Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 154 (2019) 406–415

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Congress of Information and Communication Technology, ICICT 2019

Decentralized Distributed Compressed Sensing Algorithm for Wireless Sensor Networks

Xu Dong Wen[a,*], Chun Wu Liu[b]

[a] *College of Artificial Intelligence, National University of Defense, Changsha 410072, China*
[b] *College of Artificial Intelligence, National University of Defense, Changsha 410072, China*

Abstract

Distributed Compressed Sensing (DCS) is an effective method to reduce data transmission and energy consumption in wireless sensor networks (WSN). To further enhance the data compression ratio in clustered WSNs, a new data collection method which named Decentralized Distributed Compressed Sensing (DDCS) is proposed in this paper. This method can be divided into three parts: cluster head election, data compression, and joint reconstruction. In the cluster head elections, sparsity is introduced as a reference factor. In data compression, the cluster head data is used as the common data of sensors in the same cluster. After remove the common data, member sensors in the same cluster only compress and transmits their specific data. In joint reconstruction, using residual correlation to reconstruct the signal through the OMP algorithm. Simulation results show that the performance of DDCS is better than traditional DCS. It can greatly reduce the amount of data transmission, the cluster head energy consumption, and the network delay at the cluster head.

Keywords: *Wireless Sensor Network, Distributed Compressed Sensing, Decentralized Distributed Compressed Sensing, Cluster, JSM-1*

* Corresponding author. Tel.: +(86)17608461995;
*E-mail address:*1599395598@qq.com

1877-0509
10.1016/j.procs.2019.06.058

1. Introduction

Due to the limit of energy and the difficulty in maintenance of the sensor nodes, energy consumption control is always the hot research topic of the WSN[1]. Generally, there is great correlation in time and spatial of sensor data in the same area. Therefore, while ensuring that the detection information is not lost, reducing the amount of data transmission is an effective method to reduce communication energy consumption in WSN[2].

A few scholars have done some researches about the combination of DCS and WSN. Cheng proposes the concept of Hierarchical Data Compressed Sensing (HDCS), which studies the compression method of correlation data from two hierarchical, within cluster and between clusters3. Some also study the fusion of DCS and clustering algorithm (LEACH, DEEN)4. Yang proposes Regionalized Compressive Sensing (RCS) whose purpose is to improve the practical efficiency of DCS applications5. In RCS, the direct transmission and DCS transmission are simultaneously applied in the wireless sensor network.

In general, there are two specific application methods of DCS in WSN. In the first method, the cluster head collects all the original data of member sensor and then the data is compressed and encoded in the cluster head. The sink node receives the data and performs joint reconstruction. There are two problems with this method. First, the number of data packets that need to be processed in the cluster head is large, which means the burden of entire network is uneven. Second, there will be data aggregation at the cluster head, which resulting in a long delay of the network data at the cluster head and poor real-time network performance. In the second method, the member sensors compress the raw data independently and send it to the cluster head. The main problem of this approach is that the data correlation is not fully utilized at the encoder side, a lot of redundant data has been transmitted in cluster.

To solve the above problems, this paper proposes Decentralized Distributed Compressed Sensing (DDCS) for WSN. The algorithm is based on the JSM-1 model. It can be divided into three parts: cluster head election, data compression, and joint reconstruction. In the cluster head elections, sparsity is introduced as a reference factor. In data compression, the cluster head data is used as the common data of sensors in the same cluster. After remove the common data, member sensors in the same cluster only compress and transmits their unique data. In joint reconstruction, using residual correlation to reconstruct the signal through the OMP algorithm. Simulation results show that the performance of DDCS is better than traditional DCS. It can greatly reduce the amount of data transmission, reduces the cluster head energy consumption, and reduces the network delay at the cluster head.

2. Compressed Sensing and Distributed Compressed Sensing

2.1. Compressed sensing

Compressed sensing is a new signal processing theory proposed by Donoho in 2006[6]. The basic idea of it is to combine compression and sampling. It mainly includes sparse representation of signals, observation matrix design, and signal reconstruction[7]. Assuming a finite one-dimensional discrete signal $x \in R^N$ having a sparse representation under basis $\Psi_{N\times N}$ .That is, x can be expressed as

$$x = \Psi\Theta = \sum_{i=1}^{N} \theta_i \varphi_i, \|\Theta\|_0 = K \ll \tag{1}$$

In formula, K represents the number of non-zero elements in the vector Θ , which is called sparsity. Look for an observation matrix $\Phi : M \times N\left(M < N\right)$ that is not related to the sparse basis Ψ , and make it an inner product with the original signal x to obtain the observation vector y . The signal sampling process is as follows

$$y = \Phi x = \Phi\Psi\Theta \tag{2}$$

Let sensing matrix $A = \Phi\Psi$. The above equation can be expressed as $y = A\Theta$. In order to ensure the unique reconstruction of the signal, the Restricted Isometry Property (RIP) condition needs to be satisfied[8]. After above process, the dimension of the data decreases from N to M so that data compression is achieved.

In the decoding, the best way to recover the original signal from its measurement is by solving an optimization problem trying to minimize the l_0 norm of the signal in the sparse domain. That is:

$$\overset{\wedge}{\Theta} = \arg\min_{\Theta} \|\Theta\|_0 , s.t. y = \Phi\Psi\Theta \tag{3}$$

However, this problem is computationally intractable due to its NP-hard complexity, so it is common to consider a relaxed from using l_1 norm[9], which can be solved by means of linear programming techniques:

$$\overset{\wedge}{\Theta} = \arg\min_{\Theta} \|\Theta\|_1 , s.t. y = \Phi\Psi\Theta \tag{4}$$

Signal reconstruction is an important part of compressed sensing. Among many reconstruction algorithms, greedy matching pursuit algorithms has been widely used due to the complexity of it is low[10]. The main algorithms include Matching Pursuit Algorithm (MP), Orthogonal Matching Pursuit Algorithm (OMP), Subspace Tracking Algorithm (SP).

2.2. Distributed compressed sensing

The DCS theory, which developed from CS was proposed by Dror Baron[11]. Its core idea is to improve reconstruction efficiency and accuracy by exploiting the temporal and spatial correlation between data. There are mainly four sparse models in DCS, which are JSM-1, JSM-2, JSM-3 and mixed support set model.

Among them, the JSM-1 model is suitable for describing the situation where all sensors are affected by the entire environment and individual sensors are affected by the partial environment[12]. Thus, this paper focus on the JSM-1. In the JSM-1, the original signal is represented as the sum of the sparse unique part and common parts. Besides, both the unique part and the common part are sparsely represented on the same sparse basis[13].

$$X_{\mathrm{j}} = Z_c + Z_j , j \in \{1, 2, ..., J\} \tag{5}$$

In the formula, $Z_c = \Psi \cdot \Theta_c , \|\Theta_c\|_0 = k_c , Z_j = \Psi \cdot \Theta_j , \|\Theta_j\|_0 = k_j$. X_{j} represents all original signals, Z_c represents the signal common sparse portion, and Z_j represents the signal-unique sparse portion.

3. DDCS Model Building

3.1. Cluster head election

Assume there are L clusters in a monitoring area, where the i-th cluster is represented by C_i . There are $J_i + 1$ nodes in cluster C_i , where the j -th node is represented by $S_{i,j}$, $j = 1, 2, ... J_i + 1$. Set the data collected in unit time by $S_{i,j}$ can be represented by the N-dimensional column vector $x_{i,j}$, and $x_{i,j}$ can be sparsely represented by an N-dimensional vector $\theta_{i,j}$ through sparse basis $\Psi^{N\times N}$ [13].

$$x_{i,j} = \Psi^{N\times N} \theta_{i,j} , \|\theta_{i,j}\| = k_{i,j} \tag{6}$$

Where $k_{i,j}$ is the sparsity of $\theta_{i,j}$. When electing the cluster head, each node in the same cluster broadcasts its own signal sparseness $k_{i,j}$ and residual energy $E_{i,j}$. Assume that the remaining energy of the node p is the largest, denoted as $E_{i,p}$. Then we can set the threshold L to determine the cluster head candidate set U

$$U = \left\{ S_{i,j} \middle| E_{i,p} - E_{i,j} \le L, j \in \{1, 2, ... J_i + 1\} \right\} \tag{7}$$

The above formula means that nodes whose residual energy difference with node p is smaller than L are all in the cluster head candidate set U, and the smaller the L value, the less elements in U. In the set U, we select the node with the smallest sparsity $k_{i,j}$ as the cluster head, denoted as H_i. Then, rename the remaining nodes with $S_{i,j}$, $j = 1,2,..J_i$. So far, the cluster head election has completed.

This cluster head election method comprehensively considers the node residual energy and signal sparsity. Selecting the remaining energy as the first reference element is because the cluster head needs to process and transfer all data packets in the cluster, which would consume a large amount of energy. Selecting a point with large remaining energy can avoid rapid “death” of the cluster head. Simultaneously, we also take the signal sparsity as a reference element. On the one hand, it is convenient for the implementation of the DDCS method. On the other hand, we can avoid the edge node is elected as cluster head to a certain extent. Because the edge environment is complex, sensors in edge are affected by many factors so that their signal sparsity is relatively large.

3.2. Intra-Cluster data compression

After the cluster head election is completed, the cluster head is represented by H_i. Assuming there are J_i member nodes in the cluster C_i, and the j-th node is represented by $S_{i,j}$, $j = 1,2,...J_i$. Let the data collected in unit time by $S_{i,j}$ is represented by N-dimensional column vector $x_{i,j}$. In particular, h_0 represents the data collected by the cluster head sensor. According to the structural characteristics of the clustered WSN, there is a time correlation between the internal elements $x_{i,j}$, and there is also a spatial correlation between the data $\left\{x_{i,j}, h_0 : j = 1,2,...,J_i\right\}$ collected by nodes in the same cluster C_i.

The data compression process of the DDCS algorithm is:

The cluster head H_i makes the data collected in unit time be sparse through sparse basis Ψ.

$$h_0 = \Psi\theta_0, \|\theta_0\|_0 = k_0 \tag{8}$$

Where k_0 represents the sparsity of the sparse coefficient θ_0. Then, the cluster head H_i broadcasts vector θ_0 to all members in cluster C_i. According to the JSM-1 sparse model, the monitoring vector $x_{i,j}$ is modeled as the sum of common components and unique components. In particularly, we have θ_0 as a common component. That is

$$x_{i,j} = \Psi\theta_{i,j} = \Psi\theta_0 + \Psi\theta^z_{i,j}, \left\|\theta^z_{i,j}\right\| = k^z_{i,j} \tag{9}$$

Where $\theta^z_{i,j}$ denotes the sparse coefficient of the unique component and $k^z_{i,j}$ denotes the sparsity. The member node in the same cluster calculate $\theta^z_{i,j}$ after receiving θ_0.

$$\theta^z_{i,j} = \theta_{i,j} - \theta_0 \tag{10}$$

Find out maximum sparsity $k_{\max}$

$$k_{\max} = \max\left\{k^z_{i,j}, j \in \{1,2...J_i\}\right\} \tag{11}$$

The cluster head H_i selects the observation matrix $\Phi_0^{M\times N}$ to sample its own signal h_0 to obtain the observation vector y_0

$$y_0 = \Phi_0^{M\times N} h_0 \tag{12}$$

$$y_0 = \Phi_0^{M\times N}\Psi\theta_0 = A_0\theta_0 \tag{13}$$

In the above formula, M<N, and sensing matrix A_0 needs to satisfy the RIP condition. Here, we select the Gaussian random matrix because of simplicity. Simultaneously with cluster head compression, member node $S_{i,j}$ also performs independent compression observation on its own unique component $\theta^z_{i,j}$

$$y_{i,j} = A^{M\times N}\theta^z_{i,j} \tag{14}$$

The number of observations M is calculated according to the following formula

$$M \ge cK_{\max}\log(N/K_{\max}) \tag{15}$$

Similarly, the sensing matrix $A^{M\times N}$ also chooses Gaussian random matrix. It can be proved that when the number of measurements of the random Gaussian measurement matrix M satisfies the above formula, the RIP condition is met with a great probability. $y_{i,j}$ indicates the specific components of the data of the member node $S_{i,j}$. After obtaining $y_{i,j}$, each member node sends it to the cluster head in turn through Time Division Multiplexing (TDMA).

3.3. Joint reconstruction of data

After the cluster head receives the compressed data $y_{i,j}$ of all member sensors in the cluster, it bundles it with its own compressed data y_0 and sends them to the sink node. In order to analyze and process the data, the sink node needs to reconstruct the received compressed data. From the perspective of the sensing matrix, there are mainly two types of data sent from the cluster C_i. One is the monitoring data y_0 of the cluster head H_i itself, its sensing matrix is A_0, and the other is the specific monitoring component $y_{i,j}$ of the sensor within the cluster, whose sensing matrix is $A^{M_c\times N}$.

Then, consider the joint reconstruction algorithm of DDCS:

1) Input: $A_0, A^{M_c\times N}, y_0, y_{i,j}, k^z_{i,j}\ j\in\{1,2...J_i\}$
2) Calculation: $y^d_{i,j} = y_{i,j} - y_{i,1}, j\in\{2,3...J_i\}$

Restore $\theta^d_{i,j}$ from $y^d_{i,j}$ by OMP algorithm[14]

Initialization

Let support set $I=\Phi$, number of iterations $t=1$, residual $r_1 = y^d_{i,j}$

Identification

Find the index λ_t of the closest matching atom r_t and the closest atom in the perceptron matrix $A^{M_c\times N}$.

$$\lambda_t \leftarrow \arg\max_j \left\{ \left| \left\langle r_t, a_j \right\rangle \right| \right\} \tag{16}$$

Updated

Add λ_t to the support set, $I \leftarrow I \cup \{\lambda_T\}$, and update the column set $A_t = [A_{k-1} \quad a_{\lambda_t}]$ in the perceptual matrix accordingly.

Rebuild target signal

$$\hat{\theta}^d_{i,j} \leftarrow A_t^+ y_{i,j} \tag{17}$$

Where A_t^+ represents the pseudo-inverse of A_t, $A_t^+ = \left(A_t^* A_t\right)^{-1}\cdot A_t^*$

Update residual components

$$r_{t+1} \leftarrow r_t - A_t \hat{\theta}^d_{i,j} \tag{18}$$

Increase $t = t+1$, if $t < k^z_{i,j} + 1 = T$, go back to step 1. T is the maximum number of iterations.

Calculation

Use the OMP algorithm to recover $\hat{\theta}_0$ from y_0 and $\hat{\theta}_{i,1}$ from $y_{i,1}$. The steps are similar to (3). Then calculate the estimated value of sparse signal for each sensor

$$\hat{\theta}_{i,j} = \hat{\theta}^d_{i,j} + \theta_{i,1} + \theta_0, j \in \{2,3...J_i\} \tag{19}$$

Recover the original signal

$$\hat{x}_{i,j} = \Psi \hat{\theta}_{i,j}, j \in \{1,2,3...J_i\} \tag{20}$$

$$\hat{h}_0 = \Psi \hat{\theta}_0 \tag{21}$$

The core idea of this joint reconstruction algorithm is to use the residual correlation between $y_{i,j}$. Use $y_{i,1}$ as edge information to jointly reconstruction, which greatly reduces the amount of reconstruction computation.

4. Results

This section uses MATLAB simulation to verify DDCS performance and make comparation with the traditional DCS. In the experiment, it is assumed that there are 10 sensor nodes in the cluster, and the length of monitoring data $x_{i,j}$ is N=256. According to the JSM-1 model, 10 signals are generated using a random method. Their sparsity $k_{i,j}$ is randomly selected from 15 to 35, where the common component sparsity $k_0 = 8$. To simplify the analysis, the initial remaining energy of each sensor is the same. According to DDCS, the node with the smallest signal sparsity is directly selected as the cluster head. Select the Gaussian random matrix as the sensing matrix as well.

4.1. Feasibility of DDCS

According to the DDCS method, the 10 randomly generated signals are reconstructed and randomly choose 5 signals for performance testing. Figure 1 is obtained. The blue circle in the figure represents the original signal, and the red dot represents the reconstruction signal. We can see that DDCS can achieve complete signal reconstruction.

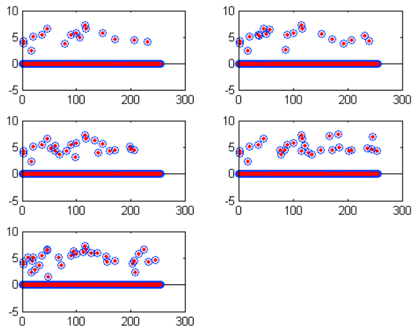

Figure. 1 Five signal compression and reconstruction performance

4.2. Performance of compressing

Figure 2 shows the relationship between the common sparsity k_0 and the number of measurements M. It shows that for the DDCS method, when the maximum sparsity $k_{\max}$ of 10 signals is equal to 28 and average sparsity k_{ave} is a fixed value, the M will decrease with increase of the k_0. In addition, when k_0 is a constant, the smaller k_{ave} is, the smaller M is. On the contrary, M does not vary as k_0 increase when the $k_{\max}$ is a fixed value for the DCS method.

Besides, by comparing the blue line and green line, we can find that when the maximum sparsity $k_{\max}$ and average sparsity k_{ave} is same, there is a turning point k_c. When the common part sparsity k_0 is less than k_c, DDCS performance is inferior to DCS. But when k_0 is greater than k_c, DDCS can greatly reduce the number of measurements and have a good performance in reducing compression ratio.

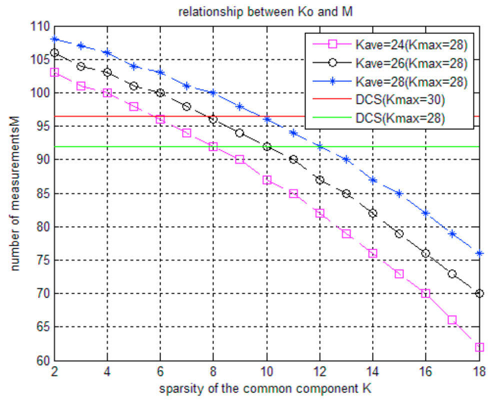

Figure. 2 relationship between sparsity of common component k_0 and the number of measurements M

4.3. Comparison of reconstruction performance

Select the reconstruction error MSE as an indicator of the reconstruction effect of the algorithm

$$MSE_i = \left\| \hat{x}_i - x_i \right\|_2^2 \Big/ \left\| x_i \right\|_2^2 \tag{22}$$

$$MSE = \sum_{i=1}^{10} MSE_i \Big/ 10 \tag{23}$$

Where x_i represents the original signal and $\hat{x}_i$ represents the reconstructed signal. MSE_i represents reconstruction error of one signal and MSE is the average of 10 signals. Expand the signal length N to 1024. Performing 20 times joint reconstructions of the signal using DCS and DDCS respectively. By calculating the reconstruction error, figure 3 is obtained. As can be seen from Fig. 3, the reconstruction error of DDCS is smaller than DCS.

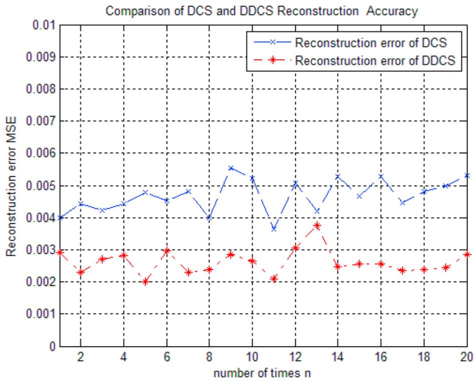

Figure. 3 Comparison of DCS and DDCS reconstruction accuracy

5. Discussion

To better understand the difference between DDCS method and DCS method. The above results will be analyzed and discussed in this section.

5.1. Calculating of point k_c

In traditional DCS method, according to the JSM-1 model, cluster head data θ_0 and cluster member data $\theta_{i,j}$ can be written as

$$\begin{cases} \theta_0 = \theta_{com} + \theta_0^z \\ \theta_{i,j} = \theta_{com} + \theta_j \end{cases} \tag{24}$$

In the formula above, θ_{com} is the common sparse component. θ_0^z and θ_j are the unique sparse component. Then maximum sparsity $k_{\max}$ can be written as

$$k_{i,j} = k_{com} + k_j \tag{25}$$

$$k_{\max} = \max\left\{k_{i,j}, j \in \{1,2...J\}\right\} \tag{26}$$

According to formula (3.10), number of measurements M is determined by $k_{\max}$ when N is fixed. Be different from DCS, in DDCS method, we subtracted θ_0 from $\theta_{i,j}$ and get

$$\theta_{i,j}^z = \theta_{i,j} - \theta_0 = \theta_j - \theta_0^z \tag{27}$$

$$k_{i,j}^z = k_j + k_0^z \tag{28}$$

$$k_{\max}^z = \max\left\{k_{i,j}^z, j \in \{1,2...J\}\right\} \tag{29}$$

Making subtraction between $k_{i,j}^z$ and $k_{i,j}$

$$\Delta k = k_{i,j}^{z} - k_{i,j} = k_0^z - k_{com} \tag{30}$$

Because of $k_{head} = k_{com} + k_0^z$, Δk can be written as

$$\Delta k = k_{head} - 2\bullet \tag{31}$$

Let $\Delta k = 0$, by calculating the above equation we can get the value of turning point k_c

$$k_c = k_{com} = k_{head}/2 \tag{32}$$

Analyzing according above deduce, when $k_0 > k_c = k_{head}/2$, we can get $k_{i,j}^{z} < k_{i,j}$, which means the sparsity of transmission signal is reduced. Further, $k_{\max}$ and number of measurements M also are reduced. This suggests that DDCS has a better performance compared with DCS when common components make up the main part of the data. The theoretical analysis agrees with the experimental results.

On the whole, in many applications of WSN, $k_0 > k_{head}/2$ can be easily satisfied. For example, monitoring the temperature of a volcano, monitoring the light intensity of farmland. This indicates that DDCS method can be widely used in WSN.

5.2. Improvement of reconstruction Accuracy

Through the experimental results in section 4.3, we find that the reconstruction accuracy of DDCS method is improved compared with traditional DCS. There are two main reasons to explain this result.

On the one hand, according to the analysis in section 5.1, the amount of data that needs to be compressed has been greatly reduced in DDCS. As a result, the computation and iteration times of the reconstruction algorithm are also greatly reduced compared with DCS. This means that the reconstruction error decreases to some extent.

On the other hand, in DDCS method, we make subtraction with θ_0 for all $\theta_{i,j}$ and get $\theta_{i,j}^{z}$ according to the formula (5.4). The correlation between the data increases because the same number is subtracted. When the signal is reconstructed jointly, taking full advantage of this correlation can also reduce the reconstruction error.

6. Conclusion

In order to fully reduce the energy consumption of wireless sensor networks and improve its lifetime, this paper proposes DDCS by improving on the basis of traditional DCS. DDCS was introduced from three aspects: cluster head election, data compression, and joint reconstruction. The simulation results show that when the correlation between the sensor data is strong, compared to DCS, DDCS has a large data compression ratio and the compression effect is better. At the same time, since the DDCS utilizes the data correlation both at the encoding and decoding, the reconstruction error of the data is also significantly reduced, and the reconstruction performance is relatively good.

Due to the limitation of experimental conditions, this paper mainly uses computer simulation to verify DDCS method and performance analysis. Further study is needed about the implementation and performance of DDCS on actual WSN. Besides, the energy consumption of DDCS method in cluster is not analyzed in this paper, which can be further discussed in the future.

References

1. B. Song, Y. Wang, H. Zhang, A Strategy to Regulate WSN Nodes' Energy Consumption Based on Emission Rate, in, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 267-274.
2. H. da S. Araújo, W.L.T. de Castro, R.H. Filho, Reduction of Energy Consumption in WSN using an Approach Geocast, in, Springer Netherlands, Dordrecht, 2010, pp. 567-572.
3. Y. Cheng, J. Si*, X. Hou, Hierarchical Distributed Compressed Sensing for Wireless Sensor Network, J Electr Inf Technol, Qinhuangdao, 2017.

4. F. lv, J. Zhang, H. Shi, Q. Wang, Y. Wang, Distributed Compressive Sensing in WSN, Chinese Journal of Sensors and Actuators, Dalian, 2013.
5. H. Yang, X. Wang, Data Gathering Based on Regionalized Compressive Sensing in WSN, CHINESE JOURNAL OF COMPUTERS, Chicago, 2017.
6. M. Rostami, Compressed Sensing, in: Compressed Sensing with Side Information on the Feasible Region, Springer International Publishing, Heidelberg, 2013, pp. 9-22.
7. Compressed Sensing, in: K. Ikeuchi (Ed.) Computer Vision: A Reference Guide, Springer US, Boston, MA, 2014, pp. 132-132.
8. W. Dan, Analysis of orthogonal multi-matching pursuit under restricted isometry property, Publisher, City, 2014.
9. Z. Chen, X. Hou, C. Gong, X. Qian, Compressive sensing reconstruction for compressible signal based on projection replacement, Publisher, City, 2016.
10. X. Bi, X. Chen, X. Li, L. Leng, Energy-based adaptive matching pursuit algorithm for binary sparse signal reconstruction in compressed sensing, Publisher, City, 2014.
11. Y. Ji, Z. Yang, A distributed compressed sensing approach for speech signal denoising, Publisher, City, 2011.
12. W. Xu, J. Lin, K. Niu, Z. He, A joint recovery algorithm for distributed compressed sensing, Publisher, City, 2012.
13. X. Hou, The Joint Reconstruction Algorithmsresearch Of Distributed Compressedsensing, in, YANSHAN UNIVERSITY, 2016.
14. P. Cui, L. Ni, Joint reconstruction algorithm for distributed compressed sensing, Infrared and Laser Engineering, Hefei, 2015.5