Cao *et al.*
EURASIP Journal on Advances in Signal Processing (2024) 2024:38
https://doi.org/10.1186/s13634-024-01133-3

EURASIP Journal on Advances
in Signal Processing

RESEARCH **Open Access**

An efficient algorithm with fast convergence rate for sparse graph signal reconstruction

Yuting Cao[1†], Xue-Qin Jiang[1*], Jian Wang[2†], Shubo Zhou[1†] and Xinxin Hou[1†]

†Yuting Cao, Jian Wang, Shubo Zhou and Xinxin Hou contributed equally to this work.

*Correspondence:
xqjiang@dhu.edu.cn

[1] School of Information Science and Technolog, Donghua University, Shanghai 201620, China
[2] School of Data Science, Fudan University, Shanghai 200433, China

Abstract

In this paper, we consider the graph signals are sparse in the graph Fourier domain and propose an iterative threshold compressed sensing reconstruction (ITCSR) algorithm to reconstruct sparse graph signals in the graph Fourier domain. The proposed ITCSR algorithm derives from the well-known compressed sensing by considering a threshold for sparsity-promoting reconstruction of the underlying graph signals. The proposed ITCSR algorithm enhances the performance of sparse graph signal reconstruction by introducing a threshold function to determine a suitable threshold. Furthermore, we demonstrate that the suitable parameters for the threshold can be automatically determined by leveraging the sparrow search algorithm. Moreover, we analytically prove the convergence property of the proposed ITCSR algorithm. In the experimental, numerical tests with synthetic as well as 3D point cloud data demonstrate the merits of the proposed ITCSR algorithm relative to the baseline algorithms.

Keywords: Sparse graph signals, Iterative threshold compressed sensing reconstruction (ITCSR), Graph Fourier domain, Compressed sensing (CS)

1 Introduction

In recent years, there is a noteworthy surge in research interest regarding the analysis of irregular structured signals across various fields and applications. In many practical scenarios, the occurrence of missing signals within these irregular structured signals poses a critical concern. The loss of signals can be attributed to various factors, including sensor malfunctions, signal transfer losses or system maintenance activities. In this context, signal reconstruction techniques are developed to substitute the missing signals with dependable estimations. Furthermore, graph signal processing (GSP) [1–3] is introduced as an intuitive framework to deal with graph signals lying on an irregular structure, which applies to a wide class of classical use cases such as traffic [4], sensor network [5], 3D point cloud [6, 7], image processing [8], air pollution monitoring platform [9] and recommendation system [10]. In the realm of GSP tools, the task of reconstruction emerges as a straightforward approach to address the intricate challenge of estimating missing signals.

In classical situations, graph signal reconstruction is often based on some assumptions, such as smoothness, bandlimitedness, stationarity and sparsity. [11] formulated the graph model for conflict resolution (GMCR) and employed the proximal gradient algorithm to solve it. The GMCR assumed signal smoothness on the graph and the penalty term utilized in this approach is based on the quadratic form of graph total variation. However, this assumption can lead to a deteriorated reconstruction if the signals of interest have dissimilar values between the vertices that are connected with respect to the ground truth topology. [12] introduced a graph signal reconstruction technique for estimating the power spectral density (PSD) of graph signals and was further utilized to develop Wiener-type estimation procedures for signals that are both noisy and partially observed. Nevertheless, the reconstruction method proposed in [12] cannot directly estimate the PSD. Instead, the method assumed that the graph signals are stationary and stochastic. Inspired by the idea of the classical Papoulis–Gerchberg iterative scheme, [13] proposed the iterative least square reconstruction (ILSR) to reconstruct the bandlimited graph signals from sampled signals. In addition, [14] proposed two efficient graph signal reconstruction methods, namely the iterative propagating reconstruction (IPR) and the iterative weighting reconstruction (IWR). To mitigate the computational expense associated with eigen-decomposition of the graph Laplacian matrix when dealing with bandlimited graph signals, [15] leveraged Gershgorin disks to optimize a sampling criterion relying on the lower bound of the smallest eigenvalue of the graph Laplacian matrix. The methods presented in [12–15] assumed that the graph signals are bandlimited in the graph Fourier domain, and the bandlimited graph signals are also defined as sparse graph signals. However, a major difference in this paper is that bandlimited graph signals, as graph Fourier domain sparse signals, have a certain bandwidth limitation across the entire graph Fourier domain. In other words, the nonzero coefficients of bandlimited graph signals only exist in the low-frequency space which cannot fully capture all the characteristics of real-world signals.

Inspired by the theory of compressed sensing (CS), many researchers proposed a variety of signal reconstruction algorithms based on sparse representation, such as basis pursuit (BP) [16], orthogonal matching pursuit (OMP) [17] and iterative hard thresholding (IHT) [18]. These algorithms provided important theoretical foundations and methodologies for sparse graph signal reconstruction. In the references related to sparse graph signal reconstruction [19–21], notable contributions include research on modeling the sparsity of graph signals and the development of compressed sensing reconstruction algorithms based on graph signals. The concept of sparse graph signal model refers to the existence of nonzero coefficients throughout the entire graph Fourier domain, including the high-frequency space. [19] proposed iterative method with adaptive thresholding for graph interpolation (IMATGI) to sparsity-promoting reconstruction of graph signals. Furthermore, [20] presented the recovery and sampling algorithms by selecting predetermined cardinality subset of vertices that guarantees reconstruction of the graph signals with the lowest possible reconstruction error. [21] proposed three sparse graph signal reconstruction algorithms based on BP, OMP and IHT, respectively. In order to achieve efficient compression, sampling and reconstruction of signals, the compressive sampling matching pursuit (CoSaMP) algorithm was developed [22]. These references contributed important insights into understanding the sparsity of graph signals and proposed effective reconstruction algorithms. However, it is unfortunate that as the vertex count increases in the graph, the distortion in the reconstructed graph signals using these algorithms also escalates.

In this paper, we consider the graph signals are sparse in the graph Fourier domain. For the sparse graph signal reconstruction, we propose an iterative threshold compressed sensing reconstruction (ITCSR) algorithm, which effectively addresses the challenges associated with signal reconstruction. The proposed ITCSR algorithm enhances the performance of sparse graph signal reconstruction by introducing a threshold function to determine a suitable threshold. Furthermore, we demonstrate that the suitable parameters for the threshold can be automatically determined by leveraging the sparrow search algorithm (SSA) [26]. Convergence analysis of the proposed ITCSR algorithm is provided. Experiments on both synthetic datasets and 3D point cloud datasets demonstrate that the reconstruction performance of the proposed ITCSR algorithm outperforms that of the baseline algorithms.

The main contributions in this paper are summarized as follows.

1. It proposes a new algorithm, referred to as ITCSR, for solving sparse graph signal reconstruction problem.
2. It gives a theoretical analysis of the convergence property of the proposed algorithm by utilizing the restricted isometry property (RIP).
3. It demonstrates that the proposed ITCSR algorithm indeed produces a significant improvement in terms of sparse graph signal reconstruction performance in publicly available 3D point cloud datasets.

1.1 Paper overview

The remainder of this paper is organized as follows. Some related basic definitions of GSP and K-sparse graph signals are stated in Sect. . Section formulates the target model and describes the proposed ITCSR algorithm. In Sect. , the convergence analysis for the proposed ITCSR algorithm is presented. In Sect. , the proposed algorithm is compared with the baseline algorithms and experiments to validate the performance of our ITCSR algorithm. Finally, we conclude the paper in Sect. .

1.2 Notation

In this paper, we adopt the following notations. Boldface lowercase letters denote vectors, whereas boldface uppercase letters denote matrices. In the tables, we represent the optimal values in bold font. $\mathrm{diag}(\cdot)$ is a diagonal matrix with its argument along the main diagonal. $(\cdot)^\top$, $(\cdot)^{-1}$, $(\cdot)^\dagger$ denotes transpose operation, inverse operation and pseudoinverse operation, respectively. The notation $|\cdot|$ indicates the cardinality of a set. For a vector $\mathbf{f}$, $\|\mathbf{f}\|_0$ and $\|\mathbf{f}\|_2$ represent its 0-norm and Euclidean norm, respectively.

2 Preliminary

In this section, we review definitions of graph Laplacian matrix and sparse graph signals that are used in the development of the proposed ITCSR algorithm and theoretical framework.

2.1 Graph Laplacian matrix

An undirected graph without self-loops is represented as $\mathcal{G} := \{\mathcal{V}, \mathcal{E}, \mathbf{A}\}$, where $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ is the vertex set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of undirected edges connecting the vertices. $\mathbf{A} \in \mathbb{R}^{N\times N}$ indicates the adjacency matrix of the graph. The elements of $\mathbf{A}$ are defined as

$$\mathbf{A}_{ij} = \begin{cases} 1, \text{ if } (v_i, v_j) \in \mathcal{E} \\ 0, \text{ otherwise} \end{cases}, \; i,j = 1,\cdots,N. \tag{1}$$

In general, adjacency matrix $\mathbf{A}$ is a matrix used to represent the connection relationship between vertices in a graph. As an extension of the adjacency matrix, the weight matrix $\mathbf{W}$ contains the weight information of the connection. The element w_{ij} of $\mathbf{W}$ represents the weight value of the connection between vertex i and vertex j. It is important to note that edge weights are usually assumed to be nonnegative real numbers. For each vertex v_i, we define degree $d_i = \sum_{j=1}^{N} w_{ij}$ as the sum of edge weights connected to it. In this way, the degree matrix is defined as $\mathbf{D} = \operatorname{diag}(d_1, d_2, \cdots, d_N)$. Along with $\mathbf{W}$ and $\mathbf{D}$, the combinatorial Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N\times N}$ is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W}. \tag{2}$$

Since $\mathbf{L}$ is a real symmetric and positive semi-definite matrix, it admits the eigen-decomposition $\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$, where the $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \cdots, \lambda_N)$ is a diagonal matrix of nonnegative eigenvalues and the unitary matrix $\mathbf{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_N]$ containing the corresponding eigenvectors.

2.2 *K*-sparse graph signals

Graph signals are scalar-valued mapping $\{ : \mathcal{V} \mapsto \mathbb{R}$ defined on the vertices of the graph. It can be represented as a vector $\mathbf{f} = [f_1, f_2, \cdots, f_N]^\top \in \mathbb{R}^N$, where the i-th component f_i represents the value of the signal at vertex v_i. Since the Laplacian matrix $\mathbf{L}$ satisfies the property of real symmetry, all the eigenvalues of $\mathbf{L}$ are nonnegative, $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$, and the corresponding eigenvectors are orthogonal. Then, the graph Fourier transform (GFT) is defined as the projection onto the orthogonal set of the eigenvectors of $\mathbf{L}$, i.e.,

$$\hat{\mathbf{f}} = \left[\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_N\right]^\top = \sum_{i=1}^{N} \mathbf{u}_i f_i = \mathbf{U}^\top \mathbf{f}. \tag{3}$$

Similar to classical Fourier analysis, the eigenvalues $\{\lambda_i\}_{0\leq N}$ in this context represent the frequencies of the graph, while $\hat{f}_i$ represents the frequency component. And the inverse graph Fourier transform (IGFT) can be expressed as

$$\mathbf{f} = U\hat{\mathbf{f}}. \tag{4}$$

For a given set $\mathcal{K} = \{k_1, k_2, \cdots, k_K\}$, where $K \leq N$, the graph signals $\hat{\mathbf{f}}$ exhibit K-sparsity in GFT domain and can be mathematically defined as follows

$$\hat{f}_{k_l} \neq 0, \text{for } \; k_l \in \mathcal{K} \; \text{ and } \; \hat{f}_{k_l} = 0, \text{for } \; k_l \notin \mathcal{K}, \; l = 1, \cdots, K. \tag{5}$$

In other words, K-sparse graph signals $\hat{\mathbf{f}}$ can be expressed as a linear combination of $K \leq N$ eigenvectors of the graph Laplacian $\mathbf{L}$, which can be represented as

$$f_n = \sum_{l=1}^{K} \hat{f}_{k_l} \mathbf{U}_{k_l,n}, \quad n = 1, 2, \cdots, N. \tag{6}$$

A major difference is that in many previous works, it is assumed that sparse graph signals in graph Fourier domain are confined to low-frequency space, meaning that the positions corresponding to frequency components $\mathcal{K} = \{k_1, k_2, \cdots, k_K\}$ are known [5] [10]. Due to the assumption of smoothness in the graph Fourier domain and the absence of any frequency components beyond its bandwidth, the bandlimited graph signal model fails to capture the details and rapid changes in the high-frequency space of the signal. Consequently, important information may be lost during the graph signal reconstruction process. Therefore, we propose a sparse graph signal model that considers all frequency components within the entire graph Fourier domain, aiming to enhance the performance of graph signal reconstruction and better preserve crucial information within the real-word signals. It implies that the positions corresponding to frequency components $\mathcal{K} = \{k_1, k_2, \cdots, k_K\}$ are unknown.

2.3 The framework of SSA

SSA is a swarm intelligence optimization algorithm based on the behavior of sparrows foraging and avoiding predators. In this paper, the position of sparrows can be represented in the following matrix

$$X = \begin{bmatrix} \alpha_{1,1} & \beta_{1,2} \\ \alpha_{2,1} & \beta_{2,2} \\ \vdots & \vdots \\ \alpha_{n,1} & \beta_{n,2} \end{bmatrix},$$

where n is the number of sparrows. Then, the fitness value of all sparrows can be expressed by the following vector

$$F = \begin{bmatrix} g([\alpha_{1,1}, \beta_{1,2}]) \\ g([\alpha_{2,1}, \beta_{2,2}]) \\ \vdots \\ g([\alpha_{n,1}, \beta_{n,2}]) \end{bmatrix}, \tag{7}$$

where the value of each row in F represents the fitness value of the individual. During each iteration, the location update of producers is defined as

$$X_P^{l+1} = \begin{cases} X_{i,j}^{l} \exp\left(-\frac{i}{\eta l_{\max}}\right) & \text{if } r < S_T \\ X_{i,j}^{l} + QI & \text{otherwise} \end{cases}, \; i = 1, \cdots, n, \text{and } j = 1, 2, \tag{8}$$

where $X_{i,j}$ represents the element in the i-th row and j-th column of the matrix X. Q is a random number following a normal distribution and I represents a matrix with all elements being 1. $\eta \in (0, 1]$ is a random number and $l_{\max}$ represents the number of maximum iteration. $r \in [0, 1]$ and $S_T \in [0.5, 1]$ represent the alarm value and the safety threshold, respectively. Similarly, the location update of scroungers is defined as

$$X_S^{l+1} = \begin{cases} Q \exp\left(\frac{X_w^l - X_{i,j}^l}{i^2}\right) & \text{if } \; i > j/2 \\ X_P^{l+1} + \left|X_{i,j} - X_P^{l+1}\right| \cdot C^{\dagger} I & \text{otherwise} \end{cases}, \tag{9}$$

where X_w represents the current globally worst position of the individual and C represents a matrix, where each element is randomly assigned a value of 1 or -1. When sparrow detect danger and the alarm value more than the safety value, the location updated by

$$X_A^{l+1} = \begin{cases} X_b^l + \zeta \left|X_{i,j}^l - X_b^l\right| & \text{if } \; F_i > F_b \\ X_{i,j}^l + R\left(\frac{\left|X_{i,j}^l - X_w^l\right|}{(F_i - F_w) + \varepsilon}\right) & \text{if } \; F_i = F_b \end{cases}, \tag{10}$$

where X_b represents the current globally best position of the individual. As the control parameter for step size, ζ is a normal distribution of random numbers with a mean of 0 and a variance of 1, ensuring the original meaning remains unchanged. F_i represents the current fitness values of the individual, F_b and F_w are the current global best and worst fitness values of the individual, respectively. In this paper, F_b is defined as

$$\hat{i} = \underset{\alpha_{i,1} \in X, \beta_{i,2} \in X}{\arg\min} \; g(\alpha_{i,1}, \beta_{i,2}), \alpha = \alpha_{\hat{i},1}, \beta = \beta_{\hat{i},2} \tag{11}$$

The pseudo-code for the model proposed in this study is shown in Algorithm 1.

Algorithm 1 The framework of SSA

```
Input:
• $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathbf{A})$: Compute Laplacian matrix
• $\mathbf{y}$: Sampled graph signal
• $l_{\max}$: The number of maximum iteration
• $p$: Percentage of producers
• $r$: Alarm value
• $S_T$: Safe value
• $\gamma, \zeta, \eta$: Algorithm parameters
Output:
• $\alpha$
• $\beta$
Iteration:
• Initialization $p = 0.2$, $l_{\max} = 100$, $l = 0$
• while $l < l_{\max}$ do
    - Calculate the $F$ by (7)
    - Update the location of producers by (8)
    - Update the location of scroungers by (9)
    - Update the location of alarmers by (10)
    - Update the $F_i$ and the sparrow position
    - Calculate the $\alpha$ and $\beta$ by (11)
    - $l = l + 1$
  end
```

3 The proposed ITCSR algorithm for *K*-sparse graph signals

In this section, we consider the reconstruction of K-sparse graph signals. First, we formulate the K-sparse graph signals reconstruction as solving an underdetermined equation system. We then present the proposed ITCSR algorithm to solve this problem and show the pseudo-code of the proposed ITCSR algorithm.

3.1 Problem formulation

K-sparse graph signals $\hat{\mathbf{f}} \in \mathbb{R}^N$ in the graph Fourier domain, with sparsity K, can be reconstructed using a reduced number of available samples, denoted as $M < N$. These samples correspond to a random subset $\mathcal{S} = \{m_1, m_2, \cdots, m_M\}$, selected from the vertex set $\mathcal{V} = \{1, 2, \cdots, N\}$. The measurements vector containing the available samples is defined as

$$\mathbf{y} = \left[f_{m_1}, f_{m_2}, \cdots, f_{m_M}\right]^\top, \tag{12}$$

where f_{m_M} represents the sampled graph signals. According to (4), it then yields the following form

$$\mathbf{y} = \mathbf{U}_{MN}\hat{\mathbf{f}} = \Phi\hat{\mathbf{f}}, \tag{13}$$

where the matrix $\mathbf{U}_{MN}$ is obtained by selecting the rows from the matrix $\mathbf{U}$ that correspond to the set $\mathcal{S}$. For simplicity, $\mathbf{U}_{MN}$ is represented by Φ. (13) denotes the underdetermined system of M equations and N unknowns ($M < N$), which can be formulated as

$$\begin{bmatrix} f_{m_1} \\ f_{m_2} \\ \cdots \\ f_{m_M} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{1,m_1} & \cdots & \mathbf{U}_{N,m_1} \\ \mathbf{U}_{1,m_2} & \cdots & \mathbf{U}_{N,m_2} \\ & \cdots & \\ \mathbf{U}_{1,m_M} & \cdots & \mathbf{U}_{N,m_M} \end{bmatrix} \begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \\ \cdots \\ \hat{f}_N \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{k_1,m_1} & \cdots & \mathbf{U}_{k_K,m_1} \\ \mathbf{U}_{k_1,m_2} & \cdots & \mathbf{U}_{k_K,m_2} \\ & \cdots & \\ \mathbf{U}_{k_1,m_M} & \cdots & \mathbf{U}_{k_K,m_M} \end{bmatrix} \begin{bmatrix} \hat{f}_{k_1} \\ \hat{f}_{k_2} \\ \cdots \\ \hat{f}_{k_K} \end{bmatrix}. \tag{14}$$

It is obvious that (14) is an underdetermined system of equations and $\hat{\mathbf{f}}$ is not uniquely reconstructive from $\mathbf{y}$ by linear algebraic means, as (14) may have many solutions. However, we aim to find a sparse solution, and sparsity can serve as a powerful constraint for specific measurement matrices Φ. It means that (14) can be solved when the position $\mathcal{K} = \{k_1, k_2, \cdots, k_K\}$ is known. The formulation of the K-sparse graph signal reconstruction in this paper considers the scenario where the positions of nonzero GFT coefficients are unknown, which can be formulated as

$$\min \|\hat{\mathbf{f}}\|_0 \ \text{ subject to } \ \mathbf{y} = \Phi\hat{\mathbf{f}}, \tag{15}$$

where the operator $\|\cdot\|_0$ is used as a sparsity measure.

3.2 The proposed ITCSR algorithm

This subsection introduces the proposed ITCSR algorithm to solve (15). The objective of ITCSR is to obtain an approximate solution to $\mathbf{y} = \Phi\hat{\mathbf{f}}$. Now, we will outline its fundamental components. We denote $\boldsymbol{\eta}$ as a vector of residual correlations and $\mathbf{r}$ as a residual vector. Next, the maximum function of the residual correlations can be expressed as follows

$$\mathcal{M}(\boldsymbol{\eta}) = \max(\boldsymbol{\eta}). \tag{16}$$

On this basis, a subvector is defined as $\mathbf{b} \in \mathbb{R}^{N-1}$ that represents the vector obtained by removing $\mathcal{M}(\boldsymbol{\eta})$ from the vector $\boldsymbol{\eta}$. After that, a threshold function V_{th} can be denoted as

$$\mathrm{V}_{th} = \alpha[\mathcal{M}(\boldsymbol{\eta}) + \beta\mathcal{M}(\boldsymbol{b})], \tag{17}$$

where α and β are parameters and can be optimized by SSA. In this paper, we set the fitness value $F = \min \|\Phi_{\mathcal{C}}^{\top}\mathbf{y} - \mathbf{y}\|_2^2$, *s.t.* $\mathcal{C} = \left\{q : \left|\Phi^{\top}\mathbf{y}\right| \geq \alpha\left[\mathcal{M}(\Phi^{\top}\mathbf{y}) + \beta\mathcal{M}(\boldsymbol{\zeta})\right]\right\}$, where $\boldsymbol{\zeta}$ represents the vector obtained by removing $\mathcal{M}(\Phi^{\top}\mathbf{y})$ from the vector $\Phi^{\top}\mathbf{y}$. According to the SSA algorithm, the values of parameters α and β should fall within the range [0, 1]. In conjunction with our threshold function, when α is set to 0, the threshold setting becomes unreasonable. Therefore, in this paper, the value of α is set to (0, 1]. After discussing these descriptions of the basic ingredients, we state the proposed ITCSR algorithm in detail. ITCSR is an iterative and greedy algorithm. At each step, ITCSR selects several elements from the residual correlation vector $\boldsymbol{\eta}$, and adds its index to the identified support set $\mathcal{K}$. ITCSR performs least square to estimate the signal. We let t be the t-th iteration of the ITCSR algorithm. Specifically, for the following descriptions, we let superscripts denote the iteration number of the ITCSR algorithm. ITCSR starts with initial graph signals $\hat{\mathbf{f}}^0 = 0$ and initial residual vector $\mathbf{r}^0 = \mathbf{y}$. The ITCSR algorithm also contains a set of estimates $\mathcal{K}$ of the positions of the nonzero coefficients and the initial estimated positions $\mathcal{K}^0 = \emptyset$. The ITCSR algorithm, proposed as a solution to the considered problem, is composed of three fundamental steps:

- *Position Estimation*: Estimate positions $\mathcal{K} = \{k_1, k_2, \cdots, k_K\}$ of nonzero coefficients.
- *Coefficient Reconstruction*: Reconstruct nonzero coefficients $\hat{f}_k$ at estimated positions k.
- *Signal Reconstruction*: Reconstruct original graph signals $\mathbf{f}$.

At the step of *Position Estimation*, the t-th iteration performs the inner product of the current residual and the measurement matrix $\boldsymbol{\Phi}$ to obtain the residual correlation vector

$$\boldsymbol{\eta}^t = \left|\left\langle \boldsymbol{\Phi}, \mathbf{r}^{t-1} \right\rangle\right| = \boldsymbol{\Phi}^{\top}\mathbf{r}^{t-1}, \tag{18}$$

which contains a small number of significant nonzeros in each entry. After a simple calculation, we could easily get the $\boldsymbol{\eta}^t$. The ITCSR algorithm performs thresholding to find the significant nonzeros. Thresholding yields an index set

$$\mathbf{L}^t = \left\{i : \left|(\boldsymbol{\eta}_i)^t\right| \geq \alpha\left[\mathcal{M}(\boldsymbol{\eta}^t) + \beta\mathcal{M}(\mathbf{b}^t)\right]\right\}, \tag{19}$$

where $(\boldsymbol{\eta}_i)^t$ represents the i-th element in the vector $\boldsymbol{\eta}$ at t-th iteration. Then, we merge the subset of newly selected coordinates with the previous support estimate, thereby updating the estimated available positions

$$\mathcal{K}^t = \mathbf{L}^t \cup \mathcal{K}^{t-1}. \tag{20}$$

At the step of *Coefficient Reconstruction*, according to estimated positions, we can calculate the nonzero coefficients by projecting the vector $\mathbf{y}$ onto the columns of $\boldsymbol{\Phi}$. Letting $\boldsymbol{\Phi}_{\mathcal{K}}$ denote the $M \times |\mathcal{K}|$ matrix with columns chosen using index set $\mathcal{K}$. The updated $(\hat{\mathbf{f}}')^t$ is defined as

$$(\hat{\mathbf{f}}')^t = (\boldsymbol{\Phi}_{\mathcal{K}^t}^{\top} \boldsymbol{\Phi}_{\mathcal{K}^t})^{-1} \boldsymbol{\Phi}_{\mathcal{K}^t}^{\top} \mathbf{y}. \tag{21}$$

In combination with (20), the estimation of $\hat{\mathbf{f}}$ is defined as

$$(\hat{\mathbf{f}}'_n)^t = \begin{cases} (\hat{\mathbf{f}}'_k)^t & \text{if } n = k, k \in \mathcal{K}^t \\ 0 & \text{otherwise} \end{cases}, n = 1, 2, \cdots, N. \tag{22}$$

By combining (20) and (22), the updated residual vector can be expressed as follows

$$\mathbf{r}^t = \mathbf{y} - \boldsymbol{\Phi}_{\mathcal{K}^t} (\hat{\mathbf{f}}')^t. \tag{23}$$

At the step of *Signal Reconstruction*, in accordance with the estimated positions and the corresponding nonzero coefficients, the reconstructed graph signals $(\hat{\mathbf{f}}')^t$ are calculated by (4)

$$(\mathbf{f}')^t = \mathbf{U}(\hat{\mathbf{f}}')^t. \tag{24}$$

Algorithm 2 The ITCSR algorithm

```
Requier:
  • $\mathscr{G} = (\mathscr{V}, \mathscr{E})$: Compute Laplacian matrix
Input:
  • $\mathscr{S}$: Sampling indexes
  • $\epsilon$: Termination conditions
  • $(\alpha, \beta)$: Algorithm parameters
Output:
  • $\mathbf{f}^{'}$: Reconstructed signals
Algorithm:
  • Initialization $\mathbf{y}$ by (13), $\mathbf{f}^0 = \mathbf{y}$, $\hat{\mathbf{f}}^0 = \mathbf{0}^{N\times 1}$, $t = 1$
  • while $(\|(\mathbf{f}')^t - (\mathbf{f}')^{t-1}\|_2 > \epsilon)$ do
      - Calculate the vector of residual correlations by (18)
      - Perform the thresholding by (19)
      - Obtain the estimate positions of non-zero coefficients by (20)
      - Estimate non-zero coefficients by (22)
      - Reconstruct the signal by (24)
      - $\mathbf{f}' = (\hat{\mathbf{f}}')^t$
      - $t = t + 1$
    end
```

Algorithm 2 provided a summary of the proposed ITCSR algorithm. It is worth noticing that the thresholding strategy used in ITCSR algorithm utilizes the nonlinear approach that will allow several elements to be selected in (19). This threshold policy allows the proposed ITCSR algorithm to converge in fewer iterations.

4 Convergence analysis

In this section, we first introduce the restricted isometry constant (RIC) [23–25] for ease of theorem statement. Then, analytically prove the convergence of the proposed ITCSR algorithm through the introduction of Theorem 1. Our convergence analysis method cites reference [34].

The δ_K represents an upper limit on the singular values of any submatrix of $\mathbf{\Phi}$ containing K or fewer elements, and it is symmetric. The RIC (Restricted Isometry Constant) is defined as the minimum value that satisfies the following condition

$$(1-\delta_K) \leq \frac{\|\mathbf{\Phi a}\|_2^2}{\|\mathbf{a}\|_2^2} \leq (1+\delta_K). \tag{25}$$

The condition holds true for all $\mathbf{a}$ vectors that contain at most K nonzero elements. If the matrix has a small constant δ_K, then every submatrix is very nearly orthogonal.

Theorem 1 *For any observation $\mathbf{y} = \mathbf{\Phi}\hat{f}$, the ITCSR algorithm will end up with selecting K nonzero elements if the threshold optimization parameters α, β satisfy the following condition,*

$$\delta_{K+1} < \frac{\alpha(1+\beta)}{\sqrt{K}+\alpha(1+\beta)}. \tag{26}$$

The estimate signal $\tilde{\mathbf{y}}^t = \Phi\hat{\mathbf{f}}^t$ satisfies

$$||\mathbf{y}-\tilde{\mathbf{y}}^t||_2 \leq c||\mathbf{y}-(\tilde{\mathbf{y}}^*)^t||_2, \tag{27}$$

where

$$c = \sqrt{1+\left(\frac{1}{\alpha(1+\beta)\sqrt{\frac{1-\delta_K}{K}}-\frac{\delta_{K+1}}{\sqrt{1-\delta_K}}}\right)^2}, \tag{28}$$

and where α and β as defined in Sect. . The superscript t represents the current iteration number. $(\tilde{\mathbf{y}}^*)^t = \mathbf{\Phi}_{\Gamma^*}\hat{\mathbf{f}}_{\mathbf{\Gamma}^*}$ and $\mathbf{\Gamma}^*$ is the index set of the largest K elements in $\hat{\mathbf{f}}$. ■

The mathematical proof for Theorem 1 is available in the Appendix. Theorem 1 states that the proposed ITCSR algorithm guarantees a solution within a finite number of iterations. In general, achieving an exact reconstruction relies on having a RIP with a small RIC. Based on this relationship, a smaller value of δ_K corresponds to a smaller constant c in equation (28). In equation (27), it is evident that a smaller value of c corresponds to a smaller reconstruction error. Similarly, equation (28) demonstrates how c changes with different values of the threshold optimization parameters α and β. In this paper, the value range of α is (0, 1] and β is [0, 1] [26]. When α and β approach their upper bounds, c tends to decrease. Additionally, when δ_K falls within the range of (0, 1), a smaller value of c leads to better reconstruction performance.

5 Experiments

This section conducts numerical experiments to gain insights into the proposed ITCSR algorithm and assess its reconstruction performance. Unless specified otherwise, we utilize GSPBOX in MATLAB for graph signal processing, graph construction and visualizations. Firstly, we test the proposed ITCSR algorithm with synthetic data under different sampling methods and compare the results with baseline algorithms. The compared algorithms are CoSaMP, ILSR and IMATGI. The computational complexity of the ILSR, CoSaMP, IMATGI and ITCSR algorithms is shown in Table 1, where K represents the level of sparsity and t represents the number of iterations. The computational complexity of the ITCSR algorithm is $O(tK \log N)$. Secondly, we apply the proposed algorithm to 3D point cloud model datasets [39] which are the *Block*, *Cactus*, *Fandisk*, *Skull*, *Gargo* and *Dino*, respectively. The experiments are conducted using MATLAB R2018a on a desktop computer equipped with an Intel Core i7-10700 CPU and 32GB RAM. Three metrics, namely maximum absolute error (MAX), mean square error (MSE) and signal noise ratio (SNR) are employed to evaluate the reconstruction performance of all algorithms. MAX is defined as

$$\mathrm{MAX} = \max\left\{f_i' - f_i\right\}, i = 1, 2, \cdots, N, \tag{29}$$

MSE is defined as

$$\mathrm{MSE} = \frac{\left\|\sum_{i=1}^{N}\left(f_i' - f_i\right)\right\|_2^2}{N_f}, \tag{30}$$

and SNR is defined as

$$\mathrm{SNR} = \frac{\|\mathbf{f}\|_2^2}{\|\mathbf{f} - \hat{\mathbf{f}}\|_2^2}, \tag{31}$$

, respectively, where N_f represents the graph signal length. In the following experiments, the parameters α and β are selected by SSA. Smaller MAX and MSE values indicate better reconstruction performance. To guarantee exact reconstruction from compressed measurements, one should choose a specific matrix which satisfies the RIP. It is notable that the coherence of a matrix $\mathbf{\Phi}$ is a computable property that offers concrete guarantees for reconstruction. The largest inner product between any two columns ϕ_i and ϕ_j of matrix $\mathbf{\Phi}$ defines the coherence of the matrix $\mathbf{\Phi}$, denoted as

Table 1 Comparison of computational complexity

Algorithm	Computational complexity
ILSR	O(N^3)
CoSaMP	O($tK(N+M)$)
IMATGI	O(tKN)
ITCSR	O($tK\log(N)$)

$$\mu(\mathbf{\Phi}) = \max_{1 \le i < j \le N} \frac{\left|\left\langle \phi_i, \phi_j \right\rangle\right|}{\|\phi_i\|_2^2 \left\|\phi_j\right\|_2^2}. \tag{32}$$

According to the Welch bound [28], the lower bound of $\mu(\mathbf{\Phi})$ is $\sqrt{\frac{N-M}{M(N-1)}}$. It is possible to show that the coherence of a matrix is always in the range $\left[\sqrt{\frac{N-M}{M(N-1)}}, 1\right]$. Similarly, in accordance with Geršgorin circle theorem [29], we can obtain that $\mathbf{\Phi}$ satisfies the RIP of order K with $\delta_K = (K-1)\mu(\mathbf{\Phi})$ if $\mathbf{\Phi}$ has coherence $\mu(\mathbf{\Phi})$ for all $K < \frac{1}{\mu(\mathbf{\Phi})}$. Based on this new relation, the coherence in this paper should be in the range $\left[\sqrt{\frac{N-M}{M(N-1)}}, \frac{1}{2K-1}\right]$ that guarantees reconstruction uniqueness.

5.1 Experiments on synthetic data

In this subsection, the performance of the proposed ITCSR algorithm is shown by two numerical experiments. In the experiments, two synthetic datasets are adopted, including the sensor network dataset and the swiss roll dataset.

5.1.1 Synthetic data: sensor network

In this experiment, we use the sensor network dataset to illustrate the reconstruction performance by the proposed ITCSR algorithm. To begin, we establish the default experimental setup in this experiment. The sensor network dataset is generated with $N = 600$ by

$$\begin{cases} x &= \frac{1}{\lceil\sqrt{N}\rceil * [x \sim U(0,1)]} + \frac{i}{\lceil\sqrt{N}\rceil} \\ y &= \frac{1}{\lceil\sqrt{N}\rceil * [y \sim U(0,1)]} + \frac{j}{\lceil\sqrt{N}\rceil} \end{cases}, \{i,j = 0, \cdots, N-1\}, \tag{33}$$

where $\lceil\sqrt{N}\rceil$ denotes the smallest integer that is not less than $\sqrt{N}$ and $x \sim U(0,1)$ represents a random variable x that follows uniform $U(0, 1)$ distribution. Then, we construct an undirected graph by 6-nearest neighboring algorithm. The weight matrix $\mathbf{W}$ can be obtained by assigning a value of 1 if there is an edge between two sensors, and it is 0 otherwise. With $\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^\top$ denoting the eigen-decomposition of graph Laplacian, the K-sparse graph signals $\mathbf{f}$ are generated as $\boldsymbol{f} = \boldsymbol{U}\hat{\boldsymbol{f}}$, where $\hat{f}$ are independent and identically distributed (*i.i.d*) Gaussian distribution and the $K = 4$ entries with most significant absolute values in the random signal are kept while the rest are set to zero. As for the selection of the sample set in this experiment, we mainly chose the following methods: maxsigmin (M1) [30], maxvolume (M2) [31], maxigmin (M3) [32] and random sampling (M4). For simplicity, the strategy of M4 in this paper includes the following two steps. First, initializing an array of size N, where N is the number of elements in the permutation. Fill the array with the values 1 to N and randomly shuffle the numbers. Then, the first M number is selected as the sampling vertex.

In this paper, we conducted signal reconstructions using various parameter values to investigate the influence of parameter settings on the performance of the reconstruction algorithm. Table 2 shows the MSEs results under different sampling methods. Different parameters may lead to different performances for different sampling methods. As

Table 2 MSE (e-04) results by different sampling methods on the synthetic dataset with the level of sparsity $K = 10$

(α, β)	**Method**	**0.5**	**0.55**	**0.6**	**0.65**	**0.7**	**0.75**	**0.8**	**0.85**
(1,0)	M1	9.3151	9.2130	8.8923	8.2633	7.9741	7.2571	6.9424	6.7387
	M2	9.4287	8.7822	8.3408	8.0105	7.6194	6.9537	6.4531	6.1303
	M3	8.3720	7.7240	6.9148	6.1557	5.8381	5.3223	5.2342	4.9766
	M4	**6.3576**	**6.0159**	**5.3556**	**4.7726**	**4.5479**	**4.1092**	**3.6688**	**3.2437**
(0.5, 0.5)	M1	8.3209	8.0074	7.2315	7.0279	6.5951	6.1238	6.0080	5.8691
	M2	7.0078	6.6124	6.3325	6.1226	5.9933	5.5821	5.2836	5.2012
	M3	7.1729	6.1226	6.0600	5.3581	5.1465	4.7184	4.5138	4.3031
	M4	**6.0781**	**5.6632**	**5.1807**	**4.7119**	**3.6494**	**3.0292**	**2.8508**	**2.6001**
(0.6733, 0.3076)	M1	6.2298	6.1240	5.8017	5.6523	4.8146	4.3353	4.1801	4.0527
	M2	5.6647	5.3052	4.6223	4.0760	3.9889	3.6483	3.1214	3.0013
	M3	4.6444	4.3012	3.9950	3.3606	3.0927	2.7887	2.5122	2.1321
	M4	**4.3353**	**3.4229**	**2.8767**	**2.2077**	**1.7236**	**1.2061**	**0.9031**	**0.8125**

Table 3 Coherence of matrix $\mathbf{\Phi}$ under different sampling methods on the synthetic dataset with the level of sparsity $K = 10$

Sampling ratio	**0.5**	**0.55**	**0.6**	**0.65**	**0.7**	**0.75**	**0.8**	**0.85**
M1	0.0517	0.0509	0.0492	0.0471	0.0426	0.0411	0.0381	0.0352
M2	0.0504	0.0497	0.0485	0.0459	0.0409	0.0378	0.0351	0.0312
M3	**0.0489**	0.0469	0.0449	0.0415	0.0354	0.0331	0.0314	0.0303
M4	0.0493	**0.0455**	**0.0419**	**0.0382**	**0.0303**	**0.0257**	**0.0216**	**0.0182**

shown in Table 2, the values of parameters α and β set to (1, 0) and (0.5, 0.5) by randomly and (0.4043, 0.6886) by SSA. As can be seen, the reconstruction performance improves as the value of optimal parameters α and β are obtained by SSA. In particular, the MSE of the reconstructed signal by the proposed ITCSR algorithm can be reach $3.2383e-04$ when sampling rate is 0.5 after adjusting the threshold parameters by SSA, where $e-04$ represents a value where the decimal point is moved four places to the left. Then, to verify the convergence of the proposed ITCSR algorithm, we calculate the coherence of the matrix $\mathbf{\Phi}$ under different sampling rates with different sampling methods in Table 3. First, we used the above four sampling methods to generate 100 groups of data at different sampling rates and then substituted the data of each group in (32) to calculate coherence and finally took the average value. It can be seen that the coherence of matrix $\mathbf{\Phi}$ guarantees uniqueness.

In Fig. 1, to reveal the impact of the value K on ITCSR algorithm, we reconstructed the sensor network graph signals using different measurements M and the value K. As also described in Fig. 1, percentage reconstructed can achieve better performance under the same value K when measurements M increased. To compared the reconstruction performance of the proposed algorithm with baseline algorithms, we set the sparsity factor K/N from 10% to 60%. For each sparsity factor and different algorithms, we reconstruct using 20 iterations and report the average achieved SNR in Fig. 2. As observed in Fig. 2, all curves experience a sudden knee like fall in reconstruction SNR as the sparsity factor increases. As expected, the proposed ITCSR algorithm can successfully reconstruct

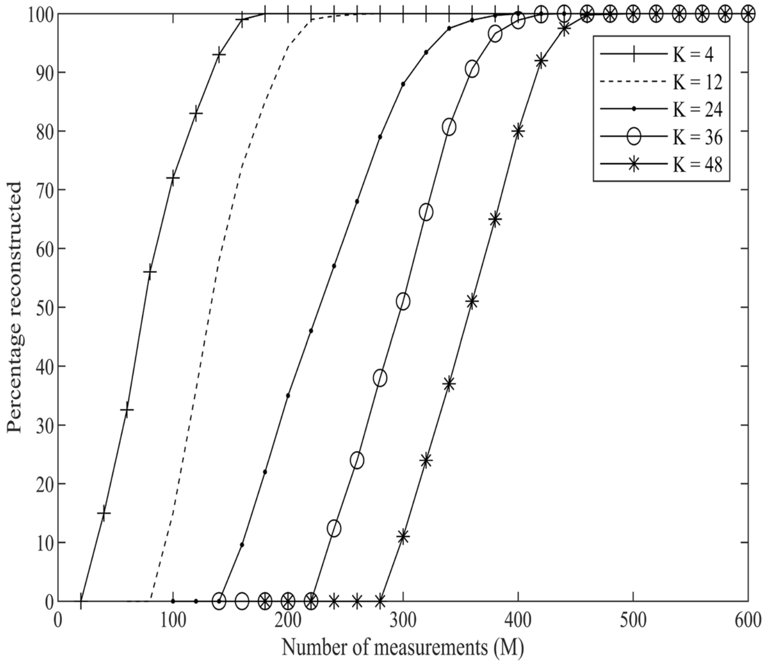


Fig. 1 Percentage of sensor network graph signals reconstructed correctly (N=600)

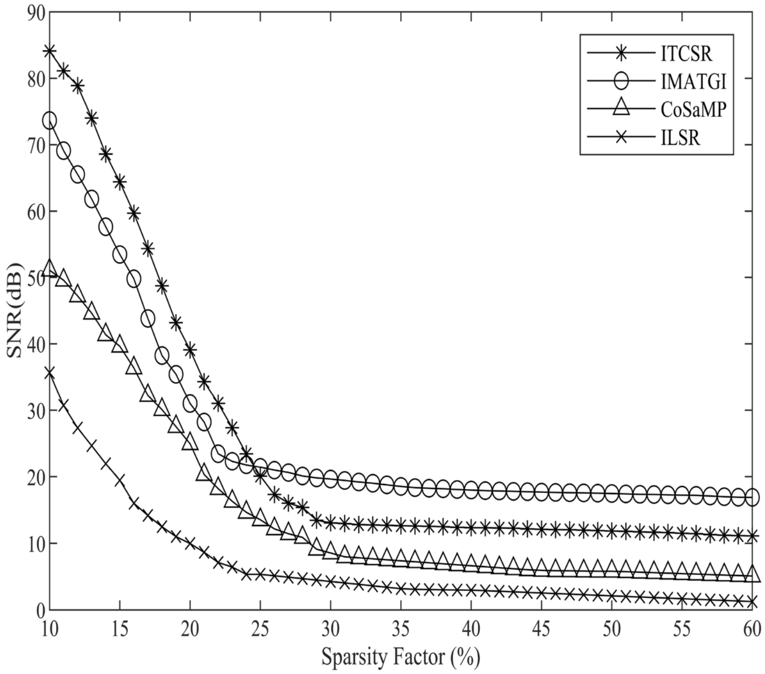


Fig. 2 The reconstruction performance of different algorithms when the value of K changes

less sparse signals. Figure 3 shows the MSE versus the number of iterations, with 60% sampling rate. It is observed that the proposed ITCSR algorithm exhibits both the fast convergence and the minimum reconstruction error. Overall, the proposed ITCSR

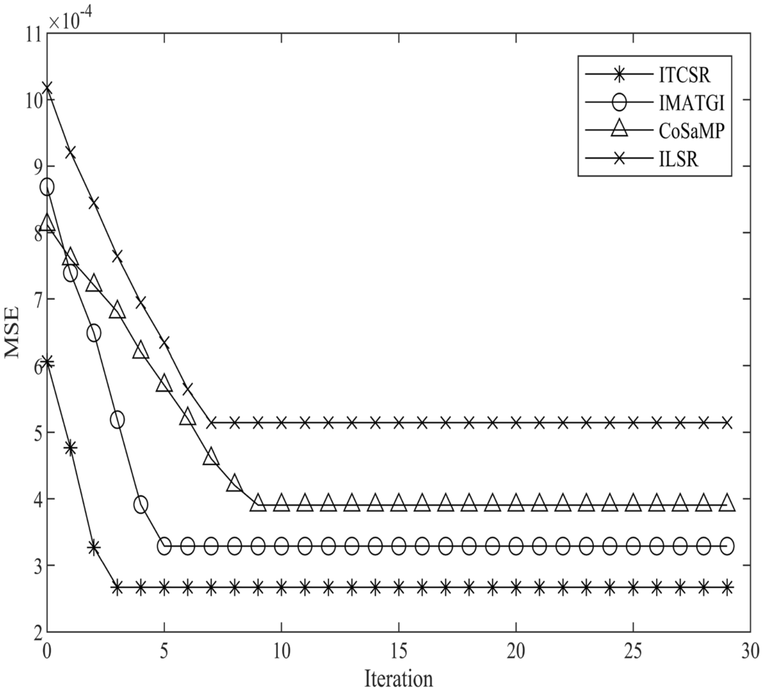


Fig. 3 MSE versus the iteration number on the sensor network dataset

algorithm can achieve good reconstruction performance while maintaining fast convergence. To evaluate the reconstruction performance of the ITCSR algorithm against conventional sensor signal reconstruction algorithms, we conducted a specific experiment. In this experiment, we additionally introduced a relative threshold-based sparsity

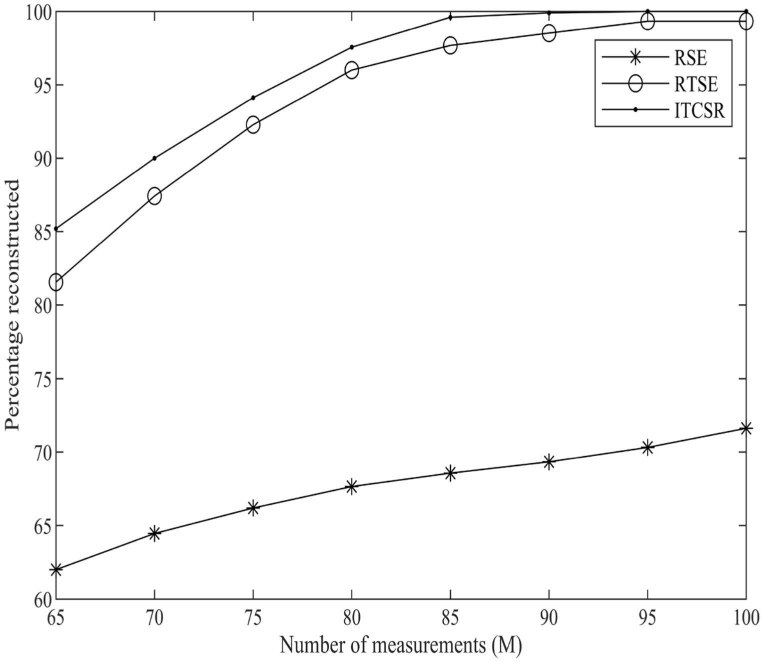


Fig. 4 The estimation percentage reconstructed with the number of measurements (*M*)

estimation method (RTSE) [37] and robust sparsity estimation method (RSE) [38] for comparative analysis. Figure 4 indicates the percentage reconstructed estimation with the number of measurements. In the simulation, we set $K = 12$. Figure 4 illustrates that the percentage reconstructed increased with the increase of M. It is obvious that the percentage reconstructed of ITCSR is larger than percentage reconstructed of RTSE and RSE. In other words, the proposed ITCSR algorithm has higher correct rate than conventional algorithms in our simulation.

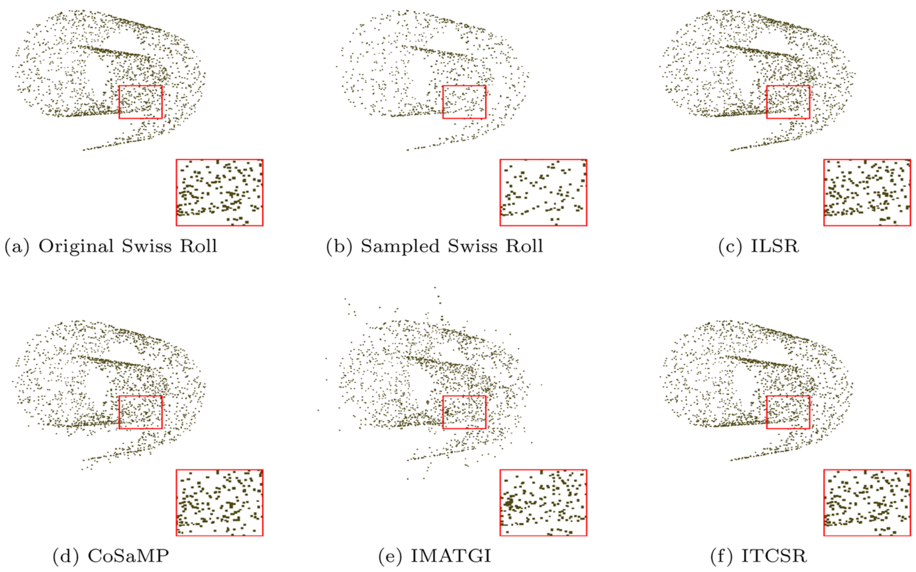
5.1.2 Synthetic data: swiss roll

In this experiment, we use the swiss roll dataset in 3D space to illustrate the reconstruction performance by the proposed ITCSR algorithm. The swiss roll dataset is generated with $N = 2000$ by

$$\begin{cases} x &= \frac{3\pi}{2} \cdot (1+t) \cdot \cos\theta, \\ y &= 21 \cdot t, \\ z &= (1+t) \cdot \sin\theta. \end{cases} \tag{34}$$

Then, we construct an undirected graph by 5-nearest neighboring algorithm. It is worth mentioning that the M4 method is selected.

In Fig. 5, we show the 3D swiss roll dataset visualization of the reconstructed data. Figure 5a, 5b illustrate the original swiss roll and sampled swiss roll. Figure 5c, d, e and f illustrates the visualization reconstruct results under sampling rate is 50% by using ILSR, CoSaMP, IMATGI and proposed ITCSR algorithm for swiss roll dataset, respectively. Overall, we observe that ITCSR outperforms other reconstruction algorithm. Even when compared to the ILSR and IMATGI algorithms, ITCSR still exhibits competitive reconstruction performance. To be specific, compare Fig. 5a and c, the reconstruction performance of the proposed ILSR algorithm has obvious deviation. Compare with Fig. 5a, d


(a) Original Swiss Roll (b) Sampled Swiss Roll (c) ILSR

(d) CoSaMP (e) IMATGI (f) ITCSR

Fig. 5 Comparative subjective performance of swiss roll dataset reconstruction. **a** Original swiss roll, **b** Sampled swiss roll, **c** Reconstructed swiss roll by ILSR, **d** Reconstructed swiss roll by CoSaMP, **e** Reconstructed swiss roll by IMATGI, **f** Reconstructed swiss roll by ITCSR

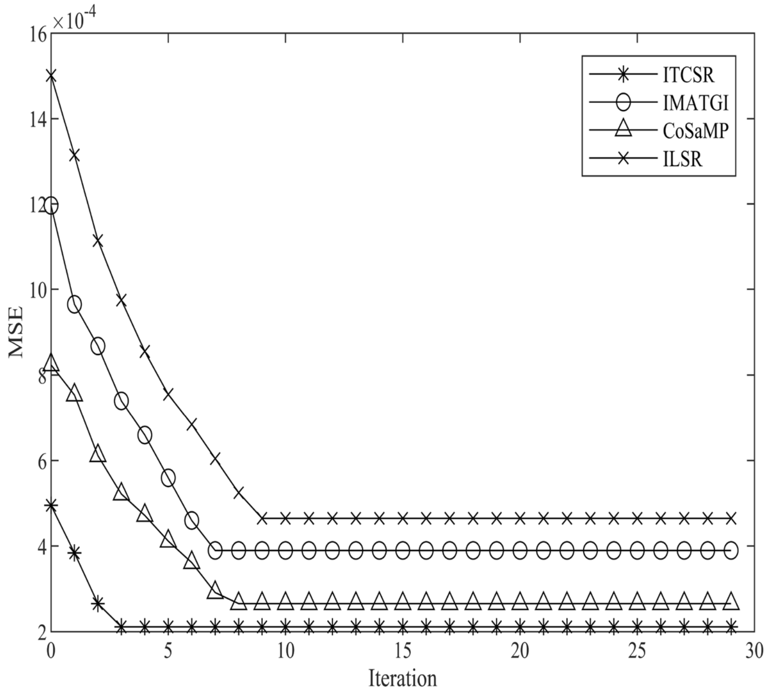


Fig. 6 MSE versus the iteration number on the swiss roll dataset

and e, many obvious outliers exist in the swiss roll data reconstructed by the CoSaMP and IMATGI algorithms. The convergence comparison of the considered methods with 60% sampling rate as shown in Fig. 6. As expected, the proposed ITCSR algorithm exhibits the minimum reconstruction errors and the fastest convergence.

5.2 Experiments on 3D point cloud data

In this subsection, we focus on applying our proposed ITCSR algorithm to six 3D point cloud datasets which are *Block*, *Cactus*, *Fandisk*, *Skull*, *Gargo* and *Dino*, respectively. To simplify the process, the graph is constructed using the 5-nearest neighboring algorithm. The weighted matrix is formed by assigning the edge weight as $\mathbf{W}(i,j) = -\frac{1}{d_{i,j}^2}$, where $d_{i,j}$ represents the distance between vertex i and j. To normalize $\mathbf{W}_{\mathrm{norm}}$, we divide $\mathbf{W}$ by its maximum value, denoted as $\max(\mathbf{W})$. Then, the graph signals are defined as "Location X," "Location Y," and "Location Z."

Our objective is to estimate the missing Location values within an incomplete matrix of relative Location data. We select 50%, 60% and 70% of the measurements by the M4 method and reconstruct all graph signals through the spectrum domain. We calculate the MAXs on six 3D point cloud datasets, and the results are shown in Table 4. In general, we observe that for all reconstruction algorithms, the reconstruction performance improves when the sampling ratio increases. For the different sampling ratios under test, the MAX performance of ITCSR is very competitive. The proposed ITCSR algorithm is far superior to the ILSR algorithm and IMATGI algorithm. Specifically, in Gargo 3d point cloud dataset, the Maxs of the ILSR algorithm, IMATGI algorithm and proposed algorithm in Location Z with 0.5 sampling ratio are 39.3352, 36.4275 and 14.5528,

Table 4 MAX results by different sampling ratio on the 3D point datasets

Dataset	(α, β)	Sampling ratio	0.5			0.6			0.7		
		Location	**X**	**Y**	**Z**	**X**	**Y**	**Z**	**X**	**Y**	**Z**
Block	(0.6789, 0.2109)	ILSR	6.3281	7.2749	6.2093	5.9185	6.1558	6.0641	5.0412	5.9810	5.8650
		CoSaMP	5.9244	4.4361	8.9907	5.1621	3.9762	7.2054	4.3462	3.4314	6.3518
		IMATGI	3.2312	4.8921	5.9865	2.6682	4.1037	5.0082	2.4326	3.8349	4.8821
		ITCSR	**1.7124**	**4.1037**	**0.7793**	**1.5164**	**1.3223**	**0.6216**	**1.2557**	**1.2117**	**0.3742**
Cactus	(0.4432, 0.7890)	ILSR	1.1755	0.8690	1.1415	1.0613	0.7823	1.0015	0.9650	0.6260	0.8995
		CoSaMP	0.5628	0.6165	0.5728	0.5349	0.5183	0.4795	0.4151	0.4572	0.4183
		IMATGI	0.5662	0.6531	0.8929	0.4979	0.5357	0.7795	0.4552	0.4633	0.7117
		ITCSR	**0.2165**	**0.5367**	**0.2101**	**0.1734**	**0.2835**	**0.1742**	**0.1051**	**0.1712**	**0.1287**
Fandisk	(0.6432, 0.1098)	ILSR	2.3195	4.0687	1.2328	2.2825	3.6638	1.0511	1.9648	3.0713	0.7115
		CoSaMP	1.2370	3.5222	1.5906	0.9168	2.4201	1.1078	0.8321	1.2370	0.9368
		IMATGI	0.9545	1.8671	0.4396	0.9074	1.6258	0.4029	0.8829	0.2795	0.3011
		ITCSR	**0.4725**	**1.6258**	**0.3211**	**0.3514**	**0.4453**	**0.2697**	**0.2795**	**0.3011**	**0.2114**
Dino	(0.8143, 0.2575)	ILSR	0.7015	0.8582	0.9647	0.5947	0.7645	0.9046	0.4535	0.6703	0.8623
		CoSaMP	0.2174	0.2845	0.3880	0.1906	0.2362	0.3136	0.1143	0.1562	0.2758
		IMATGI	0.2116	0.3064	0.4209	0.2045	0.2238	0.3368	0.1765	0.2115	0.2586
		ITCSR	**0.1781**	**0.2238**	**0.0775**	**0.1188**	**0.0644**	**0.0672**	**0.0943**	**0.0513**	**0.0437**
Gargo	(0.2551, 0.5060)	ILSR	7.7473	18.1520	39.3352	7.4173	16.1713	36.8923	6.9092	14.1135	35.0320
		CoSaMP	7.3494	**12.2096**	18.6364	6.0548	9.7299	13.5137	5.5081	5.4374	**8.5484**
		IMATGI	6.7810	14.5277	36.4275	5.8464	13.8818	32.2114	5.6599	10.3488	28.6598
		ITCSR	**3.3917**	13.8748	**14.5528**	**3.0248**	**8.5238**	**11.5415**	**2.4435**	**4.6463**	9.8188
Skull	(0.7513, 0.5472)	ILSR	3.2755	7.1945	4.4415	3.0071	6.8923	3.9876	2.8578	6.5630	3.0255
		CoSaMP	2.6399	5.3305	3.7665	2.0048	3.1885	3.1910	1.9540	2.2969	2.5520
		IMATGI	2.4570	4.6874	2.3195	2.2347	4.3642	2.1986	1.9875	3.2935	1.7513
		ITCSR	**1.2749**	**4.3642**	**1.1021**	**1.0127**	**1.1714**	**0.9905**	**0.0867**	**0.0985**	**0.8288**

Table 5 Coherence of matrix $\boldsymbol{\Phi}$ under different sampling methods on the 3D point dataset

Sampling ration	Block	Cactus	Fandisk	Dino	Gargo	Skull
0.5	0.1227	0.2390	0.2146	0.0783	0.1241	0.1113
0.6	0.0813	**0.2121**	0.1341	0.0488	0.0881	0.0990
0.7	**0.0515**	0.2175	**0.0914**	**0.0404**	**0.0476**	**0.0735**

respectively. Then, to verify the convergence of the proposed ITCSR algorithm, we calculate the coherence of the matrix $\boldsymbol{\Phi}$ under different sampling rates with different sampling methods in Table 5.

To illustrate the performance of the proposed ITCSR algorithm, as shown in Figs. 7b, 8b, the *Fandisk* with 6470 vertices and *Skull* 3D point cloud datasets with 20002 vertices are selected and sampling ratio is 50%. Compared with Figs. 7a, 8a, it is observed that the original signals are significantly missing. Overall, we observe that ITCSR improves the reconstruction performance in comparison with the other algorithms. This is because ITCSR utilizes the more general assumption of sparsity rather than the bandlimitedness of the underlying graph signals. Moreover, ITCSR combines the theory of compressed sensing and related concepts of graph signal processing and optimizes the selection of the threshold. To be specific, as indicated in Figs. 7c, 8c, the reconstructed 3D point cloud signal by using ILSR is distorted. Figure 7d, 8d shows the reconstructed 3D point cloud signal by using CoSaMP algorithm. As we can see, the accuracy of reconstructed signals by CoSaMP algorithm is insufficient as the data amount increases. Figures 7e and 8e show the reconstructed 3D point cloud signal by using IMATGI can retain its original basic shape, but there are many outliers. Moreover, it can be seen from Figs. 7f and 8f that with the increase in vertex, the proposed ITCSR algorithm is obviously superior to the ILSR and IMATGI algorithms.

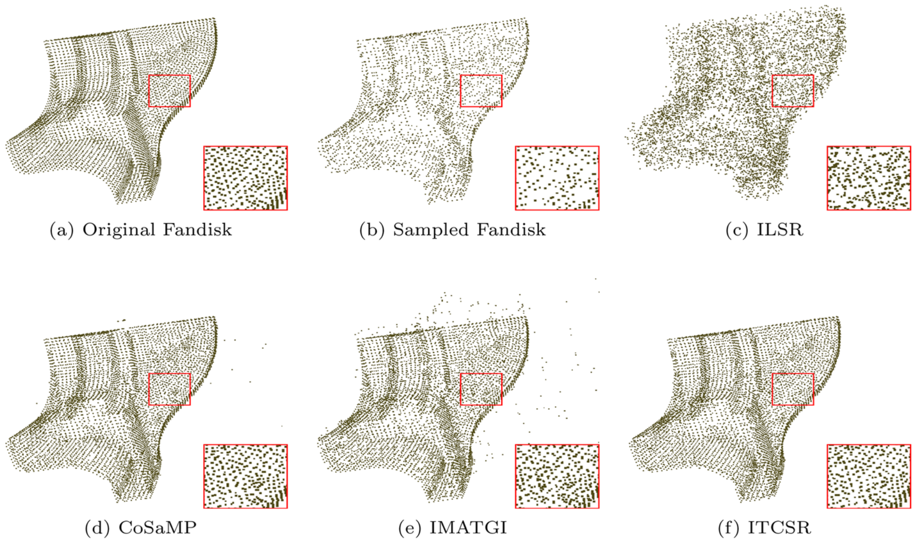

(a) Original Fandisk
(b) Sampled Fandisk
(c) ILSR
(d) CoSaMP
(e) IMATGI
(f) ITCSR

Fig. 7 Comparative subjective performance of Fandisk 3D point cloud model reconstruction. **a** Original Fandisk 3D point cloud, **b** Sampled Fandisk 3D point cloud, **c** Reconstructed Fandisk 3D point cloud by ILSR, **d** Reconstructed Fandisk 3D point cloud by CoSaMP, **e** Reconstructed Fandisk 3D point cloud by IMATGI, **f** Reconstructed Fandisk 3D point cloud by ITCSR

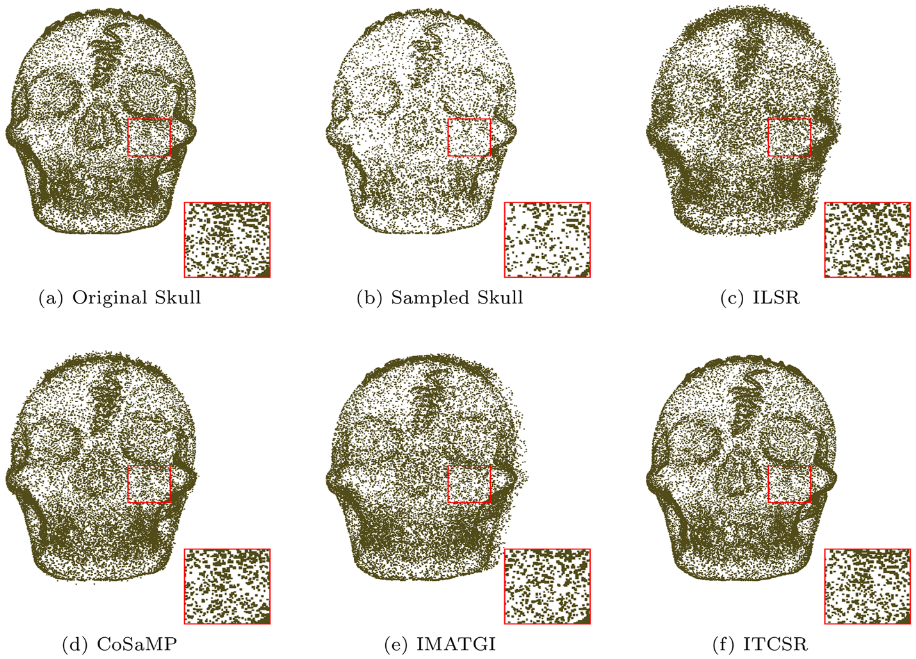


Fig. 8 Comparative subjective performance of Skull 3D point cloud model reconstruction. **a** Original Skull 3D point cloud, **b** Sampled Skull 3D point cloud, **c** Reconstructed Skull 3D point cloud by ILSR, **d** Reconstructed Skull 3D point cloud by CoSaMP, **e** Reconstructed Skull 3D point cloud by IMATGI, **f** Reconstructed Skull 3D point cloud by ITCSR

6 Conclusion

In this paper, we proposed the ITCSR algorithm for sparsity-promoting reconstruction of signals defined on graphs. The ITCSR algorithm considered the suitable threshold to ensure the performance in signal reconstruction and we provided a formal convergence analysis. Compared with the OMP, CoSaMP, ILSR and IMATGI, the proposed ITCSR algorithm decreased the MSE on synthetic datasets under different sampling methods. Experimental results on six public 3D Point Cloud datasets demonstrated that our algorithm performs better, the MAX on *Block*, *Cactus*, *Fandisk*, *Skull*, *Gargo* and *Dino* less than those of the reference algorithms under 0.5, 0.6, 0.7 sampling ratio, respectively.

Appendix A The proof of Theorem 1

Now, we prove Theorem 1 in Sect. . The method we adopted is an extension and enhancement based of prior work [34]. Before presenting the proof, we introduce two lemmas.

Lemma 1 *For a matrix* $\mathbf{\Phi}$ *whose columns are distinct* m *vectors, indexed* $\lambda_1, \cdots, \lambda_m$*, we have a conclusion that the squared singular values of* $\mathbf{\Phi}$ *exceed* $1 - \delta_K$.

Lemma 2 *It can be shown that* $\delta_K \leq K\delta_{K+1}$ *is established for every natural number* K *when matrix* $\mathbf{\Phi}$ *satisfies the RIP with* δ_K.

We omit the easy proof and move on to the demonstration of the theorem.

Proof Let $\boldsymbol{\Gamma}^*$ and $\hat{\mathbf{f}}^*_{\boldsymbol{\Gamma}^*} = \boldsymbol{\Phi}^\dagger_{\boldsymbol{\Gamma}^*}\mathbf{y}$ be under the assumption of Theorem 1. In iteration t, suppose that the algorithm has selected the index set $\boldsymbol{\Gamma}_t \subset \boldsymbol{\Gamma}^*$. Let the index set $H = \boldsymbol{\Gamma}^*$ and $J = \{i, i \notin H\}$. Then, the equation can be defined

$$\frac{\left\|\boldsymbol{\Phi}_J^\top \mathbf{r}_t\right\|_\infty}{\left\|\boldsymbol{\Phi}_H^\top \mathbf{r}^t\right\|_\infty} \le \frac{\left\|\boldsymbol{\Phi}_J^\top\left(\mathbf{y}-(\tilde{\mathbf{y}}^*)^t\right)\right\|_\infty}{\left\|\boldsymbol{\Phi}_H^\top\left((\tilde{\mathbf{y}}^*)^t-\tilde{\mathbf{y}}^t\right)\right\|_\infty} + \frac{\left\|\boldsymbol{\Phi}_J^\top\left((\tilde{\mathbf{y}}^*)^t-\tilde{\mathbf{y}}^t\right)\right\|_\infty}{\left\|\boldsymbol{\Phi}_H^\top\left((\tilde{\mathbf{y}}^*)^t-\tilde{\mathbf{y}}^t\right)\right\|_\infty}. \tag{A1}$$

We may calculate that

$$\begin{aligned}\frac{\left\|\boldsymbol{\Phi}_J^\top\left((\tilde{\mathbf{y}}^*)^t-\tilde{\mathbf{y}}^t\right)\right\|_\infty}{\left\|\boldsymbol{\Phi}_H^\top\left((\tilde{\mathbf{y}}^*)^t-\tilde{\mathbf{y}}^t\right)\right\|_\infty} &\le \max_{i\in J}\left\|\boldsymbol{\Phi}_H^\dagger\boldsymbol{\phi}_{ji}\right\|_1 \\ &\le \sqrt{K}\max_{i\in J}\left\|\left(\boldsymbol{\Phi}_H^\top\boldsymbol{\Phi}_J\right)^\dagger\right\|_2 \cdot \left\|\boldsymbol{\Phi}_H\boldsymbol{\phi}_{ji}\right\|_2 \\ &\le \sqrt{K}\cdot\frac{\delta_{K+1}}{1-\delta_K},\end{aligned} \tag{A2}$$

where the last inequality comes from standard properties of the RIP constant [see Lemma 1 [33]]. Continuing the calculation

$$\begin{aligned}&\frac{\|\ \Phi_J^\top\left(\mathbf{y}-(\tilde{\mathbf{y}}^*)^t\right)\|_\infty}{\|\ \Phi_H^\top\left((\tilde{\mathbf{y}}^*)^t-\tilde{\mathbf{y}}^t\right)\|_\infty} = \frac{\max_{i\in J}\left\|\phi_{ji}^\top\left((\mathbf{y}-(\tilde{\mathbf{y}}^*)^t\right)\right\|_2}{\|\ \Phi_H^\top\left((\tilde{\mathbf{y}}^*)^t-\tilde{\mathbf{y}}^t\right)\|_\infty} \\ &\le \frac{\max_{i\in J}\left\|\phi_{ji}^\top\left(\mathbf{y}-(\tilde{\mathbf{y}}^*)^t\right)\right\|_2}{K^{-1/2}\|\ \Phi_H^\top\left((\tilde{\mathbf{y}}^*)^t-\tilde{\mathbf{y}}^t\right)\|_2} \le \frac{\sqrt{K}\max_{i\in J}\left\|\phi_{ji}\right\|_2\cdot\left\|\mathbf{y}-(\tilde{\mathbf{y}}^*)^t\right\|_2}{\sigma_{\min}(\ \Phi_H)\left\|(\tilde{\mathbf{y}}^*)^t-\tilde{\mathbf{y}}^t\right\|} \\ &\le \frac{\sqrt{K}\left\|\mathbf{y}-(\tilde{\mathbf{y}}^*)^t\right\|_2}{\sqrt{1-\delta_K}\left\|(\tilde{\mathbf{y}}^*)^t-\tilde{\mathbf{y}}^t\right\|_2},\end{aligned} \tag{A3}$$

where the last step refers to **Lemma 2** [34]. In accordance with (A2) and (A3), the proposed ITCSR algorithm therefore selects elements from set H satisfies

$$\frac{\sqrt{K}\left\|\mathbf{y}-(\tilde{\mathbf{y}}^*)^t\right\|_2}{\sqrt{1-\delta_K}\left\|(\tilde{\mathbf{y}}^*)^t-\tilde{\mathbf{y}}^t\right\|_2} + \sqrt{K}\cdot\frac{\delta_{K+1}}{1-\delta_K} \le \alpha(1+\beta), \tag{A4}$$

which (as both terms on the left need to be positive) is only possible if $\delta_{K+1} < \frac{\alpha(1+\beta)}{\sqrt{K}+\alpha(1+\beta)}$. It then yields the following form of

$$\left\|\mathbf{y}-\tilde{\mathbf{y}}^t\right\|_2 \le \sqrt{1+\left(\frac{1}{\alpha(1+\beta)\sqrt{\frac{1-\delta_K}{K}}-\frac{\delta_{K+1}}{\sqrt{1-\delta_K}}}\right)^2}\ \left\|\mathbf{y}-\tilde{\mathbf{y}}_t^*\right\|_2. \tag{A5}$$

□

Acknowledgements

Not applicable

Author Contributions

Yuting Cao was involved in methodology, data curation, software, writing—original draft. Xue-Qin Jiang helped in conceptualization, methodology, supervision, writing—review & editing. Jian Wang contributed to writing—review & editing. Shubo Zhou helped in supervision. Xinxin Hou contributed to writing— review & editing.

Funding

This study was partly supported by the Innovation Program for Quantum Science and Technology (Grant No.2021ZD0300703) and the National Natural Science Foundation project (Grant No. 61971146).

Availability of data and material

Not applicable.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors have no competing interests to declare.

Received: 20 October 2023 Accepted: 4 March 2024
Published online: 15 March 2024

References

1. A. Sandryhaila and J.M.F. Moura, Discrete signal processing on graphs: graph fourier transform, in: 2013 International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 6167-6170 (2013) https://doi.org/10.1109/ICASSP.2013.6638850
2. D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, 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. **30**(3), 83–98 (2013). https://doi.org/10.1109/MSP.2012.2235192
3. A. Ortega, P. Forssard, J. Kovaevi, J.M.F. Moura, P. Vandergheynst, Graph signal processing: overview challenges, and applications. Proc. IEEE Inst. Electr. Electron Eng. **106**(5), 808–828 (2018)
4. P. Xu, J. Zhang, T. Gao, S. Chen, X. Wang, H. Jiang, W. Gao, Real-time fast charging station recommendation for electric vehicles in coupled power-transportation networks: a graph reinforcement learning method. Int. J Elec. Power. **141**, 108030 (2022). https://doi.org/10.1016/j.ijepes.2022.108030
5. J. Feng, F. Chen, H. Chen, Data reconstruction coverage based on graph signal processing for wireless sensor networks. IEEE Wirel. Commun. Lett. **11**(1), 48–52 (2022). https://doi.org/10.1109/LWC.2021.3120276
6. C. Dinesh, G. Cheung and I.V. Baji, 3D point cloud super-resolution via graph total variation on surface normals, in: 2019 IEEE International Conference on Image Processing (ICIP), pp. 4390-4394 (2019) https://doi.org/10.1109/ICIP.2019.8803560
7. X. Shang, R. Ye, H. Feng and Xue-Qin. Jiang, Robust feature graph for point cloud denoising, in: 2022 7th International Conference on Communication, Image and Signal Processing (CCISP), pp. 330-336 (2022) https://doi.org/10.1109/CCISP55629.2022.9974370
8. Q. Huang, R. Li, Z. Jiang, W. Feng, S.Wei, S. Lin, H.Feng and B.Hu, Fast color-guided depth denoising for RGB-D images by graph filtering, in: 2019 53rd Asilomar Conference on Signals, Systems, and Computers(ACSSC), pp. 1811-1815 (2019)https://doi.org/10.1109/IEEECONF44664.2019.9048703
9. P.F. Cid, J.M.B. Ordinas, J.G. Vidal, Graph signal reconstruction techniques for IoT air pollution monitoring platforms. IEEE Internet Things J. **9**(24), 25350–25362 (2022). https://doi.org/10.1109/JIOT.2022.3196154
10. A. Hashemi, R. Shafipour, H. Vikalo, G. Mateos, Towards accelerated greedy sampling and reconstruction of bandlimited graph signals. Signal Process. **195**(10), 0165–1684 (2022). https://doi.org/10.1016/j.sigpro.2022.108505
11. S. Chen, A. Sandryhaila, J.M.F. Moura, J. Kovaevi, Signal recovery on graphs: variation minimization. IEEE Trans. Signal Process. **63**(17), 4609–4624 (2015). https://doi.org/10.1109/TSP.2015.2441042
12. N. Perraudin, P. Vandergheynst, Stationary signal processing on graphs. IEEE Trans. Signal Process. **65**(13), 3462–3477 (2017). https://doi.org/10.1109/TSP.2017.2690388
13. S.K. Narang, A. Gadde, E. Sanou, A. Ortega, Localized iterative methods for interpolation in graph structured data. IEEE Glob. Conf. Signal Imform. Process. (GlobalSIP) **2013**, 491–494 (2013). https://doi.org/10.1109/GlobalSIP.2013.6736922
14. X. Wang, P. Liu, Y. Gu, Iterative reconstruction of graph signal in low-frequency subspace. IEEE Glob. Conf. Signal Inform. Process. (GlobalSIP) **2014**, 448–452 (2014). https://doi.org/10.1109/GlobalSIP.2014.7032157
15. Y. Bai, F. Wang, G. Cheung, Y. Nakatsukasa, W. Gao, Fast graph sampling set selection using Gershgorin disc alignment. IEEE Trans. Signal Process. **68**, 2419–2434 (2020). https://doi.org/10.1109/TSP.2020.2981202
16. D.L. Donoho, Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
17. J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inf. Theory **53**(12), 4655–4666 (2007). https://doi.org/10.1109/TIT.2007.909108
18. T. Blumensath, M. Davies, Iterative hard thresholding for compressed sensing. Appl. Comput. Harmon. Anal. **27**(3), 265–274 (2009). https://doi.org/10.1016/j.acha.2009.04.002

19. M.B. Mashhadi, M. Fallah, F. Marvasti, Interpolation of sparse graph signals by sequential adaptive thresholds, in. Int. Conf. Sampl. Theory Appl. (SampTA) **2017**, 266–270 (2017). https://doi.org/10.1109/SAMPTA.2017.8024339
20. M. Brajović, I. Stanković, M. Daković and L. Stanković, Reconstruction of sparse graph signals from reduced sets of samples, in 2023 27th International Conference on Information Technology (ICIT), pp. 1-5 (2023) https://doi.org/10.1109/IT57431.2023.10078603
21. C.C. Tseng and L. Su-Ling, A missing data recovery method of sparse graph signal in GFT domain, in: 2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), pp. 1-2, https://doi.org/10.1109/ICCE-China.2018.8448787 (2018)
22. D. Needell, J.A. Tropp, CoSaMP: iterative signal recovery from incomplete and inaccurate samples. Appl. Comput. Harmon. Anal. **26**(3), 301–321 (2009). https://doi.org/10.1016/j.acha.2008.07.002
23. E. Candes and T. Tao, Decoding by linear programming, IEEE Trans. Inf. Theory, (2005), https://doi.org/10.48550/arXiv.math/0502327
24. W. Dai, O. Milenkovic, Subspace pursuit for compressive sensing signal reconstruction. IEEE Trans. Inf. Theory (2009). https://doi.org/10.1109/TIT.2009.2016006
25. J. Wang, S. Kwon, and B. Shim, Generalized orthogonal matching pursuit, IEEE Trans. Signal Process., 60(12), pp. 6202–6216 (2012)https://doi.org/10.1109/TSP.2012.2218810
26. J. Xue, B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm. Syst. Sci. Control. Eng. **8**(1), 22–34 (2020). https://doi.org/10.1080/21642583.2019.1708830
27. E. Candes, T. Tao, Near-optimal signal recovery from random projections: universal encoding strategies? IEEE Trans. Inf. Theory **52**(12), 5406–5425 (2007). https://doi.org/10.1109/TIT.2006.885507
28. S. Datta, S. Howard, D. Cochran, Geometry of the Welch bounds. Linear Algebra Appl. **437**(10), 2455–2470 (2012). https://doi.org/10.1016/j.laa.2012.05.036
29. N. Beresford, A result complementary to Gersgorin? s circle theorem. Linear Algebra Appl. **431**(1), 20–27 (2009). https://doi.org/10.1016/j.laa.2009.01.030
30. B. Girault, A. Ortega and S. S. Narayayan, Graph vertex sampling with arbitrary graph signal hilbert spaces, in: 2020 International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5670-5674 (2020) https://doi.org/10.1109/ICASSP40776.2020.9054723
31. D. E. Olivier Tzamarias, P. Akyazi and P. Frossard, A novel method for sampling bandlimited graph signals, in 2018 26th European Signal Processing Conference (EUSIPCO), pp. 126-130 (2018) https://doi.org/10.23919/EUSIPCO.2018.8553064
32. H. Shomorony and A. S. Avestimehr, Sampling large data on graphs, in: IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 933-936 (2014) https://doi.org/10.1109/GlobalSIP.2014.7032257
33. D. Donoho, M. Elad, Maximal sparsity representation via l-1 minimization. Comput. Sci. **100**(12), 2197–2202 (2002). https://api.semanticscholar.org/CorpusID:17259152
34. J. Tropp, Greed is good: algorithmic results for sparse approximation. IEEE Trans. Inf. Theory **50**(10), 2231–2242 (2004). https://doi.org/10.1109/TIT.2004.834793
35. B. Girault, A. Ortega, and S.S. Narayayan, Graph vertex sampling with arbitrary graph signal hilbert spaces, in: 2014 International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 5670-5674, https://doi.org/10.48550/arXiv.2002.11238 (2014)
36. H. Shomorony and A. S. Avestimehr, Sampling large data on graphs, in: 2014 IEEE Global Conference on Signal and Imformation Processing (GlobalSIP), pp. 933-936 (2014) https://doi.org/10.48550/arXiv.1411.3017
37. X. Xu, J. Chen, N. Wan, D. Chen and J. Wan, Sparsity estimation method in compressed data gathering of wireless sensor networks, in: 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), pp. 833-836 (2019) https://doi.org/10.1109/ITAIC.2019.8785897
38. Qin, S., Yin, J, A Robust sparsity estimation method in compressed sensing, in: China Conference on Wireless Sensor Networks (CWSN), pp. 481-488 (2014) https://doi.org/10.1007/978-3-662-46981-1-46
39. 3D Point Datasets, http://graphics.stanford.edu/data/3Dscanrep/ (2014)

Publisher's Note