such a way that the error at node  $v$  would be computed as  $e_p = s(v) - u$  with  $s(v) = 1$  if  $u > t_{\text{th}}$  or  $s(v) = 0$  if  $u < t_{\text{th}}$  and  $u = \mathbf{x}(v) - e(v)$ , diffusing this error as  $e(i) = e(i) + (\mathbf{W}(v, i)e_p / \sum_{i \in \mathcal{N}(v)} W(v, i))$ . In this way, the error is propagated according to the local connections of each node, ignoring connections to the nodes already processed. Note that this diffusion of error is completely equivalent to the one performed in traditional halftoning [4] in the sense that the amount of propagated error is not amplified or diluted in any way because the sum of the diffusion weights is exactly equal to 1; however, each node has its own error filter kernel that is, in part, determined by the raster order, as error can only be propagated to as-not-yet-processed nodes. There is the occasional situation where a node is the last to be processed within a local neighborhood with its corresponding error simply discarded, a situation analogous to error being diffused beyond the left and right edges of an image.

Displayed in Figure 3(b) is a cropped region in a 50,000-node random sensor network that has been sampled by means



**FIGURE 3.** (a) The pair correlation and (b) the snapshot of a sampling pattern obtained by means of error diffusion on a random sensor network with 50,000 nodes.

of error diffusion with a sampling density of 2%. As a means of statistically characterizing the spatial distribution of nodes, Figure 3(a) shows the corresponding pair correlation with an inhibition of clustering with  $R_s(\rho) < 1$  for  $\rho < 0.04$ . The lack of a clear peak at the principal wavelength is indicative of excessive varying in the average spacing between sample nodes relative, for instance, to the spacing from the pair correlation in Figure 2(b). This varying in spacing is visible in Figure 3(b) by the snake-like vertical strands/paths formed by blue dots; however, Figure 3(b) is clearly more organized than random sampling as  $R_s(\rho)$  approaches 0 for smaller values of  $\rho$ .

Now as a means of quantitatively evaluating error diffusion on graphs for signal sampling, “Example 1” shows the performance of error diffusion in comparison with other sampling approaches described in the “Void and Cluster” section. We can observe a clear improvement with respect to uniform random sampling as well as VAC when  $\alpha = 0$ , providing an alternative for sampling when graphs with a local isoperimetric dimension that is not homogeneous. Although not as good as VAC for  $\alpha = 1.5$ , error diffusion performs sampling with substantially lower computational complexity. To see this, we consider complexity in terms of two separated components. The first is the presampling set search (SS) component, which is associated with the calculations that are involved before the SS, while the second part is the cost of the SS itself, i.e., finding the sampling set. For error diffusion, we have none of the preliminary calculations with pre-SS=0 and SS=O(N(2 +  $\bar{d}_{\text{deg}}$ )). For VAC, we have a pre-SS=O(N(|E(G)| + N)log N) and SS=O((N - 1)(m + 2)). For Anis et al. [15], we have pre-SS=O(q |E(G)| kT<sub>1</sub>) and SS=O(Nk), where  $q$  is the power of the Laplacian considered in the application of [15],  $m$  is the number of sampling nodes,  $k$  is the bandwidth of the signals considered,  $\bar{w}$  represents the average of the nonzero elements in  $\mathbf{W}$ ,  $\bar{d}_{\text{deg}}$  is the average number of neighbors of each node,  $\alpha$  is a constant associated to the difference between the desired number of sampling nodes and its effective value, and  $T_1$  is the average number of iterations required for the convergence of a single eigendecomposition pair.

### Blue-noise sampling on multigraphs

The principles of blue-noise sampling on graphs can be generalized to richer graphical structures like multigraphs. The interaction between Facebook and Twitter, for instance, is a system that can be represented as a multigraph. Compared to simple graphs, there are few approaches to the problem of sampling on multigraphs [22], [23] given that, despite the fact that the spectral decomposition of the matrix operators on the subgraphs exists, it is not clear how to get a general spectral decomposition that involves all of the subgraphs. A naive way to generalize the sampling methods proposed for simple graphs, such as the ones proposed by Anis et al. [15], Puy et al. [12], Tremblay [11], and the adaptive VAC introduced in the “Void and Cluster” section, is to construct an equivalent simple graph by aggregating the subgraphs information. One of the simplest approaches is to define an equivalent weight matrix as the sum

of the weight matrix  $\mathbf{W}_l$  of each set of edges,  $E_l(\mathcal{G})$ . Thus, the resulting Laplacian consisting of the sum of the simple graphs' Laplacians is

$$\mathbf{L}(\mathcal{G}) = \sum_{l=1}^p \mathbf{L}(G_l). \quad (7)$$

Other approaches suggested for the spectral representation of multigraphs consider the summation of the individual spectral kernels or the linked matrix factorization that approximates the graph through a low-rank matrix approximation using the common factors among graphs and the characteristics of each graph  $G_l$  [24]. The problem with sampling based on the equivalent weight matrix is that the combination or superposition of independent stochastic sampling patterns, each designed for a different set of edges  $E(\mathcal{G})$ , does not necessarily produce aggregate patterns with ideal characteristics. The VAC algorithm may open the door to a new generation of vertex-domain algorithms whose aim is to produce ideal patterns per set of edges  $E(\mathcal{G})$ , yet produce combined sampling patterns with ideal aggregate properties. At the very least, it will demonstrate the flexibility of the vertex-domain interpretation of blue noise.

For multigraph sampling, we generalize the VAC algorithm from the "Void and Cluster" section by measuring the sample density in each graph  $G_l$  using Gaussian kernels  $\mathbf{K}_l(u, v) = \exp(-\Gamma_l(u, v)^2/\sigma)$  with a final density, denoted as  $C$ , computed as the sum of graph densities,  $\mathbf{c}_l$ , according to  $C = \sum_{l=1}^p \mathbf{c}_l$ . As before, each density vector  $\mathbf{c}_l$  is computed for the sampling nodes as  $\mathbf{c}_l(\text{supp}(s)) = \sum \mathbf{K}_l(\text{supp}(s), \text{supp}(s))$  and for the remaining nodes  $\mathbf{c}_l(\text{supp}(s)^c) = \sum \mathbf{K}_l(\text{supp}(s), \text{supp}(s)^c) - \tau$ . The new location of the nodes is selected depending on the multigraph density measure  $C$  as  $s(\text{argmax}_i\{C(i)\}) = 0$  and  $s(\text{argmin}_i\{C(i)\}) = 1$ , moving 1s from high-density regions [large values of  $C(i)$ ] to regions of low density [low values of  $C(i)$ ]. This will force the selected vertices to be spread apart as far apart as possible in the multigraph as well as the individual graphs while maintaining an intersample spacing approximately equal to the blue-noise wavelength; alternatively, a weighted sum could increase a particular layer's influence. Regardless, it is clear that when  $p = 1$ , the algorithm reduces to the original VAC, which makes this algorithm a generalized version of VAC for  $p \geq 1$ .

To illustrate the performance of sampling on multigraphs, consider a multimodal sensing application using hyperspectral images (HSIs) merged with 3D depth maps. HSIs provide spectral signatures of materials across a wide range of wavelengths, but they do not provide structural and elevation information. On the other hand, 3D depth cameras provide the size, structure, and elevation of different objects. Their fusion provides unmatched capabilities in an array of applications [25], [27]. To this end, a multispectral point cloud was captured in five spectral bands centered at 370-nm ultraviolet (UV), 470-nm (blue), 530-nm (green), 700-nm (red), and 783-nm near-infrared response (NIR) using a calibrated set of Intel Real-Sense D415 and Nurgo UV cameras. NIR-red-green-blue-UV+ depth focal plane array measurements were

captured over 15 viewing angles separated by 10° [26]. A 19,812-multispectral point cloud was then obtained by fusing the multispectral measurements. The 3D point cloud consists of a set of points  $\mathcal{Q} = \{q_j \mid j = 1, \dots, N\}$ , where each point has an associated 3D coordinate vector  $\mathbf{t}_j = [x, y, z]$  representing its spatial location. The spectral information in  $q_j$  is stored in the vector  $\mathbf{b}_j = [b_j^{\text{UV}}, b_j^{\text{B}}, b_j^{\text{G}}, b_j^{\text{R}}, b_j^{\text{NIR}}]$ .

The multigraph  $\mathcal{G}$ , representing a 3D point cloud, assigns each voxel in the point cloud to a vertex  $v \in V(\mathcal{G})$ . Two sets of edges,  $E_1(\mathcal{G})$  and  $E_2(\mathcal{G})$ , are assigned based on the geometry of the point cloud and its relative luminance  $Y_j = 0.2126b_j^{\text{R}} + 0.7152b_j^{\text{G}} + 0.0722b_j^{\text{B}}$ . In the case of  $E_1(\mathcal{G})$ , two nodes,  $v_i$  and  $v_j$ , are connected if their 3D spatial distance  $\mathcal{D}_{ij} = \|\mathbf{t}_i - \mathbf{t}_j\|_2$  is among their  $K$ -nearest neighbors, assigning  $\mathbf{W}_1(v_i, v_j) = \mathcal{D}_{ij}$ . For the set of edges  $E_2(\mathcal{G})$ , node  $v_i$  is connected to node  $v_j$  if it is one of the  $K$ -closest and most similar neighbors according to the relative luminances  $\mathcal{D}_{Yij} = |Y_i - Y_j|$ , assigning  $\mathbf{W}_2(v_i, v_j) = \mathcal{D}_{Yij}$ . Let  $\{\mathbf{b}^c \in \mathcal{R}^N : c \in \{\text{UV, B, G, R, NIR}\}\}$  be a set of real signals on the graph, defined by the mapping  $\mathbf{b}^c : V(\mathcal{G}) \rightarrow \mathbb{R}$ , where  $\mathbf{b}^c(v)$  is the pixel value associated to  $v \in V(\mathcal{G})$  in the  $c$  spectral channel. We explore different sampling methods using this multigraph in "Example 3."

## Conclusions

In this article, we reviewed the theory of blue-noise sampling as an extension of spatial dithering to graphs in both the vertex and Fourier domains. Previous works in spectral graph sampling exist that can be considered blue noise, but this article demonstrated that popular vertex-domain halftoning algorithms can be similarly effective with substantially lower computational complexity. We specifically looked at VAC as well as error diffusion. Although VAC outperformed error diffusion, error diffusion's  $\mathcal{O}(N(2 + \bar{d}_{\text{deg}}))$  complexity makes it especially appealing for large graphs. Future research in these spatial dithering techniques will look at how to optimally sample on large graphs, especially in light of the lack of alternate approaches. We hypothesize that VAC will be especially advantageous at sampling large graphs by independently processing partitions with a smoothing of sampling nodes along the boundaries. Additionally, we consider that there is great potential in the mathematical tools presented in [13], where the properties of the ordered partitions of nodes are studied in connection with uniqueness sets for band-limited signals.

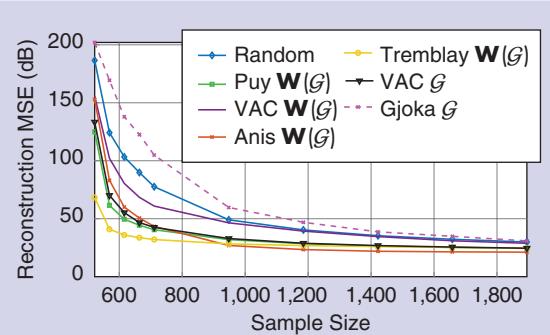
In terms of its graphical structures, graph blue-noise sampling suits, in a natural way, signal processing approaches where the knowledge of local properties is essential. As shown with multigraphs, blue noise applied for each induced subgraph obtained from the different families of edges leads to a simple strategy that does not ignore the intrinsic structure of the multigraph and does not require the spectral decomposition of an equivalent graph. Now, in some applications where the key role is to find and process a compressed representation of the information, one equivalent graph representation might not preserve or represent with fidelity the original properties of the signal on the multigraph. Beyond

### Example 3

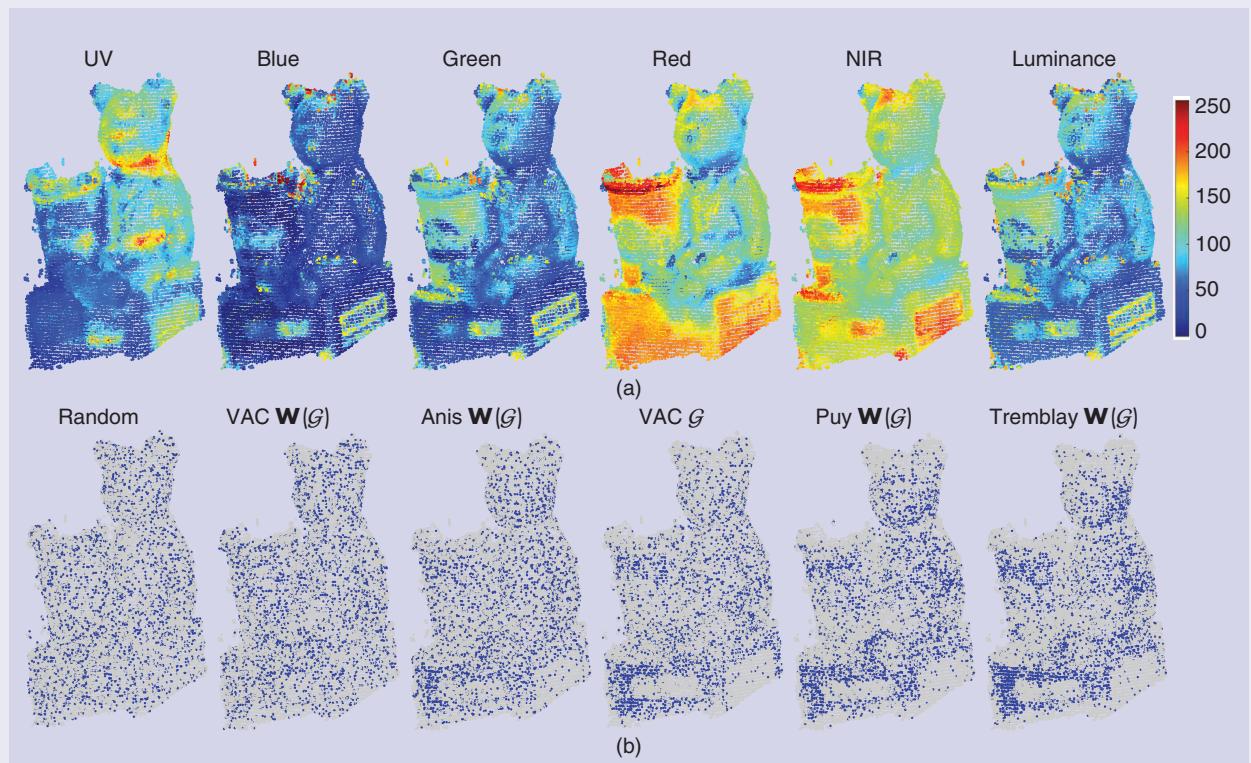
To compare the effectiveness of sampling by means of void and cluster (VAC) on multigraphs, we perform a numerical experiment using a multigraph,  $\mathcal{G}$ , with close to 20,000 nodes representing a 3D multispectral point cloud [see Figure S5(a)]. The sampling patterns for  $m = 0.2N$  using the different methods are depicted in Figure S5(b). Here, sampling methods proposed for simple graphs, such as the ones proposed by Anis et al. [15], Puy et al. [12], Tremblay [11], and the adaptive VAC introduced on multigraphs “Void and Cluster” section, use the equivalent weight matrix computed as  $\mathbf{W}(\mathcal{G}) = \gamma \mathbf{W}_1 + (1 - \gamma) \mathbf{W}_2$ , where the individual weight matrices,  $\mathbf{W}_1$  and  $\mathbf{W}_2$ , were first normalized and the parameter  $\gamma$  determines the influence of one set of edges over the other. The generalized VAC on  $\mathcal{G}$  considers instead  $(1 - \gamma) \mathbf{W}_1$  and  $\gamma \mathbf{W}_2$ , because, the smaller the weights the higher the influence of the edges when determining the sampling pattern, thus being the opposite effect when considering the addition of the weight matrices to compute  $\mathbf{W}(\mathcal{G})$ , as is the case with the former approaches. Adaptive VAC with  $\alpha = 3$  and  $\gamma = 0.6$  were used for this experiment. Note that the methods proposed by Anis et al. [15], Puy et al. [12], and Tremblay [11] require either the estimation of the spectral decomposition or the computation

of eigenvalues and eigenvectors, whereas VAC algorithms do not.

The mean square error (MSE) between the reconstructed and original signals was computed for all the spectral channels, and their average is displayed in Figure S6. Here, the proposed generalized VAC on the multigraph  $\mathcal{G}$  significantly outperforms the adaptive VAC on the equivalent weight matrix  $\mathbf{W}(\mathcal{G})$ , random sampling, and sampling on multigraphs  $\mathcal{G}$  proposed by Gjoka [22]. The



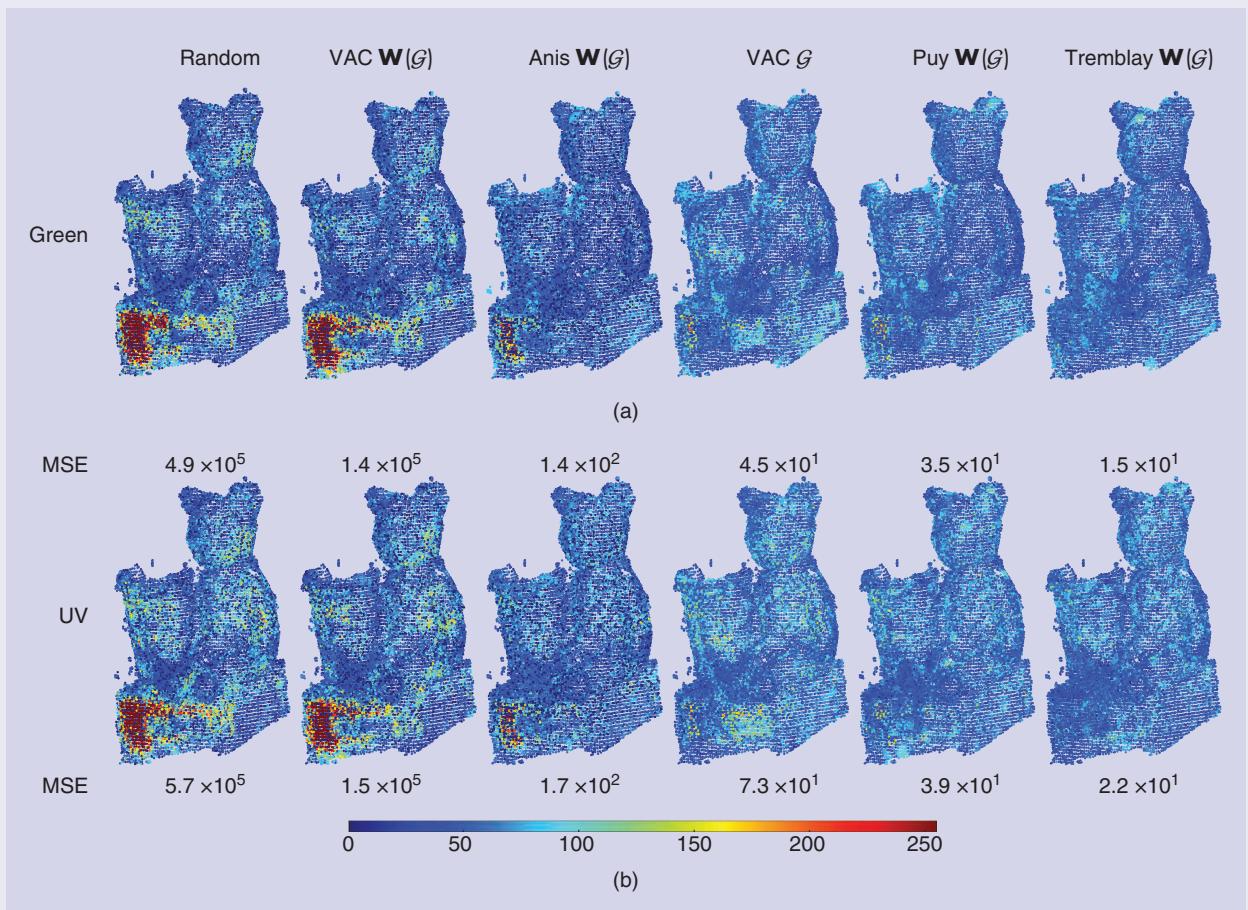
**FIGURE S6.** Reconstruction error for sampling patterns of different sizes using a down-sampled point cloud ( $N = 4,737$ ).



**FIGURE S5.** Approximately band-limited multispectral signals (approximate bandwidth of  $0.11N$ ) supported on a 3D point cloud (a) and sampling patterns with sample size  $m = 0.2N$  using different sampling methods (b). UV: ultraviolet; NIR: near-infrared response.

(Continued)

### Example 3 (Continued)



**FIGURE S7.** Absolute error between the original and reconstructed green (a) and ultraviolet (b) signals using different sampling methods. Gamma correction with  $\gamma_c = 1/3$  was applied only for display purposes.

MSE of the generalized VAC on  $\mathcal{G}$  closely approaches Puy et al.'s [12] MSE and, together with Tremblay [11], have a lower MSE compared to Anis et al.'s [15] for a small sample size. Tremblay [11] has the best performance for a small sample size.

However, because the methods proposed by Puy et al. [12] and Tremblay [11] are randomized, perfect reconstruction

is not possible. The authors then rely on a reweighting process on the reconstruction (3) to increase the chances of recovery.

Here, signals sampled by Puy et al. [12] and Tremblay [11] methods were reconstructed using (3) and all the others were reconstructed using (1). The equivalent Laplacian  $L(\mathcal{G})$  in [7] was used for all the cases (see Figure S7).

graphs and multigraphs, both multilayer and directed graphs are two important types of graphs where one can envision extending the concepts of VAC and error diffusion given the flexibility of the vertex-domain interpretation that ideal blue-noise sampling patterns on graphs have their sampling nodes spread as far apart as possible from each other.

#### Acknowledgment

This work was supported in part by the National Science Foundation under grants 1815992 and 1816003, and in part by the University of Delaware Research Foundation under the Strategic

Initiative Award and by the Institute Financial Services Analytics at the University of Delaware.

#### Authors

**Daniel L. Lau** (dllau@uky.edu) received his B.Sc. degree (with highest distinction) in electrical engineering from Purdue University, West Lafayette, Indiana, and his Ph.D. degree from the University of Delaware, Newark, in 1995 and 1999, respectively. Currently, he is a professor and the director of graduate studies for electrical and computer engineering at the University of Kentucky, Lexington. Among his many published works is

an article in *Proceedings of the IEEE* and his book *Modern Digital Halftoning*. His work has also been featured in trade magazines such as *Vision Systems Design*, *Photonics Spectra*, *Imaging Insight*, *Prosilica Camera News*, and *Inspect Magazine*. His research interests include image and signal processing, 3D imaging, and machine vision. He is a Senior Member of IEEE.

**Gonzalo R. Arce** (arce@udel.edu) is the Charles Black Evans Professor of Electrical and Computer Engineering and the JP Morgan-Chase Faculty Fellow at the University of Delaware, Newark. He held the 2010 and 2017 Fulbright-Nokia Distinguished Chair in Information and Communications Technologies at Aalto University, Finland. He has served as an associate and a guest editor of several IEEE journals, the Optical Society of America, and the Society of Photo-Optical Instrumentation Engineers (SPIE). He holds 20 U.S. patents and is the author of four books in imaging and signal processing. He received the NSF Research Initiation Award. His research interests include computational imaging, machine learning, and data science. He is a Fellow of SPIE and of IEEE.

**Alejandro Parada-Mayorga** (alejopm@seas.upenn.edu) received his B.Sc. and M.Sc. degrees in electrical engineering from Universidad Industrial de Santander, Colombia, in 2009 and 2012, respectively, and his Ph.D. degree in electrical engineering from the University of Delaware, Newark, 2019. Currently, he is a postdoctoral researcher at the University of Pennsylvania, Philadelphia, under the supervision of Prof. Alejandro Ribeiro. His research interests include geometric deep learning, algebraic neural networks, algebraic signal processing, graph neural networks, graph signal processing, topological data analysis, and topological signal processing. He is a Member of IEEE.

**Daniela Dapena** (mddapena@udel.edu) received her bachelor's degree in electrical engineering from the University de Los Andes, Merida, Venezuela, in 2018. She joined the Financial Services Analytics program at the University of Delaware, Newark, in 2018, where she is currently working toward her Ph.D. degree. Her research interests include graph signal processing with applications to machine learning. She is a Student Member of IEEE.

**Karelia Pena-Pena** (kareliap@udel.edu) received her B.Sc. degree in electrical engineering from Universidad de Los Andes, Venezuela, in 2017, and her M.Sc. degree in electrical and computer engineering in 2020 from the University of Delaware, Newark, where she is currently working toward her Ph.D. degree in electrical and computer engineering with a concentration in signal processing, communications, and control. Her research interests include graph signal processing, digital image processing, computer vision, machine learning, and optimization. She is a Student Member of IEEE.

## References

- [1] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018. doi: 10.1109/JPROC.2018.2820126.
- [2] P. D. Lorenzo, S. Barbarossa, and P. Banelli, "Sampling and recovery of graph signals," in *Cooperative and Graph Signal Processing*, P. Djuric and C. Richard, Eds. New York: Elsevier, 2018, pp. 817–837.
- [3] A. Sakiyama, Y. Tanaka, T. Tanaka, and A. Ortega, "Eigendecomposition-free sampling set selection for graph signals," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2679–2692, 2019. doi: 10.1109/TSP.2019.2908129.
- [4] R. A. Ulichney, "Dithering with blue noise," *Proc. IEEE*, vol. 76, no. 1, pp. 56–79, 1988. doi: 10.1109/5.3288.
- [5] D. L. Lau and G. R. Arce, *Modern Digital Halftoning*, 2nd ed. Boca Raton, FL: CRC Press, 2008.
- [6] A. Parada-Mayorga, D. L. Lau, J. H. Giraldo, and G. R. Arce, "Blue-noise sampling on graphs," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 3, pp. 554–569, Sept. 2019. doi: 10.1109/TSIPN.2019.2922852.
- [7] A. Parada-Mayorga, "Blue noise and optimal sampling on graphs," Ph.D. thesis, Dept. Elect. Comput. Eng., Univ. Delaware, Newark, 2019.
- [8] P. F. Stadler, "Landscapes and their correlation functions," *J. Math. Chem.*, vol. 20, no. 1, pp. 1–45, Mar 1996. doi: 10.1007/BF01165154.
- [9] I. Z. Pesenson, "A sampling theorem on homogeneous manifolds," *Trans. Amer. Math. Soc.*, vol. 352, no. 9, pp. 4257–4269, Apr. 2000. doi: 10.1090/S0002-9947-00-02592-7.
- [10] V. de Silva and R. Ghrist, "Coordinate-free coverage in sensor networks with controlled boundaries via homology," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1205–1222, 2006. doi: 10.1177/0278364906072252.
- [11] N. Tremblay, S. Barthélémy, and P. O. Amblard, "Optimized algorithms to sample determinantal point processes," 2018, arXiv:1802.08471.
- [12] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Appl. Comput. Harmon. Anal.*, vol. 44, no. 2, pp. 446–475, 2018. doi: 10.1016/j.acha.2016.05.005.
- [13] H. Fuhr and I. Z. Pesenson, "Poincaré and plancherel polya inequalities in harmonic analysis on weighted combinatorial graphs," *SIAM J. Discrete Math.*, vol. 27, no. 4, pp. 2007–2028, 2013. doi: 10.1137/120873674.
- [14] D. L. Lau, R. Ulichney, and G. R. Arce, "Blue and green noise halftoning models," *IEEE Signal Process. Mag.*, vol. 20, no. 4, pp. 28–38, July 2003. doi: 10.1109/MS.2003.1215229.
- [15] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for band-limited graph signals using graph spectral proxies," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3775–3789, July 2016. doi: 10.1109/TSP.2016.2546233.
- [16] D. L. Lau, G. R. Arce, and N. C. Gallagher, "Digital color halftoning with generalized error diffusion and multichannel green-noise masks," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 923–935, Dec. 1998. doi: 10.1109/83.841537.
- [17] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, Dec 2015. doi: 10.1109/TSP.2015.2469645.
- [18] I. Z. Pesenson, "Sampling solutions of schrodinger equations on combinatorial graphs," in *Proc. Int. Conf. Sampling Theory and Applications (SampTA)*, May 2015, pp. 82–85. doi: 10.1109/SAMPTA.2015.7148855.
- [19] N. Tremblay, P. O. Amblard, and S. Barthélémy, "Graph sampling with determinantal processes," in *Proc. 2017 IEEE 25th European Signal Processing Conf. (EUSIPCO)*, pp. 1674–1678. doi: 10.23919/EUSIPCO.2017.8081494.
- [20] D. L. Lau, G. R. Arce, and N. C. Gallagher, "Green-noise digital halftoning," *Proc. IEEE*, vol. 86, no. 12, pp. 2424–2444, Dec. 1998. doi: 10.1109/5.735449.
- [21] R. A. Ulichney, "The void-and-cluster method for dither array generation," in *Proc. SPIE, Human Vision, Visual Processing, Digital Displays IV*, 1993, vol. 1913, pp. 332–343. doi: 10.1117/12.152707.
- [22] M. Gjoka, C. T. Butts, M. Kurant, and A. Markopoulou, "Multigraph sampling of online social networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 9, pp. 1893–1905, Oct. 2011. doi: 10.1109/JSAC.2011.111012.
- [23] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering with multi-layer graphs: A spectral perspective," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5820–5831, Nov. 2012. doi: 10.1109/TSP.2012.2212886.
- [24] W. Tang, Z. Lu, and I. S. Dhillon, "Clustering with multiple graphs," in *Proc. 2009 9th IEEE Int. Conf. Data Mining*, pp. 1016–1021. doi: 10.1109/ICDM.2009.125.
- [25] R. Luo, W. Liao, H. Zhang, L. Zhang, P. Scheunders, Y. Pi, and W. Philips, "Fusion of hyperspectral and lidar data for classification of cloud-shadow mixed remote sensed scene," *IEEE J. Sel. Topics Appl. Earth Observ Remote Sens.*, vol. 10, no. 8, pp. 3768–3781, 2017. doi: 10.1109/JSTARS.2017.2684085.
- [26] K. Pena-Pena, X. Ma, D. L. Lau, and G. R. Arce, "Per-pixel calibration using CALTag and dense 3D point cloud reconstruction," in *Proc. Applications Digital Image Processing XLII*, 2019, vol. 11137, p. 1,113,71C. doi: 10.1117/12.2529332.
- [27] H. Rueda-Chacon, J. F. Florez-Ospina, D. L. Lau, and G. R. Arce, "Snapshot compressive ToF+spectral imaging via optimized color-coded apertures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2346–2360, Oct. 1, 2020.