An Immune Clone Particle Swarm Optimization Algorithm for Sparse Representation of Hyperspectral Images

Li Wang(✉), Wei Wang, and Boni Liu

Xi'an Aeronautical University, Xi'an 710077, Shaanxi, China

Abstract. The sparse representation of hyperspectral images can reduce the amount of data and facilitate image classification and interpretation processing. An immune clone particle swarm optimization algorithm (ICPSO) to achieve sparse representation of hyperspectral images is proposed in this paper. The main idea of the algorithm is to use the evolutionary process of particle swarm to simulate the atomic matching process of the orthogonal chasing algorithm to improve the diversity and efficiency of atom selection. Further, according to the clonal selection theory of biological immunology, immune cloning, cloning mutation and cloning selection operators are used to expand the local search range, fully maintain the diversity of the population, improve the convergence speed and avoid the premature convergence. Sparse representation experiments are carried out on hyperspectral images using proposed ICPSO to verify the performance of the algorithm. Compared with the orthogonal matching pursuit algorithm, the proposed algorithm can improve the reconstruction accuracy as well as the computing efficiency.

Keywords: Hyperspectral image · Immune clone · Particle swarm algorithm · Reconstruction · Sparse representation

1 Introduction

Containing abundant spatial geometric information and spectral feature information, hyperspectral images (HSIs) [1] are suitable for terrain reconnaissance [2], target detection and recognition [3, 4], classification [5, 6]. The development of hyperspectral imaging spectrometer [7–9] and the research of hyperspectral image interpretation [10–12] have become the focus of attention. The expanding application fields require hyperspectral images to provide more detailed target information. The higher the spatial and spectral resolution is, the higher the challenge to the system's data storage and transmission capabilities.

Effective sparse representation could capture the main features of the signal, which leads to achieve a small amount of data for signal description. The basic idea of sparse representation is that, from the basic functions set of the signal projection, only a small number of the basic functions could be extracted to represent the original signal with little

X. Wang et al. (Eds.): AICON 2021, LNICST 396, pp. 593–611, 2021.
https://doi.org/10.1007/978-3-030-90196-7_51

distortion [13, 14]. Sparse representation model requires that in the signal expansion, the coefficients of most of the basic functions are zero, and only a few basic functions have large non-zero coefficients. Its applications include image classification [15], signal de-noising [16], object detection [17], and face recognition [18]. Applying the signal sparse decomposition theory to hyperspectral images to obtain sparse representations, can greatly reduce the amount of data and achieve further classification and interpretation of hyperspectral images.

A typical way to obtain such signal sparse representation is the orthogonal matching pursuit (OMP) algorithm [19, 20]. OMP, an iterative greedy algorithm, chooses the best atom from the dictionary to match the signal, thereby achieving high reconstruction accuracy. Hyperspectral image features are complex, and redundant dictionaries can be used to obtain good sparse representations, but the computation time of OMP is unbearable under the existing computing conditions. Instead, the evolutionary algorithms [21, 22] could be used to achieve the sparse representation.

This paper proposes an immune clone particle swarm algorithm to achieve sparse representation of hyperspectral images to overcome the problem of low calculation efficiency of OMP algorithm. The main idea is: utilize particle swarm optimization (Particle Swarm Optimization, PSO) to simulate the atomic matching process of OMP, which can improve the search process of the optimal atom; further, introduce the immune clone algorithm to increase the diversity of the population and increase the convergence rate, and ultimately improve the efficiency of sparse representation. Four classic hyperspectral images are used to test the effectiveness of the proposed algorithm.

The remainder of this paper is organized as follows. In Sect. 2, the sparse decomposition process using OMP and the basic PSO algorithm are described. And then in Sect. 3, the principle of the proposed Immune Clone Particle Swarm Algorithm (ICPSO) are presented, included four aspects, namely initial Particle production, Immune cloning operator, cloning mutation operator, cloning selection operator, particle updating operator. Moreover, the implementation process of proposed algorithm is also introduced in this section. Next in Sect. 4, experimental results are shown and analyzed. Finally, in Sect. 5, some concluding remarks are made.

2 Principles of OMP and PSO

2.1 Sparse Representation

The signal sparse representation model based on redundant dictionary is expressed as:

$$\min\|\boldsymbol{\theta}\|_0 \quad s.t. \quad \boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{\theta} \tag{1}$$

Where, $\boldsymbol{x} \in R^N$ is the original signal, $\boldsymbol{\Phi} \in R^{N\times M}$ is the redundant dictionary, and $\boldsymbol{\theta} \in R^M$ is the sparse coefficient vector. Each column in $\boldsymbol{\Phi}$ is called as an atom. In sparse representation, there are two issues needed to be solved: 1) construct the redundant dictionary $\boldsymbol{\Phi}$, and 2) solve the optimal function in Eq. (1) to obtain the sparse coefficient vector $\boldsymbol{\theta}$. In the proposed algorithm, Gabor dictionary [23, 24] is used to be the sparse basis for hyperspectral images, and its generating function is expressed as,

$$\boldsymbol{g}_\gamma(n) = \frac{1}{\sqrt{s}} e^{-\pi\left(\frac{n-u}{s}\right)^2} \cos(\upsilon n + \varphi) \tag{2}$$

Where, $n = 0, 1, ..., N$, $\gamma = (s, u, \upsilon, \varphi)$ is the time-frequency parameter vector. The variation range is: $1 \leq s \leq N$, $1 \leq u \leq N$, $0 \leq \upsilon \leq 2\pi$, $0 \leq \varphi \leq 2\pi$. After the discretization of the time-frequency parameters, the number of atoms in the Gabor dictionary is $M = 52\left(N \log_2 N + N - 1\right)$.

2.2 OMP Algorithm

Solving signal sparse representation under a redundant dictionary is a NP-hard problem. There does not exist known polynomial time algorithm to solve this optimization problem, a sub-optimal approximation method is needed. The basic idea of OMP for sparse representation is to find the optimal atoms that can linearly represent the original signal in the redundant dictionary through continuous iteration. When the residual between the original signal and the represented signal is continuously reduced, an approximate linear representation of the original signal can be obtained.

Specifically, through continuous iteration, OMP selects some optimal atoms from the dictionary $\boldsymbol{\Phi}$, these atoms make up the set of the true signal. The implementation process of OMP is summarized as follows:

Step 1: Initializition: iterative number $k = 1$, residual $\boldsymbol{r}_0 = \boldsymbol{x}$, index collection of optimal atoms $\Lambda_0 = []$.
Step 2: Select the index of optimal atom: The selected atom needs to meet the principle of maximum relevance, therefore, we need to traversal all the atoms in dictionary $\boldsymbol{\Phi}$, compute the relevance and sort them,

$$\lambda_k = \arg\max_l |\langle \boldsymbol{r}_{k-1}, \boldsymbol{\Phi}_l \rangle| \tag{3}$$

Where, $\boldsymbol{\Phi}_l$ is l-th column of $\boldsymbol{\Phi}$.
Step 3: Update index collection: Add the optimal atom index to the collection, $\Lambda_k = \Lambda_{k-1} \cup \lambda_k$.
Step 4: Update residual: Use the searched best atoms to linearly represent the signal and calculate the residual,

$$\boldsymbol{r}_k = \boldsymbol{x} - \boldsymbol{\Phi}_{\Lambda_k} \left(\boldsymbol{\Phi}_{\Lambda_k}^{\mathrm{T}} \boldsymbol{\Phi}_{\Lambda_k} \right)^{-1} \boldsymbol{\Phi}_{\Lambda_k}^{\mathrm{T}} \boldsymbol{x} \tag{4}$$

Where, $\boldsymbol{\Phi}_{\Lambda_k}$ represents the sub-matrix constructed by the atoms indexed by Λ_k.
Step 5: Determine whether the maximum iteration number K is satisfied, if not then $k = k + 1$, repeat Step 2–Step 4, else stop iteration.

In other words, the maximum iteration number in OMP is the number of optimal atoms to represent the original signal. The original signal $\boldsymbol{x}$ could be represented using the K optimal atoms indexed by Λ_K. The sparse coefficient vector is,

$$\hat{\boldsymbol{\theta}} = \left(\boldsymbol{\Phi}_{\Lambda_K}^{\mathrm{T}} \boldsymbol{\Phi}_{\Lambda_K} \right)^{-1} \boldsymbol{\Phi}_{\Lambda_K}^{\mathrm{T}} \boldsymbol{x} \tag{5}$$

Where, $\boldsymbol{\Phi}_{\Lambda_K}$ represents the sub-matrix consisting of column vectors in $\boldsymbol{\Phi}$ indexed by Λ_K.

Use these optimal atoms to linearly represent the signal and the reconstructed signal is expressed as:

$$\hat{\boldsymbol{x}} = \boldsymbol{\Phi}_{\Lambda_K} \hat{\boldsymbol{\theta}} \tag{6}$$

2.3 Particle Swarm Optimization

The particle swarm optimization algorithm treats the potential solution of the optimization problem as a particle in the search space without mass [25, 26]. The particle has a certain flying speed and position. Assume that the dimensional of search space is L, the population $\boldsymbol{A} = \left(\boldsymbol{A}_1, \boldsymbol{A}_2, ..., \boldsymbol{A}_i, ..., \boldsymbol{A}_Q\right)$ contains Q particles, where the i th particle represents its position in the search space and is represented as a vector $\boldsymbol{A}_i = (a_{i1}, a_{i2}, ..., a_{il}, ..., a_{iL})$, $i = 1, 2, ..., Q$, $l = 1, 2, ..., L$. The velocity of the i th particle is $\boldsymbol{V}_i = (v_{i1}, v_{i2}, ..., v_{il}, ..., v_{iL})$, the individual extremum is $\boldsymbol{P}_{besti} = (P_{i1}, P_{i2}, ..., P_{il}, ..., P_{iL})$ and the population extremum is $\boldsymbol{G}_{best} = (G_1, G_2, ..., G_l, ..., G_L)$.

Particles update their speed and position by tracking individual extremes and group extremes,

$$v_{il}^t = w v_{il}^{t-1} + c_1 r_1 \left(P_{il}^{t-1} - a_{il}^{t-1}\right) + c_2 r_2 \left(G_l^{t-1} - a_{il}^{t-1}\right) \tag{7}$$

$$a_{il}^t = a_{il}^{t-1} + v_{il}^t \tag{8}$$

Where, t is the current evolution generation, w is the inertia weights, c_1 and c_2 are non-negative constants, r_1 and r_2 are random numbers distributed in the interval [0, 1].

Particle swarm algorithm can use individual experience information and population experience information to adjust its state. The advantage of the algorithm is the fast convergence, but the disadvantage is that the particles easily fall into local extremes and cannot be rid of them, which limits the search range of the particles and reduces the search efficiency. Clonal selection [27, 28] is an important theory of biological immune system theory. Compared with the evolutionary algorithm, the clone selection algorithm can increase the speed of convergence whereas expanding the local search range, and maintain the diversity of the population through cloning, mutation, selection, etc., thereby improving the performance of the particle swarm algorithm. For the reason that, combining immune clone with particle swarm algorithm, an immune clone particle swarm optimization (ICPSO) algorithm for sparse representation is proposed in this paper.

3 Proposed ICPSO Algorithm for Sparse Representation

In this section, the details about the proposed ICPSO are presented. First of all, the optimization problem and initial particle production in the proposed ICPSO are described. And then, the four operators in the proposed algorithm, including the immune cloning operator, cloning mutation operator, cloning selection operator and particle updating operator are designed. Finally, the implementation process is summarized.

3.1 Optimization Problem and Initial Particle Production

The main framework of the proposed algorithm is based on the OMP algorithm, that is, the sparse representation is completed in an iterative way. In its iteration, the proposed ICPSO use the particle evolution to find one optimal atom, then updates the optimal atom set, updates the residuals, and finally uses the found optimal atoms to sparsely approximate the original signal.

The core idea of ICPSO is using particle evolution to simulate the atom matching process of OMP. Particularly, ICPSO does not need generate dictionary in advance, whereas particles expressed by position, velocity and fitness value are used to represent the atoms. According to the generation principle of Gabor atoms, the search space dimension of our optimization problem is $L = 4$. The initial population is denoted as $\boldsymbol{A} = \left(\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_Q\right)$, where Q is the population size. The initial position of the i th particle is $\boldsymbol{a}_i = \left(a_{i1}, a_{i2}, a_{i3}, a_{i4}\right)$, and each component is corresponding to the time-frequency parameter vector in Gabor dictionary.

The principle of maximum correlation is used to find the optimal atoms; therefore, the fitness value for i th particle can be described as,

$$f(\boldsymbol{a}_i) = \left|\left\langle \boldsymbol{r}_{k-1}, \boldsymbol{g}_{\boldsymbol{a}_i} \right\rangle\right| \tag{9}$$

$$\boldsymbol{g}_{\boldsymbol{a}_i}(n) = \frac{1}{\sqrt{a_{i1}}} e^{-\pi\left(\frac{n-a_{i2}}{a_{i1}}\right)^2} \cos(a_{i3}n + a_{i4}) \tag{10}$$

Where, $\boldsymbol{g}_{\boldsymbol{a}_i}$ is the corresponding atom generated by particle $\boldsymbol{a}_i$.

The optimization problem of the algorithm ICPSO can be summarized as,

$$\max\{f(\boldsymbol{a}): \boldsymbol{a} \in (s, u, \upsilon, \varphi)\} \tag{11}$$

If the initial population is randomly generated, the individual cannot be representative, which is not conducive to searching for the global best. We expect that the initial individuals can be dispersed as evenly as possible over the entire feasible solution space, so that the algorithm can search uniformly over the entire feasible solution space. Here we use chaotic logic mapping to generate a set of particles as the initial population $\boldsymbol{A}^0 = \left(\boldsymbol{a}_1^0, \boldsymbol{a}_2^0, ..., \boldsymbol{a}_Q^0\right)$. Set the optimal position of the i th particle as $\boldsymbol{P}_{besti} = \boldsymbol{a}_i^0$, and the extreme value of the initial population to $\boldsymbol{G}_{best} = \max_{\boldsymbol{a}_i^0} f\left(\boldsymbol{a}_i^0\right)$, $i = 1, 2, ..., Q$. The initial velocity of the particles is $\boldsymbol{v}_i^0 = \left(v_{i1}^0, v_{i2}^0, v_{i3}^0, v_{i4}^0\right)$. The positions and velocities are updated by the following four operators.

3.2 ICPSO Operators

Immune Cloning Operator. For the current population $\boldsymbol{A}(t-1)$, we perform immune cloning operations on it, that is,

$$\vartheta(\boldsymbol{A}(t-1)) = \left(\vartheta(\boldsymbol{a}_1); \vartheta(\boldsymbol{a}_2); ...; \vartheta\left(\boldsymbol{a}_Q\right)\right) \tag{12}$$

Where, $\vartheta(\boldsymbol{a}_i)$ is the clone of $\boldsymbol{a}_i$. The clone number is defined as,

$$C_i = C\frac{f(\boldsymbol{a}_i)}{\sum_{i=1}^{Q} f(\boldsymbol{a}_i)}, i = 1, 2, ..., Q \tag{13}$$

Where, C is the clone constant.

After Immune cloning operator, the population turns to,

$$\boldsymbol{A}'(t-1) = \left(\boldsymbol{A}; \boldsymbol{A}_1'; \boldsymbol{A}_2'; \ldots; \boldsymbol{A}_Q'\right) \tag{14}$$

Where, $\boldsymbol{A}_i' = \left(\boldsymbol{a}_{i1}; \boldsymbol{a}_{i2}; ...; \boldsymbol{a}_{iC_i-1}\right), \boldsymbol{a}_{iq} = \boldsymbol{a}_i, q = 1, 2, ..., C_i - 1.$

Cloning Mutation Operator. In order to promote the exchange of useful information between individuals and improve the uniformity of the individual solution of the offspring, a cloning mutation operator is used to generate representative offspring individuals. During this process, in order to retain the original information of the population, the mutation operator is only applied to clone population $\vartheta(\boldsymbol{a}_i)$. Specify mutation probability P_{m}, for each component of the particle, generates a random number p_m between 0 and 1. If $p_m < P_{\mathrm{m}}$, then the corresponding component would be re-selected as one number from its variant range, otherwise the component would not change. Mutated individuals are indicated by $\boldsymbol{d}_i$ and the population is denoted as $\boldsymbol{D}(t-1)$.

Cloning Selection Operator. The clone selection operator selects excellent individuals from the progeny of the cloned mutations of the particles, thereby forming a new population. Calculate the fitness of the current population, and calculate the fitness of all the offspring that have been cloned and mutated.

If there exists the outstanding mutation particle which satisfies the following two conditions,

$$\boldsymbol{b} = \max\left\{f\left(\boldsymbol{a}_{iq}\right) \middle| q = 1, 2, ..., C_i - 1\right\} \tag{15}$$

$$f(\boldsymbol{a}_i) < f(\boldsymbol{b}) \quad \boldsymbol{a}_i \in \boldsymbol{A}(t-1) \tag{16}$$

Then particle $\boldsymbol{b}$ is chosen to replace the parent particle $\boldsymbol{a}_i$ to update the population.

Particle Updating Operator. Define the speed and position update operators are,

$$\boldsymbol{v}_i(t) = w\boldsymbol{v}_i(t-1) + c_1 r_1\left(\boldsymbol{P}_{besti}(t-1) - \boldsymbol{a}_i(t-1)\right) + c_2 r_2\left(\boldsymbol{G}_{\mathrm{best}}(t-1) - \boldsymbol{a}_i(t-1)\right) \tag{17}$$

$$\boldsymbol{a}_i(t) = \boldsymbol{a}_i(t-1) + \boldsymbol{v}_i(t) \tag{18}$$

The inertia weight w adopts a linear differential decreasing strategy [29], expressed as,

$$w(t) = w_{\max} - \frac{(w_{\max} - w_{\min})}{T^2}t \tag{19}$$

Where, T is the maximum evolution algebra. In the early stage of algorithm evolution, the decreasing trend of w is slow, and the global search ability is strong, which is benefit for finding good particles. In the later stage of algorithm evolution, the decreasing trend of w is accelerated, which can improve the convergence speed of the algorithm. After the above operators, the tth population would be evolved to the $t+1$th generation population.

3.3 Implementation Process of ICPSO

Combining the clonal selection theory of the immune system with the particle swarm optimization algorithm of advanced evolution theory, an immune clone particle swarm optimization algorithm is proposed to solve the sparse representation problem. The block diagram of the proposed algorithm is depicted in Fig. 1, and the implementation process is summarized as follows.

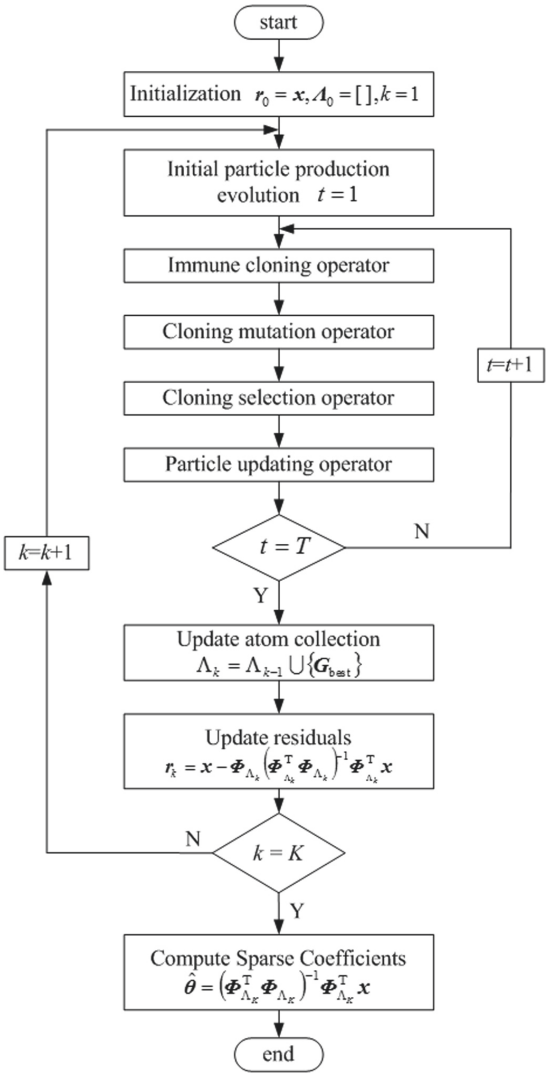

Fig. 1. Block diagram of proposed algorithm.

Step 1: Algorithm initialization, iterative number $k = 1$, residual $\boldsymbol{r}_0 = \boldsymbol{x}$, optimal atom collection $\Lambda_0 = []$.
Step 2: Initial particle production: Initialize the particle population to $\boldsymbol{A}^0$, calculate the fitness of the particles. Determine $\boldsymbol{P}_{besti}$ and $\boldsymbol{G}_{best}$, and set the evolution algebra to $t = 1$.
Step 3: Utilizing immune cloning operator to obtain a cloned population $\boldsymbol{A}^{'}(t-1)$ using (12)–(14).
Step 4: Utilizing cloning mutation operator to obtain the mutated population $\boldsymbol{D}(t-1)$.
Step 5: Utilizing cloning selection operator to obtain $\overline{\boldsymbol{A}}(t-1)$ using (15)–(16).
Step 6: According to particle updating operator, update speed and position of particles to obtain $\boldsymbol{A}(t)$, calculate the new fitness function, and updating $\boldsymbol{P}_{besti}$ and $\boldsymbol{G}_{best}$.
Step 7: Check whether the maximum evolution algebra T is reached. If not then let $t = t + 1$ and repeat Step 3–Step 6, else output $\boldsymbol{G}_{\text{best}}$.
Step 8: Update atom collection with $\boldsymbol{G}_{\text{best}}$, $\boldsymbol{\Lambda}_k = \boldsymbol{\Lambda}_{k-1} \cup \boldsymbol{G}_{\text{best}}$.
Step 9: Update residuals using (4), here $\boldsymbol{\Phi}_{\Lambda_k}$ represents the dictionary consisting of atoms in Λ_k.
Step 10: Check whether the maximum iteration number K is satisfied, if then stop iteration. Else let $k = k + 1$, repeat Step 2–Step 9.

Sparsely represent the original signal using the optimal atom obtained by ICPSO and the reconstructed signal could be expressed using (5) and (6).

4 Experimental Results and Analysis

In order to evaluate the performance of the proposed algorithm, some experimental results and analysis on four hyperspectral images are carried out in this section. At the beginning, utilize OMP algorithm for sparse representation to determine the number of optimal atoms. Next, the parameters, including the population size and maximum evolution algebra, number of optimal atoms, for proposed algorithm ICPSO are discussed. And then, the comparison between the proposed algorithm ICPSO and standard OMP algorithm is shown, using peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [30] and runtime to assess the performance. The hardware and software environments for these experiments are: AMD quad-core CPU, 3.80 GHz, 16G memory and Matlab2012b.

4.1 Hyperspectral Datasets and Evaluation Metrics

Four hyperspectral images were selected to evaluate the performance of the proposed sparse representation algorithm. The first two datasets are Cuprite1 and Cuprite2 collected by AVIRIS with a total of 224 bands. The number of available bands is 188, excluded the abnormal bands and all zero bands. The third dataset is Indian Pines collected by AVIRIS with a total of 220 bands. After removing the water absorption band, the number of available bands is 200. The fourth dataset is Pavia University collected by ROSIS with a total of 115 bands. Removing the noisy bands, the number of available bands is 103.

Based on the above analysis, the computational complexity of OMP algorithm has high relevance with the atoms in redundant dictionary, and the number of atoms is greatly

increasing with the length of signal. Taking into account the computation efficiency, the hyperspectral images are processing block by block. Given consideration to accuracy and efficiency, the block size is set as 16 in the following experiments. Without loss of generality, the Indian Pines is spatially cropped to 128 * 128 and the other datasets are cropped to 256 * 256. The hyperspectral image is represented by a two-dimensional matrix $\boldsymbol{X} \in R^{N_p \times N_\lambda}$, where, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_b, ..., \boldsymbol{x}_{N_\lambda}]$, $\boldsymbol{x}_b$ is the vectorized representation of the b th band image, N_p is the pixels of a single band image, and N_λ represents the band number. The original image of the 50th band is shown in Fig. 2.

Two metrics are adopted to evaluate the reconstruction accuracy of the proposed algorithm, PSNR and SSIM between the reconstructed image and the original image. The PSNR measured in dB is defined as,

$$\mathrm{PSNR}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = 20 \log_{10} \frac{\max(\boldsymbol{x})}{\sqrt{\mathrm{MSE}(\boldsymbol{x}, \hat{\boldsymbol{x}})}} \tag{20}$$

Where, $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are the original and reconstructed image, $\max(\boldsymbol{x})$ is the peak value of $\boldsymbol{x}$, $\mathrm{MSE}(\boldsymbol{x}, \hat{\boldsymbol{x}})$ is the mean squared error,

$$\mathrm{MSE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{N} \left\| \boldsymbol{x} - \hat{\boldsymbol{x}} \right\|_2^2 \tag{21}$$

The SSIM between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ is defined as,

$$\mathrm{SSIM}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{(2\mu_1\mu_2 + C_1)(2\sigma_{12} + C_2)}{(\mu_1^2 + \mu_2^2 + C_1)(\sigma_1^2 + \sigma_2^2 + C_2)} \tag{22}$$

Where, μ_1 and μ_2 are the mean values of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, σ_1 and σ_2 are the standard deviation values of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, σ_{12} represents the correlation coefficient between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, C_1 and C_2 are constants related to the dynamic range of the pixel values. The details for these parameters can refer to [30].

4.2 Parameter Selection for OMP

Analysis from the implementation process of OMP, one atom is increased into the optimal atom collection every iteration; that is, the number of iterations is the number of optimal atoms. The accuracy of the reconstructed signal is decided by the optimal atoms. There is no doubt that the number of optimal atoms K would have significant effect on performance of OMP. In this part, experiments on the 50th band images using OMP with different K are presented. The atom number is varied from 1 to 100 at the interval of one. The reconstructed images are represented by the optimal K atoms. The PSNR and relative change between two iterations are shown in Fig. 3.

It is not surprising that, with the increase of the atom number, the reconstructed PSNR is increasing as well. We noticed that the PSNR is not keeping increasing with the atom number. When the atom number increases to 50, PSNR would tend to be smooth and has little oscillation. The relative change between two adjacent iterations, shown in Fig. 3(b), drops quickly when the iteration starts. With the iteration progresses, the relative change descends slowly and almost tends to be stable after 50 iterations. Therefore, we can draw such a conclusion that the number of optimal atoms of OMP algorithm could be set as $K = 50$ with good reconstruction performance.

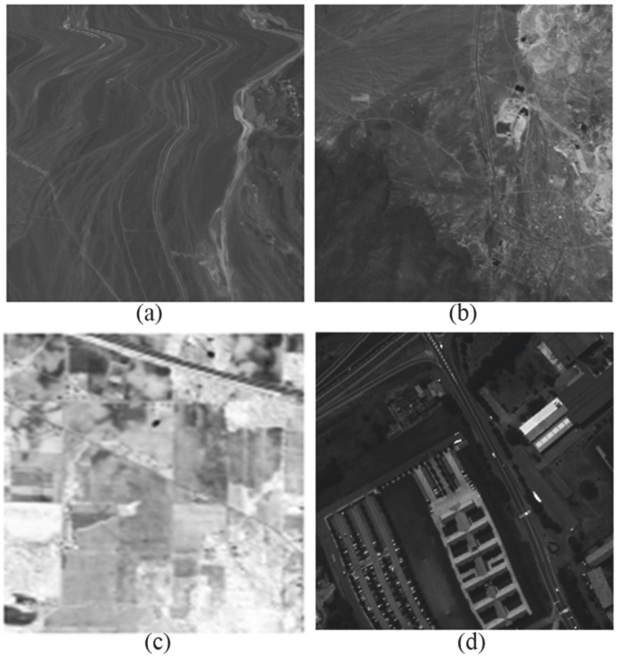

Fig. 2. Original 50th images of four datasets. (a) Cuprite1; (b) Cuprite2; (c) Indian Pines; (d) Pavia University.

4.3 Parameter Selection for ICPSO

In the implementation process of proposed ICPSO algorithm, the maximum evolution algebra T, the population size Q and the optimal atom number K would have significance on the performance. In this part, experiments on the 50th band images using ICPSO with different parameters are presented. The maximum evolution algebra T and the population size Q are varied from 5 to 50 at the interval of 5. The optimal atom number K is varied from 10 to 100 at the interval of 10. In addition, the clone constant C and mutation probability P_{m} are set as $C = 15$ and $P_{\mathrm{m}} = 0.2$. The reconstructed images are represented by the optimal K atoms. Considering the randomness of ICPSO algorithm, the simulation would run 10 times under the same parameter to obtain the average value. The PSNR changes with different parameters for Cuprite1 and Indian Pines are shown in Fig. 4 and Fig. 5, respectively.

When the optimal atom number K is fixed at 100, the PSNR changes with maximum evolution algebra T and population size Q is shown in Fig. 4(a). Seen from the figure, when the maximum evolution algebra T increases, the variation of PSNR is not obviously, whereas the variation of PSNR with increasing of population size Q is more significant. When the population size reaches 30, the PSNR tends stable. Through this comparison, we can say that population size has more importance on the PSNR than that of maximum evolution algebra.

When the population size Q is fixed at 30, the PSNR changes with the maximum evolution algebra T and optimal atom number K is shown in Fig. 4(b). With the increase

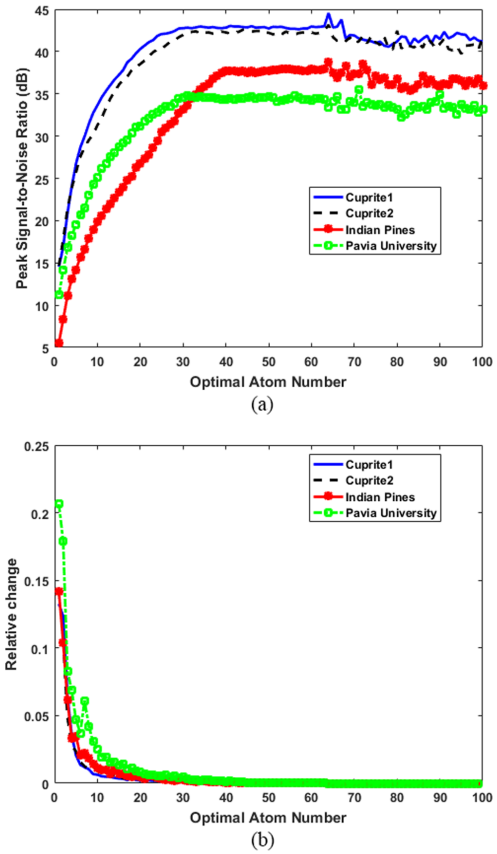

Fig. 3. The PSNR and relative change with different atom number for four band images using OMP. (a) PSNR; (b) Relative change.

of optimal atom number, the reconstructed PSNR would gradually be improved. However, the maximum evolution algebra has little effect on the PSNR. The result can further confirm the truth that the maximum evolution algebra has little effect of on reconstructed PSNR.

To further analyze the impact of maximum evolution algebra on PSNR, the results of maximum evolution algebra varied from 1 to 5 is given in Fig. 4(c). The optimal atom number is 100 and the PSNR is changed with population size. Seen from these curves, the effect of evolution algebra on PSNR is really very weak. Particularly, when the population size is greater than 30, several curves are cross together. This could further verify that we can set the parameter T to the lowest level without loss of reconstruction performance.

The analysis of Fig. 5 is similar with that of Fig. 4, therefore, we can draw a conclusion that the optimal atom number has the largest influence on the reconstructed PSNR, followed by population size, and the maximum evolution algebra has the smallest. We can also find that when population size is greater than 30, the PSNR has little increase.

Even if maximum evolution algebra is as low as 1, the proposed ICPSO could achieve good performance when population size is greater than 30. The other two hyperspectral images have the similar results and are able to reach the same conclusion. Based on these experimental results, we set the maximum evolution algebra $T = 1$ and $Q = 30$ in the proposed ICPSO algorithm.

After the population size and maximum evolution algebra are determined, experiments are carried out on four hyperspectral images to determine the optimal atoms number in proposed ICPSO. Results on the 50th band images using proposed ICPSO with different optimal atom number K are presented. The optimal atom number is varied from 10 to 100 at the interval of 10. The reconstructed images are represented by the optimal K atoms. The SSIM and relative change between two iterations are shown in Fig. 6. With the increase of the atom number, the reconstructed SSIM is increasing as well, whereas the relative change decreases. Due to the uncertainty of evolution algorithm, the SSIM and relative change are not a smooth process. However, with the help of the three immune cloning operators, the proposed ICPSO still converges quickly and finally tends stable. Thence, the optimal atom number in ICPSO is set as $K = 100$ in the following experiments.

4.4 Comparison Between OMP and ICPSO

In this subsection, the comparison between standard OMP algorithm and the proposed ICPSO algorithm is presented. The four hyperspectral datasets are sparse represented using the optimal atoms searched by OMP and ICPSO, respectively. The reconstructed PSNR, SSIM and computation time are used to evaluate the performance. The optimal atom number in OMP is $K = 50$. The parameters in ICPSO are $Q = 30$, $T = 1$ and $K = 100$.

The PSNR, SSIM and runtime (the three parameters are averaged on all bands) are shown in Table 1. Seen from the table, the PSNR of proposed ICPSO is about 1–2 dB higher than that of OMP. Moreover, the SSIM of two algorithms could both reach 0.97 or higher, which demonstrates that the optimal atoms searched by the two algorithms could describe the structure of the images very well. This fully states that by searching optimal atoms using the biological evolution manner, the proposed ICPSO could achieve better reconstruction accuracy than that of OMP. The immune cloning operators, which maintain the diversity of population, allow the algorithm converging to a good optimal solution.

The computation efficiency of these two algorithms is analyzed. The runtime shown in Table 1 is the average computation time consumed by one band. The OMP algorithm needs to traversal all the atoms in the redundant dictionary to find one optimal atom. In our experiments, the atoms in the Gabor dictionary are 119756 when the signal length is 256. Therefore, the OMP algorithm needs to complete 5987800 inner product operators to find 50 atoms. Whereas in the proposed ICPSO algorithm, in its iteration, it needs to complete immune cloning, cloning mutation, cloning selection and updating operator of 30 particles. Although the proposed ICPSO requires 100 iterations, the evolution process of ICPSO is still faster than that of OMP.

The comparison between the reconstructed image and the original image using OMP and proposed ICPSO are shown in Fig. 7 and Fig. 8. The 50th band results of Cuprite2 and

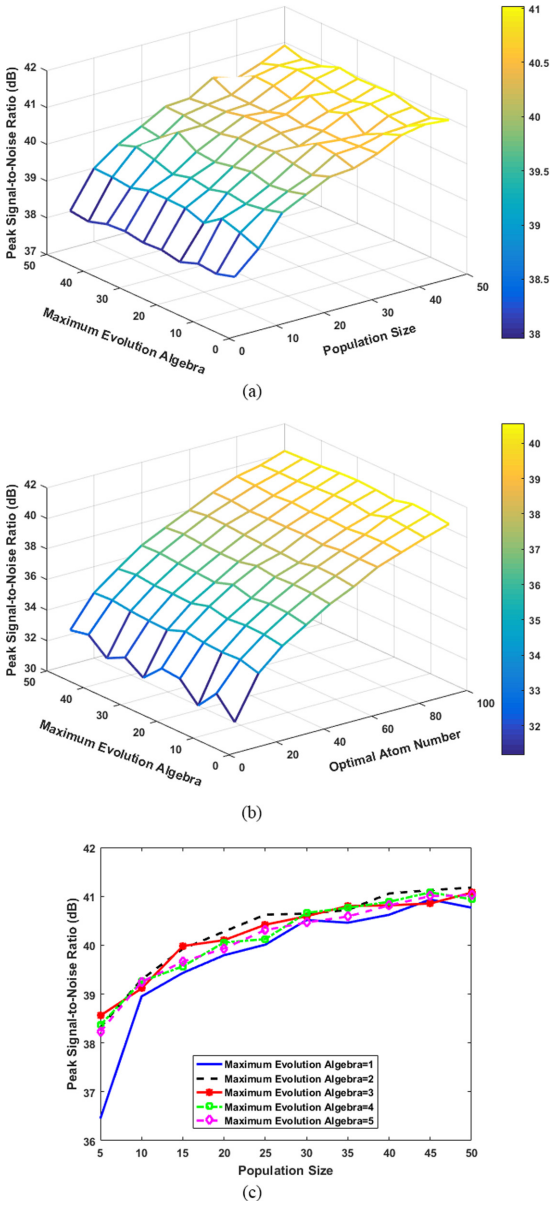

Fig. 4. The PSNR changes with different parameters for Cuprite1. (a) PSNR changes with evolution algebra and population size with optimal atom number fixed at 100; (b) PSNR changes with evolution algebra and optimal atom number with population size fixed at 30; (c) PSNR changes with population size, when optimal atom number is fixed at 100.

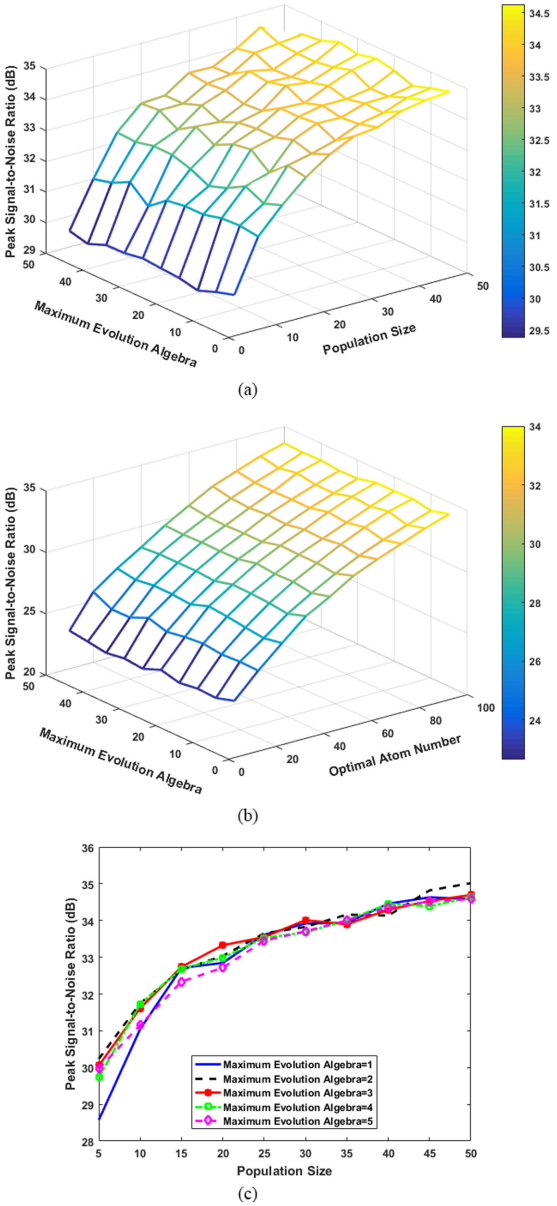

Fig. 5. The PSNR changes with different parameters for Indian Pines. (a) PSNR changes with evolution algebra and population size with optimal atom number fixed at 100; (b) PSNR changes with evolution algebra and optimal atom number with population size fixed at 30; (c) PSNR changes with population size, when optimal atom number is fixed at 100.

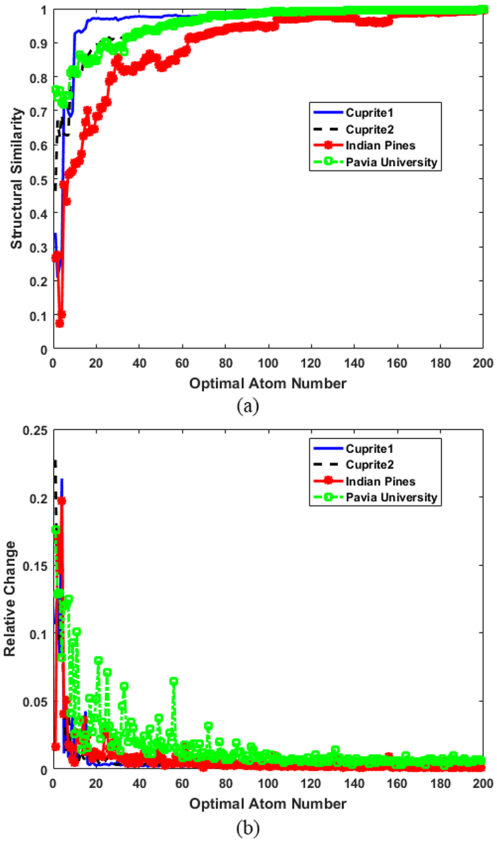

Fig. 6. The SSIM and relative change with different atom number for four band images using ICPSO. (a) SSIM; (b) Relative change.

Table 1. Comparison between OMP and proposed ICPSO.

Hyperspectral datasets	Algorithm	PSNR/dB	SSIM	Runtime/s
Cuprite1	OMP	40.2079	0.9792	372.81
	ICPSO	41.9437	0.9870	108.17
Cuprite2	OMP	39.6879	0.9722	379.93
	ICPSO	41.8877	0.9842	125.05
Indian Pines	OMP	31.8922	0.9787	93.47
	ICPSO	32.8223	0.9836	26.57
Pavia University	OMP	39.0706	0.9717	375.68
	ICPSO	40.0297	0.9778	113.21

Pavia University are given here. Objectively speaking, the PSNR of ICPSO is higher than that of OMP. From the naked eye, the difference between the two reconstructed images and the original image is very small. This fully demonstrates that the atoms searched by ICPSO can represent the original images effectively. Compared with OMP, the proposed ICPSO could improve the reconstruction accuracy as well as the computation efficiency.

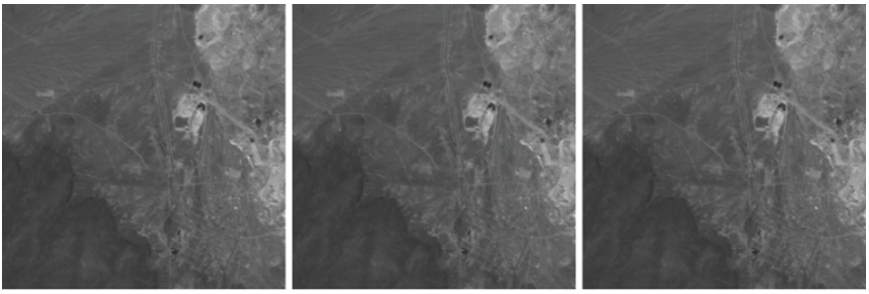

Fig. 7. The 50th band image comparison of Cuprite2. (a) Original image, (b) reconstructed image using OMP, PSNR = 33.5874 dB, and (c) reconstructed image using proposed ICPSO, PSNR = 34.6696 dB.

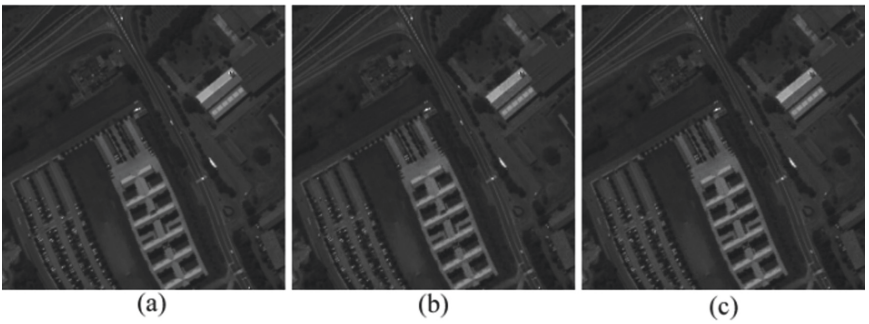

Fig. 8. The 50th band image comparison of Pavia University. (a) Original image, (b) reconstructed image using OMP, PSNR = 39.6655 dB, and (c) reconstructed image using proposed ICPSO, PSNR = 40.1998 dB.

5 Conclusion

Combining immune cloning and evolutionary theory, an immune clone particle swarm optimization algorithm (ICPSO) for sparse representation of hyperspectral images is proposed. Based on the OMP algorithm, particle position and velocity updating is used to simulate the process of searching optimal atoms. The idea of immune cloning is introduced, with immune cloning, cloning mutation, and cloning selection operators on the particle population, to prompt the particle population converging quickly to the optimal solution. We discussed the effects of optimal atoms number, population size and maximum evolutionary algebra on the performance of the algorithm through experiments,

and determined the parameters for OMP and ICPSO. The reconstructed images are represented by the optimal atoms searched by OMP and ICPSO, reconstructed PSNR, SSIM and runtime are compared. The results carried on four hyperspectral datasets could demonstrate that the proposed ICPSO could find the optimal atoms to sparse represent the original images. Compared with OMP algorithm, the proposed ICPSO has 1–2 dB higher PSNR and SSIM than that of OMP, whereas consume less time. In summary, ICPSO algorithm could improve the reconstruction accuracy as well as computation efficiency for sparse representation of hyperspectral images.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under grant number 61901350, Aeronautical Science Foundation of China under grant number 2019ZH0T7001, and Science Research Foundation of Xi'an Aeronautical University under grant number 2019KY0208.

References

1. Chang, C.: A review of virtual dimensionality for hyperspectral imagery. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **11**(4), 1285–1305 (2018). https://doi.org/10.1109/JSTARS.2017.2782706
2. Das, S., Jain, L., Das, A.: Deep learning for military image captioning. In: 2018 21st International Conference on Information Fusion (FUSION), Cambridge, pp. 2165–2171 (2018). https://doi.org/10.23919/ICIF.2018.8455321
3. Niu, Y., Wang, B.: Extracting target spectrum for hyperspectral target detection: an adaptive weighted learning method using a self-completed background dictionary. IEEE Trans. Geosci. Remote Sens. **55**(3), 1604–1617 (2017). https://doi.org/10.1109/TGRS.2016.2628085
4. Pan, L., Li, H., Li, W., Chen, X., Wu, G., Du, Q.: Discriminant analysis of hyperspectral imagery using fast kernel sparse and low-rank graph. IEEE Trans. Geosci. Remote Sens. **55**(11), 6085–6098 (2017). https://doi.org/10.1109/TGRS.2017.2720584
5. AlSuwaidi, A., Grieve, B., Yin, H.: Feature-ensemble-based novelty detection for analyzing plant hyperspectral datasets. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **11**(4), 1041–1055 (2018). https://doi.org/10.1109/JSTARS.2017.2788426
6. Pan, L., Li, H., Meng, H., Li, W., Du, Q., Emery, W.J.: Hyperspectral image classification via low-rank and sparse representation with spectral consistency constraint. IEEE Geosci. Remote Sens. Lett. **14**(11), 2117–2121 (2017). https://doi.org/10.1109/LGRS.2017.2753401
7. Picone, D., Dolet, A., Gousset, S., Voisin, D., Mura, M.D., Le Coarer, E.: Characterisation of a snapshot Fourier transform imaging spectrometer based on an array of fabry-perot interferometers. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, pp. 1529–1533 (2020). https://doi.org/10.1109/ICASSP40776.2020.9053032
8. Uto, K., Seki, H., Saito, G., Kosugi, Y., Komatsu, T.: Measurement of a coastal area by a hyperspectral imager using an optical fiber bundle, a swing mirror and compact spectrometers. In: 2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Los Angeles, CA, pp. 1–4 (2016). https://doi.org/10.1109/WHISPERS.2016.8071670
9. Uto, K., Seki, H., Saito, G., Kosugi, Y., Komatsu, T.: Development of hyperspectral imaging system using optical fiber bundle and swing mirror. In: 2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Tokyo, pp. 1–4 (2015). https://doi.org/10.1109/WHISPERS.2015.8075489

10. Fang, L., He, N., Li, S., Plaza, A.J., Plaza, J.: A new spatial-spectral feature extraction method for hyperspectral images using local covariance matrix representation. IEEE Trans. Geosci. Remote Sens. **56**(6), 3534–3546 (2018). https://doi.org/10.1109/TGRS.2018.2801387
11. Zhang, X., Huang, W., Wang, Q., Li, X.: SSR-NET: spatial-spectral reconstruction network for hyperspectral and multispectral image fusion. IEEE Trans. Geosci. Remote Sens. https://doi.org/10.1109/TGRS.2020.3018732
12. Sun, L., Wu, F., He, C., Zhan, T., Liu, W., Zhang, D.: Weighted collaborative sparse and L1/2 low-rank regularizations with superpixel segmentation for hyperspectral unmixing. IEEE Geosci. Remote Sens. Lett. https://doi.org/10.1109/LGRS.2020.3019427
13. Mallat, S.G., Zhang, Z.: Matching pursuits with time-frequency dictionaries. IEEE Trans. Sig. Process. **41**(12), 3397–3415 (1993). https://doi.org/10.1109/78.258082
14. Zayyani, H., Babaie-Zadeh, M., Jutten, C.: Bayesian Pursuit algorithm for sparse representation. In: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, pp. 1549–1552 (2009). https://doi.org/10.1109/ICASSP.2009.4959892
15. Yang, W., Peng, J., Sun, W., Du, Q.: Log-euclidean kernel-based joint sparse representation for hyperspectral image classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **12**(12), 5023–5034 (2019). https://doi.org/10.1109/JSTARS.2019.2952408
16. Song, X., Wu, L., Hao, H.: Hyperspectral image denoising base on adaptive sparse representation. In: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), Guangzhou, pp. 735–739 (2018). https://doi.org/10.1109/DSC.2018.00117
17. Yao, L., Du, X.: Identification of underwater targets based on sparse representation. IEEE Access **8**, 215–228 (2020). https://doi.org/10.1109/ACCESS.2019.2962005
18. Tan, S., Sun, X., Chan, W., Qu, L., Shao, L.: Robust face recognition with kernelized locality-sensitive group sparsity representation. IEEE Trans. Image Process. **26**(10), 4661–4668 (2017). https://doi.org/10.1109/TIP.2017.2716180
19. Kulkarni, A., Mohsenin, T.: Low overhead architectures for OMP compressive sensing reconstruction algorithm. IEEE Trans. Circ. Syst. I Regul. Pap. **64**(6), 1468–1480 (2017). https://doi.org/10.1109/TCSI.2017.2648854
20. Tian, Y., Wang, Z.: An adaptive orthogonal matching pursuit algorithm based on redundancy dictionary. In: 2013 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Shenyang, pp. 578–582 (2013). https://doi.org/10.1109/FSKD.2013.6816263
21. Shivagunde, S., Biswas, M.: Saliency guided image super-resolution using PSO and MLP based interpolation in wavelet domain. In: 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, pp. 613–620 (2019). https://doi.org/10.1109/ICCES45898.2019.9002042
22. Wang, R., Wu, Y., Shen, M., Cao, W.: Sparse representation for color image based on geometric algebra. In: 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, pp. 1–6 (2018). https://doi.org/10.1109/ICME.2018.8486524
23. Chardon, G., Necciari, T., Balazs, P.: Perceptual matching pursuit with Gabor dictionaries and time-frequency masking. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, pp. 3102–3106 (2014). https://doi.org/10.1109/ICASSP.2014.6854171
24. Jia, S., Hu, J., Tang, G., Shen, L., Deng, L.: Gabor feature based dictionary fusion for hyperspectral imagery classification. In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, pp. 433–436 (2015). https://doi.org/10.1109/IGARSS.2015.7325793
25. Guo, H., Li, B., Li, W., Qiao, F., Rong, X., Li, Y.: Local coupled extreme learning machine based on particle swarm optimization. Algorithms **11**(11), 174 (2018). https://doi.org/10.3390/a11110174

26. Shen, Y.: Research on swarm size of multi-swarm particle swarm optimization algorithm. In: 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, pp. 2243–2247 (2018). https://doi.org/10.1109/CompComm.2018.8781013
27. Yang, Z., Chen, L., Liu, S., Liu, Z., Chen, Q., Li, H.: Coverage control strategy based on multi-objective optimization of immune clone in wireless sensor networks. In: 2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP), Wanzhou, pp. 79–85 (2018). https://doi.org/10.1109/ICICIP.2018.8606691
28. Li, L., Lin, W., Lin, Q., Ming, Z.: Balancing convergence and diversity in multiobjective immune algorithm. In: 2020 12th International Conference on Advanced Computational Intelligence (ICACI), Dali, China, pp. 102–109 (2020). https://doi.org/10.1109/ICACI49185.2020.9177787
29. Dai, H., Chen, D., Zheng, S.: Effects of random values for particle swarm optimization algorithm. Algorithms **11**(2), 23 (2018). https://doi.org/10.3390/a11020023
30. Wang, L., Feng, Y., Gao, Y., Wang, Z., He, M.: Compressed sensing reconstruction of hyperspectral images based on spectral unmixing. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **11**(4), 1266–1284 (2018). https://doi.org/10.1109/JSTARS.2017.2787483