INTERPOLATION AND DENOISING OF GRAPH SIGNALS USING PLUG-AND-PLAY ADMM

Yoshinao Yazaki[1], Yuichi Tanaka[1,2], and Stanley H. Chan[3]

[1]Tokyo University of Agriculture and Technology, Tokyo, Japan
[2]PRESTO, Japan Science and Technology Agency, Saitama, Japan
[3]Purdue University, West Lafayette, IN, USA

ABSTRACT

Signals defined on a network or a graph are often prone to errors due to missing data and noise. In order to restore the graph signal, interpolation and denoising are two necessary steps along with other graph signal processing procedures. However, existing graph signal interpolation and denoising methods are largely decoupled due to the opposite objectives of the two tasks and the inherent high computational complexity. The goal of this paper is to integrate graph interpolation and denoising using the Plug-and-Play (PnP) ADMM, a recently developed technique in image processing. When using the subsampling process as the forward model and graph filter as the denoiser, we show that PnP ADMM is equivalent to interpolating a bandlimited signal. Preliminary results are demonstrated via experiments, where the proposed method shows significantly better performance over existing methods.

Index Terms— Graph signal processing, interpolation, denoising, Plug-and-Play ADMM, graph sampling theory

1. INTRODUCTION

Analyzing the topology of a network (or a graph) and processing signals that live on the network is the center of many problems in networking, brain-analysis, weather forecasting, etc [1–6]. However, an observed graph signal is often subsampled due to missing data, and is contaiminated with noise. For example, measurements in sensor networks often have missing values due to sensors currently not working [7–10]. EEG contains large noise because a brain activity is measured from the surface of the head [11]. In semi-supervised learning, the number of labeled data is much smaller than that of the unlabeled data, and we have to estimate labels on the unlabeled vertices correctly [12]. Therefore, the restoration algorithms for graph signals are highly needed.

A typical signal observation model $\mathbf{y} \in \mathbb{R}^N$ is given by

$$\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{D} \in \mathbb{R}^{N\times N}$ is the degradation matrix, $\mathbf{x} \in \mathbb{R}^N$ is the original (graph) signal, $\mathbf{n} \in \mathbb{R}^N$ is an additive white Gaussian noise with zero mean and variance σ^2. Given $\mathbf{y}$, the goal is to recover $\mathbf{x}$ by overcoming the degradation and the noise.

Interpolation methods for graph signals have been studied in the context of graph sampling theory [13, 14]. However, most of the known results are based on the perfect recovery of "bandlimited" graph signals. Methods for non-bandlimited signals have been studied little so far [10], and in many cases they need eigen-decomposition which are computationally expensive.

Denoising methods for graph signals are available. Diffusion using a heat kernel [15] is one example, but its simple structure makes the denoising performance limited. Methods such as bilateral/trilateral filters are known to have a strong denoising performance [16,17]. However, the naive implementation of these filters in the graph frequency domain requires eigen-decomposition of graph Laplacian, which could be expensive. While Chebyshev polynomial approximation [17] or Krylov subspace technique [15] allows us to bypass the eigen-decomposition, intensive parameter tuning is still required to obtain the best result.

In order to integrate interpolation and denoising into a single optimization framework, in this paper we leverage a newly developed tool in the image processing literature called the Plug-and-Play (PnP) ADMM. PnP ADMM is a variation of the standard ADMM image restoration algorithm [18–21]. It is inspired by a convex optimization solver ADMM [22–24] but does not need to set a regularization function explicitly; instead, a part of the ADMM algorithm is replaced with a sophisticated denoiser that significantly enhances the image restoration performance. However, PnP ADMM cannot be directly used for our graph problem, because an image denoiser has a very different characteristic than a graph denoiser. For example, "patch-based" denoisers such as BM3D [25] is NP-hard for a graph problem as it needs to search similar signal values that are lying on similar (i.e., isomorphic) subgraphs to the target subgraph to be denoised.

The key contribution of this paper is the application of PnP ADMM to graph signal restoration problems, which is also the first demonstration of PnP ADMM beyond imaging tasks. The proposed approach has the following advantages:

- Its restoration performance is state-of-the-art compared to the existing restoration methods for graph signals.
- It does not require eigen-decomposition in the algorithm.
- The algorithm has a strong relationship to graph sampling theory for bandlimited signals [14].

In the experimental results for interpolation and denoising, the proposed method significantly outperforms the existing methods.

2. GRAPH SIGNAL PROCESSING

2.1. Graph and Graph Signal

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of the vertex set $\mathcal{V}$ and the edge set $\mathcal{E}$. The number of vertices is $N = |\mathcal{V}|$. Here, $\mathbf{x} \in \mathbb{R}^N$ is a graph signal whose ith sample is placed on the ith vertex of the graph. The structure of the graph is represented by the adjacency matrix

YY and YT are supported in part by JST PRESTO (JPMJPR1656) and JSPS KAKENHI (JP16H04362). SC is supported in part by the US National Science Foundation under grants CCF-1763896 and CCF-1718007.

$\mathbf{A} \in \mathbb{R}^{N\times N}$, the diagonal degree matrix $\mathbf{\Delta} \in \mathbb{R}^{N\times N}$, and the graph Laplacian matrix $\mathbf{L} := \mathbf{\Delta} - \mathbf{A}$. The $(m,\ n)$ element of $\mathbf{A}$, i.e., $a_{mn} \geq 0$, represents the edge weight between mth and nth vertices. The mth diagonal element of $\mathbf{\Delta}$ is defined as $\delta_{mm} = \sum_n a_{mn}$.

Since $\mathbf{L}$ is a real symmetric matrix, it can be decomposed as $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, where $\mathbf{V} \in \mathbb{R}^{N\times N}$ is the eigenvector matrix in which eigenvectors are arranged in each column and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_0, \lambda_1, \dots, \lambda_{N-1})$ is the diagonal eigenvalue matrix. We assume the eigenvalues are ordered as follows: $\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{N-1}$. The graph Fourier transform (GFT) is defined as $\widehat{\mathbf{x}} = \mathbf{V}^\top\mathbf{x}$.

2.2. Graph Sampling Theory

Let us consider the subsampling of $\mathbf{x}$ that generates a sampled signal whose number of samples is M $(M < N)$. The sampling matrix $\mathbf{\Psi} \in \{0,1\}^{M\times N}$ is defined as

$$\Psi_{ij} = \begin{cases} 1 & j = \mathcal{M}_i, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where $\mathcal{M}_i$ is the index of the sampled signal; the indices of the sampled signal is represented as $\mathcal{M} = \{\mathcal{M}_0, \dots, \mathcal{M}_{M-1}\}$. Simply, subsampling of $\mathbf{x}$ is defined as $\mathbf{x}_\mathcal{M} = \mathbf{\Psi}\mathbf{x}$.

In graph sampling theory, we consider the problem to recover $\mathbf{x}$ perfectly from its subsampled version $\mathbf{x}_\mathcal{M}$. While there exist many methods for interpolation [13, 14], we describe one of the interpolation methods [14] for bandlimited graph signals. The bandlimited graph signal satisfies the following condition:

$$\widehat{x}_k = 0 \text{ for all } k \geq K, \tag{3}$$

where $\widehat{x}_k$ is the kth element of $\widehat{\mathbf{x}}$ and K is the bandwidth.

If the graph signal satisfies (3), it can be perfectly recovered from $\mathbf{x}_\mathcal{M}$ if $\mathbf{\Psi}$ satisfies [14, Theorem 1]

$$\mathrm{rank}(\mathbf{\Psi}\mathbf{V}_{(K)}) = K, \tag{4}$$

where $\mathbf{V}_{(K)} = [\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_{K-1}]$ is the first K columns of $\mathbf{V}$. As a result, the interpolation of the downsampled graph signal is represented as

$$\mathbf{x} = \mathbf{V}_{(K)}(\mathbf{\Psi}\mathbf{V}_{(K)})^{-1}\mathbf{x}_\mathcal{M}. \tag{5}$$

3. PLUG-AND-PLAY ADMM

3.1. ADMM

ADMM [22] is known as an algorithm for solving a certain convex optimization problem and widely used for image restoration [23, 24, 26]. ADMM solves the minimization problem with the non-differentiable convex function

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) + \lambda s(\mathbf{x}), \tag{6}$$

where $f(\mathbf{x})$ is strongly convex and $s(\mathbf{x})$ is convex. For solving (6) using ADMM, it is transformed to

$$\arg\min_{\mathbf{x},\mathbf{v}} f(\mathbf{x}) + \lambda s(\mathbf{v}) \quad \text{s.t. } \mathbf{x} = \mathbf{v}. \tag{7}$$

This minimization problem can be solved by the following iterative algorithm:

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2}||\mathbf{x} - \tilde{\mathbf{x}}^{(k)}||^2, \tag{8}$$

$$\mathbf{v}^{(k+1)} = \arg\min_{\mathbf{v}} \lambda s(\mathbf{v}) + \frac{\rho}{2}||\mathbf{v} - \tilde{\mathbf{v}}^{(k)}||^2, \tag{9}$$

$$\bar{\mathbf{u}}^{(k+1)} = \bar{\mathbf{u}}^{(k)} + (\mathbf{x}^{(k+1)} - \mathbf{v}^{(k+1)}), \tag{10}$$

where $\tilde{\mathbf{x}}^{(k)} = \mathbf{v}^{(k)} - \bar{\mathbf{u}}^{(k)}$, $\tilde{\mathbf{v}}^{(k)} = \mathbf{x}^{(k+1)} + \bar{\mathbf{u}}^{(k)}$ and $\rho > 0$. Since (8) relates to the image model of (1), it is called an inverse module. Since (9) relates to the regularization function $s(\mathbf{x})$, it is called a denoising module.

3.2. PnP ADMM

In the minimization problem (7), the objective function $f(\mathbf{x})$ and the regularization function $s(\mathbf{x})$ must be determined before running the algorithm. $f(\mathbf{x})$ is determined according to the image degradation model. On the other hand, $s(\mathbf{x})$ must be determined beforehand according to the subjective belief of the original signal.

PnP ADMM [18, 27] has been proposed as a method of automatically and implicitly determining $s(\mathbf{x})$ by replacing the denoisng module in (9) by some off-the-shelf image denoising algorithm, e.g., non-local means [28, 29], BM3D [25], etc. Let $\mathcal{R}(\cdot)$ be one of the off-the-shelf image denoising algorithms. Here, (9) is rewritten in the PnP algorithm as

$$\mathbf{v}^{(k+1)} = \mathcal{R}(\tilde{\mathbf{v}}^{(k)}). \tag{11}$$

By doing so, the image restoration performance is improved while the entire optimization problem generally becomes nonconvex.

In PnP ADMM, $s(\mathbf{x})$ is usually unknown due to the complex denoiser $\mathcal{R}(\cdot)$. However, in the case that $\mathcal{R}(\tilde{\mathbf{v}}) = \mathbf{W}\tilde{\mathbf{v}}$, where $\mathbf{W} \in \mathbb{R}^{N\times N}$ is a symmetric smoothing filter, $s(\mathbf{x})$ becomes [18, 27]

$$s(\mathbf{x}) = \frac{\rho}{2\lambda}\mathbf{x}^\top(\mathbf{W}^{-1} - \mathbf{I})\mathbf{x}, \tag{12}$$

where $\mathbf{I}$ is the identity matrix. By substituting (12) and $f(\mathbf{x}) = \frac{1}{2}||\mathbf{D}\mathbf{x} - \mathbf{y}||^2$ for (6), the restored signal can be written as the following closed-form solution:

$$\mathbf{x}' = \mathbf{U}\mathbf{\Sigma}(\mathbf{\Sigma}\mathbf{U}^\top\mathbf{D}^\top\mathbf{D}\mathbf{U}\mathbf{\Sigma} + \rho\mathbf{\Sigma}(\mathbf{I} - \mathbf{\Sigma}))^{-1}\mathbf{\Sigma}\mathbf{U}^\top\mathbf{D}^\top\mathbf{y}, \tag{13}$$

where $\mathbf{U}$ is the eigenvector matrix of $\mathbf{W}$ and $\mathbf{\Sigma}$ is the eigenvalue matrix of $\mathbf{W}$, i.e., $\mathbf{W} := \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$.

4. PNP ADMM-BASED GRAPH SIGNAL RESTORATION

4.1. Applying PnP ADMM to Graph Signals

It is relatively easy to use the PnP iteration (8), (11), (10) for graph signal restoration. The inverse module (8) can be applied straightforwardly for graph signals because the degradation model can be described in (1) even for graph signal processing.

In the denoising module, the image denoiser $\mathcal{R}(\cdot)$ is replaced by a denoising method using graphs and graph spectra. However, such method should be light-weighted, meaning that we should avoid

- eigen-decomposition to calculate the GFT basis, and
- parameter tuning which could be costly.

The first problem can be mitigated by using Chebyshev polynomial approximation, whereas the second problem can be solved with a "one-shot" denoising algorithm. In this paper, we use the diffusion on graphs [15] and the graph trilateral filter (GTF) with the fixed parameter [17] as the light-weight denoisers for graph signals. We also use Chebyshev polynomial approximation for GTF and Krylov subspace technique for heat kernel to bypass the eigen-decomposition, as presented by the original papers.

The iteration of PnP ADMM of the proposed method is therefore represented as follows:

$$\mathbf{x}^{(k+1)} = (2\mathbf{D}^\top\mathbf{D} + \rho\mathbf{I})^{-1}(2\mathbf{D}\mathbf{y} + \rho\tilde{\mathbf{x}}^{(k)}), \tag{14}$$

$$\mathbf{v}^{(k+1)} = \mathcal{R}_g(\tilde{\mathbf{v}}^{(k)}), \tag{15}$$

$$\bar{\mathbf{u}}^{(k+1)} = \bar{\mathbf{u}}^{(k)} + (\mathbf{x}^{(k+1)} - \mathbf{v}^{(k+1)}), \tag{16}$$

where $\mathcal{R}_g$ is the light-weight graph signal denoiser. (14) is derived from solving (8) with $f(\mathbf{x}) = ||\mathbf{D}\mathbf{x} - \mathbf{y}||^2$. If $\mathcal{R}_g(\cdot)$ does not require eigen-decomposition and the degradation matrix $\mathbf{D}$ is a diagonal matrix, e.g., interpolation or denoising, the entire algorithm does not need the matrix decomposition or inversion.

While the GTF [17] with Chebyshev polynomial approximation is also an eigen-decomposition-free algorithm, it requires hundreds of the GTFs to determine the optimal parameter that minimizes the estimated MSE. Additionally, in its optimization process, the GTF filtering matrix (not a filtered signal vector) of size $N \times N$ should be calculated and stored: It leads to the huge storage costs. In contrast, the above PnP ADMM algorithm just needs the filtered vector in (14)–(16) that significantly reduces the computation cost.

In the proposed method, the parameter ρ in (14) should be tuned depending on the denoising algorithm used and the (estimated) noise level. In this paper, we select ρ with $\rho = c_1\sigma^2 + c_2$, where c_1 and c_2 are chosen experimentally. In our experiments in the following section, such ρ is robust enough for all situations.

4.2. Relationship with Graph Sampling Theory

Here, we reveal the relationship between the interpolation method of the graph sampling theory (5) and the graph signal interpolation with PnP ADMM. In this subsection, we assume that the degradation matrix $\mathbf{D}$ is a binary diagonal matrix $\mathbf{D} \in \{0, 1\}^{N\times N}$.

Theorem 1. *Assume that the graph signal is noise-free and the sampling matrix satisfies* (4)*. Additionally, let $\mathcal{R}_g(\cdot)$ be the denoising module* (11) *with the ideal graph low-pass filter represented as follows:*

$$\mathcal{R}_g(\tilde{\mathbf{v}}) = \mathbf{W}'\tilde{\mathbf{v}} := \mathbf{V}h_{\text{ideal}}(\mathbf{\Lambda})\mathbf{V}^\top\tilde{\mathbf{v}}, \tag{17}$$

where $h_{\text{ideal}}(\mathbf{\Lambda}) = \operatorname{diag}(\underbrace{1, \dots, 1}_{K}, \underbrace{0, \dots, 0}_{N-K})$*. In this case, the graph signal interpolation using PnP ADMM, i.e.,* (14)–(16)*, is identical to the interpolation of the bandlimited graph signal in* (5)*.*

Proof. Since $\mathbf{W}'$ is a symmetric matrix, the interpolated signal using PnP ADMM can be computed by a closed-form as in (13). It is represented as

$$\mathbf{x}' = \mathbf{V}\mathbf{\Lambda}'(\mathbf{\Lambda}'\mathbf{V}^\top\mathbf{D}^\top\mathbf{D}\mathbf{V}\mathbf{\Lambda}' + \rho\mathbf{\Lambda}'(\mathbf{I} - \mathbf{\Lambda}'))^{-1}\mathbf{\Lambda}'\mathbf{V}^\top\mathbf{D}^\top\mathbf{y}, \tag{18}$$

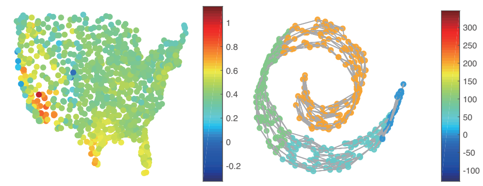
where $\mathbf{\Lambda}' := h_{\text{ideal}}(\mathbf{\Lambda})$.

Because $\mathbf{\Lambda}'$ is the diagonal matrix whose elements are binary values, it is clear that $\mathbf{\Lambda}'(\mathbf{I} - \mathbf{\Lambda}') = \mathbf{O}$. Additionally, $\mathbf{V}\mathbf{\Lambda}' = \begin{bmatrix}\mathbf{V}_{(K)} & \mathbf{O}\end{bmatrix}$. Therefore, (18) can be rewritten as

$$\mathbf{x}' = \mathbf{V}_{(K)}(\mathbf{V}_{(K)}^\top\mathbf{D}^\top\mathbf{D}\mathbf{V}_{(K)})^{-1}\mathbf{V}_{(K)}^\top\mathbf{D}^\top\mathbf{y}. \tag{19}$$

Here, $\mathbf{D}$ is a zero-padding matrix that fills missing values with 0. The sampling matrix $\mathbf{\Psi}$ is also a matrix that extracts samples at the sampling point specified by $\mathcal{M}$. $\mathbf{D}$ and $\mathbf{\Psi}$ have the relationship $\mathbf{D} = \mathbf{\Psi}^\top\mathbf{\Psi}$. $\mathbf{D}^\top\mathbf{D} = \mathbf{D}$ is also satisfied. With the above relationships, we can further rewrite (19) as

$$\mathbf{x}' = \mathbf{V}_{(K)}((\mathbf{\Psi}\mathbf{V}_{(K)})^\top(\mathbf{\Psi}\mathbf{V}_{(K)}))^{-1}(\mathbf{\Psi}\mathbf{V}_{(K)})^\top\mathbf{\Psi}\mathbf{y}. \tag{20}$$



(a) (b)

Fig. 1. Original graph signals. (a) Real weather data in the US used for the interpolation experiment. (b) Graph signal on Swiss roll graph used for the denoising experiment.

Here, the sampling matrix satisfies (4): $(\mathbf{\Psi}\mathbf{V}_{(K)})^{-1}$ can be computed. (20) then becomes

$$\begin{aligned}\mathbf{x}' &= \mathbf{V}_{(K)}(\mathbf{\Psi}\mathbf{V}_{(K)})^{-1}((\mathbf{\Psi}\mathbf{V}_{(K)})^\top)^{-1}(\mathbf{\Psi}\mathbf{V}_{(K)})^\top\mathbf{\Psi}\mathbf{y} \\ &= \mathbf{V}_{(K)}(\mathbf{\Psi}\mathbf{V}_{(K)})^{-1}\mathbf{\Psi}\mathbf{y}.\end{aligned} \tag{21}$$

Since $\mathbf{x}_{\mathcal{M}} = \mathbf{\Psi}\mathbf{y}$, the interpolated signal finally becomes

$$\mathbf{x}' = \mathbf{V}_{(K)}(\mathbf{\Psi}\mathbf{V}_{(K)})^{-1}\mathbf{x}_{\mathcal{M}}. \tag{22}$$

This is identical to (5). □

As a result, PnP ADMM with the ideal smoothing filter (17) is the same as the interpolation based on the graph sampling theory, regardless of ρ. This means the PnP-based method can perfectly recover bandlimited graph signals.

While the original recovering method needs matrix inversion, our PnP-based interpolation can avoid it as mentioned previously. It leads to that, for larger graphs, the PnP-based interpolation is faster than the original one. We numerically verify this in the following section.

5. EXPERIMENTAL RESULTS

In this section, we compare the restoration performance of graph siganls for interpolation and denoising. The computation time is also compared. All experiments were implemented with MATLAB R2015b, and they were run on Intel Xeon E5-2690 2.9 GHz CPU with 64 GB RAM.

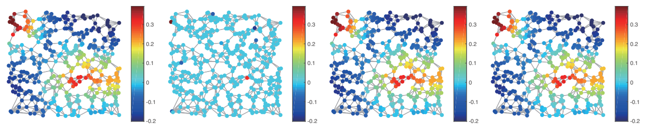
In this paper, we use the GTF or heat kernel as the denoising module in (11). They are abbreviated as PnP-GTF and PnP-Heat. In the original GTF, a regularization parameter (of the GTF; not ρ) is optimized in order to minimize the MSE between the restored and original signals. The GTFs with/without parameter optimization are denoted as GTF_{opt} and $\text{GTF}_{\text{unopt}}$, respectively. In PnP-GTF, we used $\text{GTF}_{\text{unopt}}$ with the fixed parameter 1 as the denoising module, since its optimization is time-consuming.

GTF depends not only on the graph but also on the signal values on the graph. However, in our algorithm, it is inefficient to change the filter coefficients of the GTF in each iteration according to $\tilde{\mathbf{v}}^{(k)}$. Therefore, we reuse its filter coefficients in each iteration; they are determined from the pre-restored signal.

For the heat kernel $h(\lambda) = e^{-\lambda t}$, a time parameter t is fixed to $t = 0.1$ in all experiments.

5.1. Interpolation

In the experiment for interpolation, the proposed method is compared with `gsp_interpolate` [30] in GSPBOX [31] (abbrevi-



(a) (b) (c) (d)

Fig. 2. Interpolation result of graph sampling theory and proposed method. (a) Original signal. (b) Sampled signal. (c) Interpolated signal by GSamp (MSE: 3.49×10^{-29}). (d) Interpolated signal by the proposed method using (17) (MSE: 4.39×10^{-29}).

Table 1. Interpolation results (average MSEs ($\times 10^{-3}$) for 100 runs)

Missing rate [%]	10	20	30	40	50
GInt	9.54	9.06	8.69	8.41	8.14
GSamp	4.74	4.94	5.25	5.74	6.45
GInt + GTF$_{\text{opt}}$	**3.57**	**3.92**	4.35	4.80	5.38
PnP-Heat	4.17	4.37	4.62	4.90	**5.28**
PnP-GTF	3.74	4.00	**4.33**	**4.74**	5.32

ated as GInt) and the reconstruction method (5) of the graph sampling theory [14] (abbreviated as GSamp). Because the bandwidth K for (5) is needed (but it is generally not known a priori), we set $K = \lfloor N/10 \rfloor$. The interpolation result of GInt is used to calculate edge weighs for PnP-GTF. For a comparison purpose, the two-step algorithm is also applied to the data. It first estimates the missing values with GInt, and then performs denoising to the interpolated signal with GTF$_{\text{opt}}$ (abbreviated as GInt + GTF$_{\text{opt}}$).

In this experiment, we use Quality Controlled Local Climatological Data, which is real weather data recorded by NOAA [32]. We select average temperature data in contiguous United States on Aug. 1, 2017. The number of stations is $N = 1052$. The dynamic range of the data is normalized to $[0, 1]$. After then, random sampling is performed and the white Gaussian noise with $\sigma = 0.1$ is added. The graph is generated by k nearest neighbor with $k = 10$.

The original signal is shown in Fig. 1(a), and the interpolation results are shown in Table 1. It is clear that the proposed methods are significantly better than GInt and GSamp for all missing rates, and is comparable to or even outperforms GInt + GTF$_{\text{opt}}$. Note that the proposed method is a one-step integrated restoration and is significantly faster than GInt + GTF$_{\text{opt}}$ (see Section **??**).

Within the proposed methods, PnP-GTF shows the best results with the 10–40 % missing rates, and PnP-Heat shows the best results for the 50 % missing rate.

5.2. Comparison with Graph Sampling Theory

Here, a completely bandlimited graph signal with the bandwidth $K = 10$ is recovered. In this experiment, the signal on a random sensor graph with $N = 400$ is used, and the bandlimited graph signal is generated by $\mathbf{x} = \mathbf{V}\left[\tilde{\mathbf{f}}^\top \ \mathbf{0}^\top\right]^\top$, where $\tilde{\mathbf{f}} \in \mathbb{R}^K$ is a random signal conforming to the normal distribution with zero mean and variance 1. The sampled signal is chosen to satisfy (4), where the number of sampled signal is $M = 10$. As the denoiser $\mathcal{R}_g(\cdot)$, we use the ideal graph low-pass filter in (17) to demonstrate Theorem 1.

The recovered graph signals are shown in Fig. 2 along with their MSEs. It was also experimentally determined that the proposed method with the ideal filter can perfectly recover the bandlimited graph signal.

Table 2. Denoising Results (average MSEs for 100 runs)

σ	10	20	30	40	50
Heat kernel	67.6	154	298	499	757
GTF$_{\text{opt}}$	**62.6**	139	205	263	312
GTF$_{\text{unopt}}$	65.6	156	306	517	787
PnP-Heat	82.2	**138**	**194**	**244**	**287**
PnP-GTF	79.6	155	208	256	300
Noisy signal	98.4	394	885	1574	2459

Table 3. Computation time [sec.] (OoM: Out of Memory)

Interpolation					
N	10^2	10^3	10^4	10^5	10^6
GSamp	0.003	0.110	64.97	OoM	OoM
GInt	0.003	0.018	2.552	OoM	OoM
PnP-GTF	0.412	1.270	12.61	140.7	2694
PnP-Heat	2.648	4.436	37.30	360.9	3538
Denoising					
N	10^2	10^3	10^4	10^5	10^6
GTF$_{\text{opt}}$	33.48	1292	42735	OoM	OoM
GTF$_{\text{unopt}}$	0.028	0.237	2.452	29.05	1126
Heat kernel	0.003	0.005	0.037	0.335	3.429
PnP-GTF	0.412	1.347	12.48	152.9	3220
PnP-Heat	2.637	4.650	37.72	365.8	3508

5.3. Denoising

Next, the denoising performance is compared. In this experiment, the signal on Swiss roll graph with $N = 400$, as shown in Fig. 1(b), is used.

The denoising results are summarized in Table 2. According to the table, PnP-Heat shows the best results for $\sigma \in [20, 50]$. It is worth noting that PnP-GTF is better than the GTF$_{\text{opt}}$ for $\sigma = 40$ and 50, despite of the fixed regularization parameter.

5.4. Computation time

The average computation times after 10 runs are shown in Table 3. For interpolation, the proposed method is faster than GSamp for $N \geq 10^4$. The computation time for GInt is shorter than the other two, however, its reconstruction performance is worse than them. For denoising, the proposed methods are always faster than GTF$_{\text{opt}}$. While GTF$_{\text{unopt}}$ and the diffusion are fast, their performance is poor.

It is important to note that GSamp, GInt, and GTF$_{\text{opt}}$ cannot compute the restored signal for $N \geq 10^5$ due to the excessive usage of memory. In contrast, the proposed method was stably executed until (at least) $N = 10^6$.

6. CONCLUSION

In this paper, the graph signal restoration method using PnP ADMM is proposed. Though its basic algorithm is similar to that for image restoration, it has many desirable properties for graph signal processing. Furthermore, it has a strong relationship with the interpolation based on graph sampling theory. Since there have been proposed various sampling theories for graph signals like [13], studying further connection to the sampling theory is one of future works. In the experiment, it was shown that the performances of interpolation and denoising of the proposed method are significantly better than those of the existing approaches.

7. 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 Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[2] 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.

[3] Y. Tanaka, "Spectral domain sampling of graph signals," *IEEE Trans. Signal Process.*, vol. 66, no. 14, pp. 3752–3767, Jul. 2018.

[4] A. Sakiyama, K. Watanabe, Y. Tanaka, and A. Ortega, "Two-channel critically-sampled graph filter banks with spectral domain sampling," *IEEE Trans. Signal Process.*, vol. 67, no. 6, pp. 1447–1460, Mar. 2019.

[5] J. Mei and J. M. F. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 2077–2092, Apr. 2017.

[6] G. Cheung, E. Magli, Y. Tanaka, and M. Ng, "Graph spectral image processing," *Proc. IEEE*, vol. 106, no. 5, pp. 907–930, May 2018.

[7] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, pp. 235–284, 2008.

[8] A. Sakiyama, Y. Tanaka, T. Tanaka, and A. Ortega, "Efficient sensor position selection using graph signal sampling theory," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2016, pp. 6225–6229.

[9] ——, "Accelerated sensor position selection using graph localization operator," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2017, pp. 5890–5894.

[10] ——, "Eigendecomposition-free sampling set selection for graph signals," *arXiv preprint arXiv:1809.01827*, 2018.

[11] H. Higashi, T. M. Rutkowski, T. Tanaka, and Y. Tanaka, "Multilinear discriminant analysis with subspace constraints for single-trial classification of event-related potentials," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 7, pp. 1295–1305, Oct. 2016.

[12] A. Gadde, A. Anis, and A. Ortega, "Active semi-supervised learning using sampling theory for graph signals," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2014, pp. 492–501.

[13] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3775–3789, Jul. 2016.

[14] 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.

[15] F. Zhang and E. R. Hancock, "Graph spectral image smoothing using the heat kernel," *Pattern Recognition*, vol. 41, no. 11, pp. 3328–3342, Nov. 2008.

[16] A. Gadde, S. K. Narang, and A. Ortega, "Bilateral filter: Graph spectral interpretation and extensions," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 1222–1226.

[17] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka, "Graph signal denoising via trilateral filter on graph spectral domain," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 137–148, Jun. 2016.

[18] S. H. Chan, "Algorithm-induced prior for image restoration," *arXiv preprint arXiv:1602.00715*, 2016.

[19] Y. Dar, A. M. Bruckstein, M. Elad, and R. Giryes, "Postprocessing of compressed images via sequential denoising," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3044–3058, Jul. 2016.

[20] A. Rond, R. Giryes, and M. Elad, "Poisson inverse problems by the plug-and-play scheme," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 96–108, Nov. 2016.

[21] S. Ono, "Primal-dual plug-and-play image restoration," *IEEE Signal Process. Lett.*, vol. 24, no. 8, pp. 1108–1112, Aug. 2017.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine learning*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[23] J. Yang, Y. Zhang, and W. Yin, "An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise," *SIAM Journal on Scientific Computing*, vol. 31, no. 4, pp. 2842–2865, Jul. 2009.

[24] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, Sep. 2010.

[25] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[26] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An augmented Lagrangian method for total variation video restoration," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3097–3111, Nov. 2011.

[27] S. H. Chan, "Performance analysis of Plug-and-Play ADMM: A graph signal processing perspective," *arXiv preprint arXiv:1809.00020*, 2018.

[28] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," *SIAM Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, Jan. 2005.

[29] S. H. Chan, T. Zickler, and Y. M. Lu, "Monte carlo nonlocal means: Random sampling for large-scale image filtering," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3711–3725, Aug. 2014.

[30] I. Pesenson, "Variational splines and Paley–Wiener spaces on combinatorial graphs," *Constructive Approximation*, vol. 29, no. 1, pp. 1–21, Feb. 2009.

[31] N. Perraudin, J. Paratte, D. I. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *arXiv preprint arXiv:1408.5781*, 2014.

[32] National Centers for Environmental Information, "Quality controlled local climatological data," 2015.