Unrolled Denoising of Attributes on Graphs with Local-and-Global Smoothness Assumption

Reina Kaneko, Junya Hara, Hiroshi Higashi, and Yuichi Tanaka
The University of Osaka, Osaka, Japan
{r.kaneko@sip., j.hara@, higashi@, ytanaka@}comm.eng.osaka-u.ac.jp

***Abstract*—This paper presents a denoising method for attributes on graphs by using a *local-and-global smoothness* assumption. Restoration of attributes on graphs is one of the most fundamental studies in graph signal processing. Many work assumes graphs are noiseless (while signal values are noisy), however, both of the graphs and signal values may be noisy. In this paper, we address denoising of attributes by assuming *local-and-global smoothness* nature of graph signals: They are locally smooth on the graph and are globally smooth in an appropriate space regardless of the graphs. Since the global smoothness is not affected by the degradation of graphs, the proposed method is robust to it. We formulate a corresponding convex problem that can be iteratively solved with a monotone operator splitting algorithm. Moreover, we unroll the iterations to apply deep algorithm unrolling (DAU), and these internal parameters are automatically turned from the training data. In experiments for denoising of color point clouds, our method exhibits higher restoration performance in PSNR than existing model-based and DAU-based methods even when graphs are obtained in a noisy environment.**

***Index Terms*—Graph signal processing, attribute denoising, monotone operator splitting, deep algorithm unrolling**

I. Introduction

Irregularly-distributed measurements, i.e., attributes or features, are used in many applications like those in sensor networks, IoT, and point clouds, to name a few [1], [2]. Interconnectivity among samples can often be utilized during restoration processes. For example, attributes associated with sensor networks and point clouds exhibit spatial connectivity, which can typically be represented by a graph.

Attributes on graph nodes can be represented by a set of *graph signals*, where edges represent the relationships between the samples (i.e., the nodes). In contrast to standard signals on a regular grid such as images, graph signal processing (GSP) explicitly exploits the underlying structure of attributes [3]. GSP is highly effective for analyzing data that are irregularly distributed in space [4].

One major challenge in GSP is denoising of graph-structured attributes. For example in point clouds, individual points may be degraded by lighting distortions, environmental noise, and/or errors in camera calibration [5], [6]. Such degradations corrupt the attributes. Furthermore, during the measurement process, spatial coordinates themselves may also be noisy: It leads to noisy graphs. However, many GSP-based signal restoration methods only consider the case for noiseless graphs [2], [7].

This work is supported in part by JST AdCORP under Grant JPMJKB2307 and JSPS KAKENHI under Grant 23K26110.

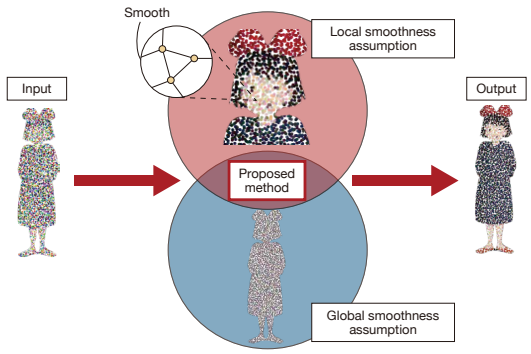


Fig. 1: LGS assumption considered in this paper. Our method aims to restore graph signals at the intersection of outputs from local and global smoothness assumptions.

In this paper, to address the above-mentioned challenges, we propose a denoising method for attributes on graphs by considering the following *local-and-global smoothness* (LGS) assumption:

Assumption 1 (Local smoothness on graphs): For graph-structured attributes, two signal values connected with a large edge weight are usually similar. Therefore, the *local smoothness assumption* is often utilized for graph-based restoration [2], [7]. However, the sole use of it will limit the restoration accuracy under a noisy graph case, which is often obtained from corrupted coordinates.

Assumption 2 (Global smoothness of attributes): In this paper, we also consider *global smoothness* as well as the local smoothness. Since the global smoothness assumes that the attributes are smooth in an appropriate space, this assumption is helpful for attribute denoising if graphs are noisy because the global smoothness assumption does not depend on the graphs.

The LGS assumption is illustrated in Fig. 1. While many GSP methods use Assumption 1, and Assumption 2 is also utilized in various image processing methods like [8], they have not been integrated in GSP so far.

In the proposed method, we first formulate a convex graph signal denoising problem based on the LGS assumption. The local smoothness assumption is represented by a regularization term taking into account total variation of attributes on a given

graph. The global smoothness assumption is translated to the regularization term of the Euclidean distance matrix among attributes.

In fact, the minimization problem is solved by a monotone operator splitting (MOS) algorithm [9], [10]. Moreover, we *unroll* its iterations and train the regularization parameters by deep algorithm unrolling (DAU) [11].

In experiments on color point cloud denoising, our method exhibits better restoration performance in PSNRs than existing model-based and DAU-based methods.

Notation: $[\mathbf{A}]_{n,m}$ denotes the (n,m) element of $\mathbf{A}$. A weighted undirected graph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which $\mathcal{V}$ and $\mathcal{E}$ are sets of nodes and edges, respectively. We use a weighted adjacency matrix $\mathbf{W}$ for representing the connection between nodes, where $[\mathbf{W}]_{m,n} \geq 0$ is the edge weight between the mth and nth nodes. The weighted graph incidence matrix is denoted by $\mathbf{B} \in \mathbb{R}^{|\mathcal{E}| \times N}$. We index the set of edges as $\mathcal{E} = \{e_s\}_{s=1}^{|\mathcal{E}|}$. Then, the (s,t) entry of $\mathbf{B}$ is

$$[\mathbf{B}]_{s,t} = \begin{cases} \sqrt{[\mathbf{W}]_{p,q}} & e_s = (v_p, v_q) \quad \text{and} \quad t = p, \\ -\sqrt{[\mathbf{W}]_{p,q}} & e_s = (v_p, v_q) \quad \text{and} \quad t = q, \\ 0 \quad \text{otherwise}, \end{cases} \tag{1}$$

where v_p is the pth node. We also use $\|\cdot\|_1$ for ℓ_1 norm and $\|\cdot\|_F$ for Frobenius norm.

II. Related Work

In this section, we review existing graph-based signal/attribute denoising techniques. Basically, they assume smoothness of attributes on a local region and perform (iterative) low-pass filtering on a graph. These methods can be classified into three categories: Model-based restoration, deep learning–based restoration, and integrated restoration.

A. Model-based Restoration

Model-based approaches typically formulate a (convex) optimization problem that includes a data fidelity term and regularization term(s) [12]. Signal priors are required in the formulation. The widely-used prior is local smoothness and it can be defined as follows:

Assumption 1 (Local smoothness assumption with total variation)**.** Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times Q}$ be the original attributes with the number of nodes N and the number of attributes/measurements Q. Signals satisfying the local smoothness assumption are defined as follows:

$$\mathcal{T}_{\text{LS}} = \{\mathbf{X} \in \mathbb{R}^{N \times Q} \mid \|\mathbf{B}\mathbf{X}\|_1 \leq \delta_l\}, \tag{2}$$

where $\delta_l > 0$.

Other definitions of smoothness like in [2] can also be considered. A lot of model-based methods stem from this assumption. While they are completely interpretable, they often require many iterations for convergence and careful tuning of regularization parameters.

B. Deep Learning-based Restoration

For irregularly-distributed attributes, graph neural networks are often applied [13], [14]. Graph convolutional networks (GCNs) are one of them, which are an extension of convolutional neural networks [13].

GCNs learn the neural network parameters from the data by minimizing a loss function. However, these methods often lack interpretability and require large training datasets. Note that, for the graph-structured data, we usually do not have enough measurements for training many parameters in neural networks, especially for the noisy graph cases.

C. Integrated Restoration

To address the above-mentioned challenges, intermediate solutions between the model- and deep learning-based approaches have attracted researchers. This approach typically uses "black-box" denoisers in its iterative algorithm based on convex optimization. Well-known approaches are plug-and-play (PnP) [15] and regularization-by-denoising [16]. They were first proposed in image processing, and are now utilized in graph-based signal analysis [7], [17].

Furthermore, they can be combined with a deep learning methodology as DAU [11]. Intuitively, it trains parameters in an iterative (convex) optimization algorithm by utilizing deep learning techniques like backpropagation. As a result, the trained regularization parameters are different in all iterations (corresponding to layers). This maintains the interpretability of the algorithm and practically decreases the number of iterations.

DAU for graph-structured signals have been studied in [7], [17]. A method in [7] is the unrolled PnP algorithms and presents the state-of-the-art restoration performance for graph signals. While the robustness of graph perturbations is studied in [7], it still assumes that graphs are noiseless.

In summary, existing methods highly depend on local smoothness on a given (noiseless) graph. If the graphs are noisy, they will limit the restoration performance. Therefore, one needs to adopt an alternative formulation/methodology that can be combined with the local smoothness assumption. In the following, we propose a denoising method for attributes when graphs may be corrupted by noise.

III. Denoising for Attributes with LGS Assumption

In this section, we present the proposed denoising method based on the LGS assumption. First, we introduce the signal observation model and assumptions. Second, minimization problems based on the two assumptions are integrated into one convex minimization problem. Third, we derive the iterative denoising algorithm. Finally, we unroll the iterations to make the algorithm trainable by using DAU.

A. Signal Observation Model and LGS Assumption

In general, the attributes measured in a noisy condition are given as follows:

$$\mathbf{Y} = \mathbf{X} + \mathbf{N}, \tag{3}$$

where $\mathbf{N}$ is a noise matrix whose entry is an i.i.d. random Gaussian noise.

Here, we mathematically formulate our LGS assumption and corresponding signal denoising problems.

Local Smoothness Assumption: If we only consider the local smoothness assumption on the graph, the signal denoising problem is formulated as follows:

$$\widetilde{\mathbf{X}}_1 = \arg\min_{\mathbf{X}_1} \|\mathbf{X}_1 - \mathbf{Y}\|_F^2 + \lambda_1 \|\mathbf{B}\mathbf{X}_1\|_1, \tag{4}$$

where $\lambda_1 > 0$ is the regularization parameter. The above equation is widely used and can be found anywhere [2], [7]. As previously mentioned, if $\mathcal{G}$ includes noise, $\mathbf{B}$ must be noisy. Therefore, the regularization term in (4) does not work properly due to the noisy $\mathbf{B}$.

Global Smoothness Assumption: First, the global smoothness assumption is defined as follows:

Assumption 2 (Global smoothness assumption with total variation)**.** Let $\mathbf{X} \in \mathbb{R}^{N\times Q}$ be the observed attributes. The global smoothness assumption is defined as

$$\mathcal{T}_{\text{GS}} = \{\mathbf{X} \in \mathbb{R}^{N\times Q} \mid \|\mathcal{D}(\mathbf{X})\|_1 \leq \delta_g\}, \tag{5}$$

where $\mathcal{D}(\mathbf{X}) \coloneqq (\mathbf{I} \circ \mathbf{X}\mathbf{X}^\top)\mathbf{1}\mathbf{1}^\top + \mathbf{1}\mathbf{1}^\top(\mathbf{I} \circ \mathbf{X}\mathbf{X}^\top) - 2\mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{N\times N}$ ($\mathbf{1}$ is the all one vector) is the Euclidean distance matrix ($[\mathcal{D}(\mathbf{X})]_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$) for the entries in $\mathbf{X}$ and $\delta_g > 0$.

With the sole use of Assumption 2, we can consider the following minimization problem:

$$\widetilde{\mathbf{X}}_2 = \arg\min_{\mathbf{X}_2} \|\mathbf{X}_2 - \mathbf{Y}\|_F^2 + \lambda_2 \|\mathcal{D}(\mathbf{X}_2)\|_1, \tag{6}$$

where $\lambda_2 > 0$ is the parameter. It is the same as (4) except for the regularization term. The problem (6) can perform denoising regardless of whether the graphs are noisy or not, since it does not utilize the graph.

B. Problem Formulation

We formulate a convex optimization problem taking into account (4) and (6) simultaneously.

First, we simply add (4) and (6) to utilize both local smoothness and global smoothness assumptions. As a result, we have the following optimization problem:

$$\begin{aligned} \min_{\mathbf{X}_1,\mathbf{X}_2}\ & 2\|\mathbf{X}_1 - \mathbf{Y}\|_F^2 + 2\|\mathbf{X}_2 - \mathbf{Y}\|_F^2 \\ & + \lambda_1 \|\mathbf{B}\mathbf{X}_1\|_1 + \lambda_2 \|\mathcal{D}(\mathbf{X}_2)\|_1 \\ & \text{s.t.} \quad \mathbf{X} = \mathbf{Y}, \quad \mathbf{X}_1 = \mathbf{X}, \quad \mathbf{X}_2 = \mathbf{X}. \end{aligned} \tag{7}$$

We then convert (7) into an unconstrained convex problem by introducing a fidelity term $\|\mathbf{X}_1 - \mathbf{X}_2\|_F^2$:

$$\begin{aligned} \min_{\mathbf{X}_1,\mathbf{X}_2}\ & 2\|\mathbf{X}_1 - \mathbf{Y}\|_F^2 + 2\|\mathbf{X}_2 - \mathbf{Y}\|_F^2 + 2\|\mathbf{X}_1 - \mathbf{X}_2\|_F^2 \\ & + \lambda_1 \|\mathbf{B}\mathbf{X}_1\|_1 + \lambda_2 \|\mathcal{D}(\mathbf{X}_2)\|_1 + \lambda_3 \|\mathbf{X}_1 - \mathbf{X}_2\|_1, \end{aligned} \tag{8}$$

where $\lambda_3 > 0$ is the parameter controlling the similarity between the output signals based on the local and global smoothness. The term $\|\mathbf{X}_1 - \mathbf{X}_2\|_1$ is inspired by the penalty term of the elastic net regularization [18].

Algorithm 1: Iterative algorithm to solve (9)

Input: $\mathbf{X}_1^{(0)}$, $\mathbf{X}_2^{(0)}$, $\mathbf{A_1}^{(0)}$, $\mathbf{A}_2^{(0)}$, $\mathbf{A}_3^{(0)}$
Output: $\mathbf{X}_1^{(n)}$, $\mathbf{X}_2^{(n)}$

while A stopping criterion is not satisfied **do**
- $\mathbf{X}_1^{(n+1)} = \mathbf{X}_1^{(n)} - \gamma_1(8\mathbf{X}_1 - 4\mathbf{X}_2 - 4\mathbf{Y} + \mathbf{B}(\mathbf{A}_1^{(n)}) + \mathbf{A}_3^{(n)})$
- $\mathbf{X}_2^{(n+1)} = \mathbf{X}_2^{(n)} - \gamma_2(4\mathbf{X}_1 - 4\mathbf{Y} + \mathcal{D}^*(\mathbf{A}_2^{(n)}) - \mathbf{A}_3^{(n)})$
- $\mathbf{A}_1^{(n+1)} = \text{prox}_{\lambda_1\|\cdot\|_1^*}(\mathbf{A}_1^{(n)} + \lambda_1(\mathbf{B}(2\mathbf{X}_1^{(n+1)} - \mathbf{X}_1^{(n)})))$
- $\mathbf{A}_2^{(n+1)} = \text{prox}_{\lambda_2\|\cdot\|_1^*}(\mathbf{A}_2^{(n)} + \lambda_2(\mathcal{D}(2\mathbf{X}_2^{(n+1)} - \mathbf{X}_2^{(n)})))$
- $\mathbf{A}_3^{(n+1)} = \text{prox}_{\lambda_3\|\cdot\|_1^*}(\mathbf{A}_3^{(n)} + \lambda_3(2\mathbf{X}_1^{(n+1)} - \mathbf{X}_1^{(n)} - (2\mathbf{X}_2^{(n+1)} - \mathbf{X}_2^{(n)}))$
- $n \leftarrow n + 1$

end while

C. Convex Algorithm

We derive the iterative algorithm to solve (8) using a monotone operator splitting (MOS) [9], [10]. We rewrite (8) by introducing auxiliary variables $\mathbf{A}_k$ ($k = 1, 2, 3$) as follows:

$$\begin{aligned} \min_{\mathbf{X}_1,\mathbf{X}_2}\ & 2\|\mathbf{X}_1 - \mathbf{Y}\|_F^2 + 2\|\mathbf{X}_2 - \mathbf{Y}\|_F^2 + 2\|\mathbf{X}_1 - \mathbf{X}_2\|_F^2 \\ & + \lambda_1 \|\mathbf{A}_1\|_1 + \lambda_2 \|\mathbf{A}_2\|_1 + \lambda_3 \|\mathbf{A}_3\|_1 \\ & \text{s.t.} \quad \mathbf{A}_1 = \mathbf{B}\mathbf{X}_1, \quad \mathbf{A}_2 = \mathcal{D}(\mathbf{X}_2), \quad \mathbf{A}_3 = \mathbf{X}_1 - \mathbf{X}_2. \end{aligned} \tag{9}$$

It is a MOS applicable form and its iterative solver is shown in Algorithm 1. The detailed derivation can be found in Appendix. In the algorithm, we use the following operators:

- ∇: Differential operator.
- $\mathcal{D}^*$: The adjoint operator of $\mathcal{D}$ and given by $\mathcal{D}^*(\mathbf{A}_2) = (\text{diag}(\mathbf{A}_2\mathbf{1}) + \text{diag}(\mathbf{1}^\top\mathbf{A}_2) - 2\mathbf{A}_2)\mathbf{X}_2$.
- $\text{prox}_{\lambda\|\cdot\|_1}$: Soft-thresholding operator [19].

Under appropriate conditions on $\{\gamma_1, \gamma_2, \lambda_1, \lambda_2, \lambda_3\}$, Algorithm 1 converges to the solution to (8).

D. Deep Algorithm Unrolling

While Algorithm 1 can be converged to the global minimum, we sometimes need many iterations for convergence. Here, we *unroll* the algorithm and train the regularization parameters in each iteration by using DAU.

The unrolled algorithm is visualized in Fig. 2. To solve (9), five hyperparameters are needed: $\{\gamma_1, \gamma_2, \lambda_1, \lambda_2, \lambda_3\}$. We replace these hyperparameters with trainable ones as $\{\gamma_1^{(l)}, \gamma_2^{(l)}, \lambda_1^{(l)}, \lambda_2^{(l)}, \lambda_3^{(l)}\}_{l=0}^{L-1}$, where L is the number of iterations (layers) of the algorithm.

Each unrolled layer corresponds to a single iteration in Algorithm 1 and parameters are trained with deep neural network techniques [11]. In the proposed method, the total number of trainable parameters is $5L$.

IV. Denoising Experiments

In this section, we perform experiments on denoising of color point clouds (CPCs) and compare our method with existing methods.

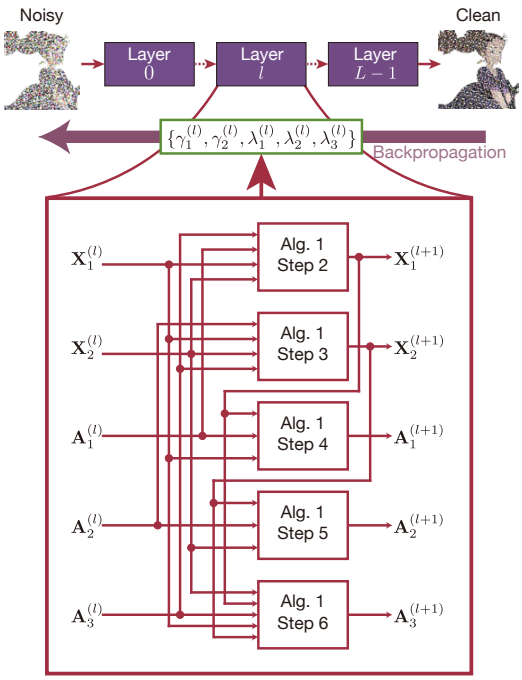


Fig. 2: Unrolled version of the proposed algorithm with DAU.

Setup: We use three CPCs, `Andrew`, `Sarah`, and `David`, taken from [20]. The original CPCs are (index-based) uniformly sampled to $N = 500$. For attributes and coordinates, we consider four pairs of noise levels: $(\sigma_{\text{RGB}}, \sigma_{\text{3-D}}) = (0.1, 0.1)$, $(\sigma_{\text{RGB}}, \sigma_{\text{3-D}}) = (0.1, 0.2)$, $(\sigma_{\text{RGB}}, \sigma_{\text{3-D}}) = (0.2, 0.2)$ and $(\sigma_{\text{RGB}}, \sigma_{\text{3-D}}) = (0.2, 0.3)$ where σ_{RGB} and $\sigma_{\text{3-D}}$ are the standard deviations of noise for RGB values and 3-D coordinates, respectively. In all methods, we construct a five-nearest neighbor graph from noisy coordinates.

Our method is compared with three methods: k-NN graph-based denoising method using geometry and color (abbreviated as k-NN) [1], fast graph-based CPC denoising (abbreviated as FastCPC) [21], and graph deep algorithm unrolling (abbreviated as GraphDAU) [7]. The proposed method has two specifications: 1) iterative convex optimization with fixed parameters, which is decided by grid-search (Section III-C) and 2) DAU with trained parameters (Section III-D). These two results are also reported.

The dataset contains 40 samples for every class, i.e., `Andrew`, `Sarah`, and `David`. We split these samples into 30 training data and 10 test ones. The training configurations are summarized in Table I. For training of the proposed DAU and GraphDAU [7], we use the mean squared error as a loss function.

We evaluate restoration performance based on peak signal-to-noise ratio (PSNR) of denoised color signals. We average the results obtained from 30 independent runs.

Results: Fig. 3 visualizes errors between the original and restoration results of `Sarah`. We observe that our method can suppress the effect of noise compared to the other methods.

Table II summarizes PSNRs of denoised attributes. As observed, the proposed method outperforms the other methods in most cases. For $(\sigma_{\text{RGB}}, \sigma_{\text{3-D}}) = (0.1, 0.1)$, the k-NN method presents the best PSNR. This is because it has hand-crafted parameters (provided by the original paper) that best fit to point clouds with small noises. As a result, at high noise levels, its PSNRs are quite similar to the noisy inputs. In contrast, our method exhibits stable PSNRs thanks to automatic parameter tuning via algorithm unrolling, whose objective function incorporates noisy attributes as well as noisy coordinates. Note that, in the testing phase, our DAU-based method has much smaller number of iterations than the model-based counterpart: The model-based approach of the proposed method typically runs about 500 iterations to converge, while the DAU counterpart only needs five iterations.

TABLE I: Training configuration.

Batch size	1
Epochs	7
Optimizer	Adam
Learning rate	0.001
Scheduler	StepLR

V. Conclusion

In this paper, we propose a denoising method for attributes on graphs by considering the following two assumptions simultaneously: 1) local smoothness assumption where two signal values connected with a large edge weight are usually similar, and 2) global smoothness assumption where attributes are smooth in the appropriate space. Our method is formulated as a convex optimization that can be solved by the MOS. We then unroll the iterative algorithm to learn the regularization parameters. In the denoising experiments of color point clouds, our method shows superior results to existing ones.

Appendix
Monotone Operator Splitting

MOS methods can be applied to solve problems of the following form [9], [10]:

$$\min_{\mathbf{X}\in\mathbb{R}^N} f_1(\mathbf{X}) + f_2(\mathbf{X}) + f_3\big(\mathcal{R}(\mathbf{X})\big), \tag{10}$$

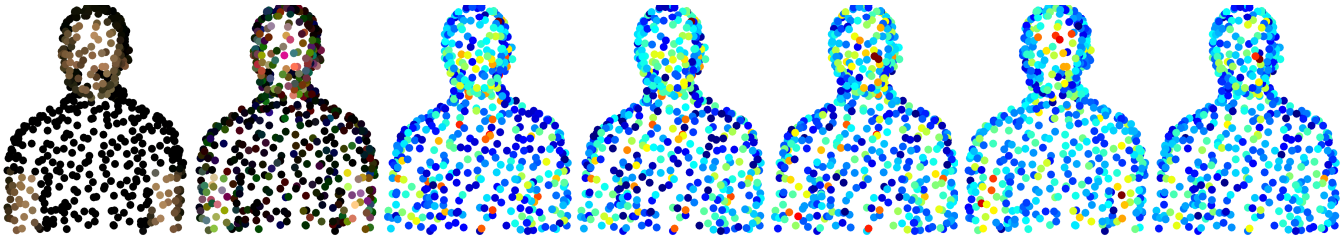
where f_1 is a differentiable convex function with β-Lipschitz continuous gradient ∇f_1, f_2 and f_3 are convex, lower-semicontinuous, and *proximable*, and $\mathcal{R}$ is a (possibly non-linear) monotone operator[1]. A function f is called proximable if its *proximal operator*, i.e., $\text{prox}_{\gamma f}(\mathbf{x}) = \arg\min_{\mathbf{y}} f(\mathbf{y}) + \frac{1}{2\gamma}\|\mathbf{x}-\mathbf{y}\|_2^2$, can be computed efficiently for some $\gamma > 0$.

A typical MOS algorithm alternates a forward (gradient) step on f_1 with backward (proximal) steps on f_2 and f_3^*. Concretely, one can write:

$$\mathbf{X}^{(n+1)} = \underset{\gamma f_2}{\text{prox}}[\mathbf{X}^{(n)} - \gamma(\nabla f_1(\mathbf{X}^{(n)}) + \mathcal{R}^*(\mathbf{A}^{(n)}))] \tag{11}$$

$$\mathbf{A}^{(n+1)} = \underset{\lambda f_3^*}{\text{prox}}[\mathbf{A}^{(n)} + \lambda\mathcal{R}\big(2\mathbf{X}^{(n+1)} - \mathbf{X}^{(n)}\big)], \tag{12}$$

[1]In many studies, $\mathcal{R}$ can be written as a matrix. In the case, MOS is known as a primal dual splitting [22], [23].



(a) Original (b) Noisy (10.25) (c) k-NN (10.23) (d) FastCPC (10.34) (e) GraphDAU (11.67) (f) Proposed (Model) (10.84) (g) Proposed (DAU) (11.97)

Fig. 3: CPC denoising results of `Sarah` with $(\sigma_{\text{RGB}}, \sigma_{\text{3-D}}) = (0.2, 0.3)$. (a) original and (b) noisy signals, and (c)–(g) visualizations of squared errors. PSNRs are also shown. Red points correspond to large errors and blue ones small.

TABLE II: Average PSNR of denoised CPCs.

Category	Method	$(\sigma_{\text{RGB}}, \sigma_{\text{3-D}}) = (0.1, 0.1)$	$(\sigma_{\text{RGB}}, \sigma_{\text{3-D}}) = (0.1, 0.2)$	$(\sigma_{\text{RGB}}, \sigma_{\text{3-D}}) = (0.2, 0.2)$	$(\sigma_{\text{RGB}}, \sigma_{\text{3-D}}) = (0.2, 0.3)$
Model	FastCPC [21]	15.24	15.24	9.71	9.71
	k-NN [1]	**16.12**	15.38	9.76	9.64
	Proposed (Model)	15.15	13.21	10.81	10.43
DAU	GraphDAU [7]	15.23	15.03	11.24	11.07
	Proposed (DAU)	15.79	**15.44**	**12.39**	**11.10**
	Noisy	15.15	15.15	9.61	9.61

where $\mathcal{R}^*$ is the adjoint operator of $\mathcal{R}$, $\mathbf{X}$ and $\mathbf{A}$ are the updating variables, f_3^* is the Fenchel conjugate of f_3[2].

We assign the terms in (8) into the form of (10) as

$$f_1(\mathbf{X}) = 2\|\mathbf{X}_1 - \mathbf{Y}\|_F^2 + 2\|\mathbf{X}_2 - \mathbf{Y}\|_F^2 + 2\|\mathbf{X}_1 - \mathbf{X}_2\|_F^2,$$
$$f_2(\mathbf{X}) = 0,$$
$$f_3(\mathcal{R}(\mathbf{X})) = \lambda_1\|\mathbf{B}\,\mathbf{X}_1\|_1 + \lambda_2\|\mathcal{D}(\mathbf{X}_2)\|_1 + \lambda_3\|\mathbf{X}_1 - \mathbf{X}_2\|_1.$$

As can be seen, all regularization terms are collected into $f_3(\mathcal{R}(\mathbf{X}))$. Although $\mathcal{D}$ is not linear, $\|\mathcal{D}(\mathbf{X}_2)\|_1$ is still convex according to $\mathbf{X}_2$. As a result, the subgradient of $f_3(\mathcal{R}(\mathbf{X}))$ is monotone and applicable to MOS.

REFERENCES

[1] M. A. Irfan and E. Magli, "3D point cloud denoising using a joint geometry and color k-NN graph," in *2020 28th Eur. Signal Process. Conf. EUSIPCO*, 2021, pp. 585–589.

[2] S. Ono, I. Yamada, and I. Kumazawa, "Total generalized variation for graph signals," in *2015 IEEE Int. Conf. Acoust. Speech Signal Process. ICASSP*, 2015, pp. 5456–5460.

[3] 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, 2013.

[4] 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, 2018.

[5] R. Fabio, "Image-based 3D Modelling: A Review," 2006.

[6] S. Soudarissanane, R. Lindenbergh, M. Menenti, and P. Teunissen, "Scanning geometry: Influencing factor on the quality of terrestrial laser scanning points," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 4, pp. 389–399, 2011.

[7] M. Nagahama, K. Yamada, Y. Tanaka, S. H. Chan, and Y. C. Eldar, "Graph signal restoration using nested deep algorithm unrolling," *IEEE Trans. Signal Process.*, vol. 70, pp. 3296–3311, 2022.

[8] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, 2013.

[2]The Fenchel conjugate is defined as $f^*(\xi) := \sup_{\mathbf{x}}\{\langle \mathbf{x}, \xi\rangle - f(\mathbf{x})\}$, and the step sizes $\gamma, \lambda > 0$ satisfy suitable conditions for convergence.

[9] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2017.

[10] P. Tseng, "A modified forward-backward splitting method for maximal monotone mappings," *SIAM J. Control Optim.*, vol. 38, no. 2, pp. 431–446, 2000.

[11] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, 2021.

[12] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovacevic, "Signal denoising on graphs via graph filtering," in *2014 IEEE Glob. Conf. Signal Inf. Process. Glob.*, 2014, pp. 872–876.

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[14] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2018, arXiv:1710.10903.

[15] S. H. Chan, "Performance analysis of plug-and-play ADMM: A graph signal processing perspective," *IEEE Trans. Comput. Imaging*, vol. 5, no. 2, pp. 274–286, 2019.

[16] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM J. Imaging Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017.

[17] H. Kojima, H. Higashi, and Y. Tanaka, "Interpretable graph signal denoising using regularization by denoising," in *2024 32nd Eur. Signal Process. Conf. EUSIPCO*, 2024, pp. 2322–2326.

[18] H. Trevor, T. Robert, and F. Jerome, "Elements of statistical learning: Data mining, inference, and prediction. 2nd edition." 2009.

[19] N. Parikh and S. Boyd, "Proximal algorithms," *OPT*, vol. 1, no. 3, pp. 127–239, 2014.

[20] C. Loop, Q. Cai, S.Orts-Escolano, and P.A.Chou, "Microsoft voxelized upper bodies - a voxelized point cloud dataset," *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m38673/M72012*, May 2016.

[21] R. Watanabe, K. Nonaka, E. Pavez, T. Kobayashi, and A. Ortega, "Fast graph-based denoising for point cloud color information," in *ICASSP 2024 - 2024 IEEE Int. Conf. Acoust. Speech Signal Process. ICASSP*, 2024, pp. 4025–4029.

[22] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *J Optim Theory Appl*, vol. 158, no. 2, pp. 460–479, 2013.

[23] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Adv Comput Math*, vol. 38, no. 3, pp. 667–681, 2013.