IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 26, 2019, accepted September 6, 2019, date of publication September 17, 2019, date of current version October 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2941725

A Fuzzy Selection Compressive Sampling Matching Pursuit Algorithm for Its Practical Application

HU YUNFENG AND ZHAO LIQUAN

Key Laboratory of Modern Power System Simulation and Control and Renewable Energy Technology, Ministry of Education (Northeast Electric Power University), Jilin 132012, China

Corresponding author: Zhao Liquan (zhao_liquan@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 61271115.

ABSTRACT In compressive sampling matching pursuit algorithm, it requires that the sparsity information of original signal to control the size of the preliminary atomic set and the maximum number of the algorithm iteration. This weakens the reconstruction accuracy, increases the computation complexity and limits its practical application capacity. To overcome the problem, an improved method is proposed. The proposed method firstly sets a fixed step-size as the assumed sparsity to expand the preliminary atomic set at the initial stage when the sparsity information is unknown. Secondly, the proposed algorithm adopts the fuzzy threshold strategy to select the more relevant atoms from the preliminary atomic set to expand the candidate atomic set. Finally, the double threshold control method, multiply stages setting and variable step-size method are used to control the iteration stop condition and adjust the estimated sparsity. When the two threshold iteration stop conditions are simultaneously satisfied, the iteration stops, which shows that the reconstructed signal better approximated the original signal, and the reconstruction performance is the best. Otherwise, if only one of the conditions is satisfied, the size of the estimated sparsity is increased by the variable step size method to reduce the error between the reconstructed signal and original signal. In addition, we extended the proposed algorithm to the multiple measurement vectors scenario for joint sparse signal recovery. Simulation results indicate that the proposed algorithm is better than the other method in terms of the reconstruction performance in single measurement vector and multiple measurement vector cases.

INDEX TERMS Compressed sensing, compressive sampling matching pursuit, adaptive sparsity, fuzzy threshold, variable step-size, multiply stage, single measurement vector, multiple measurement vector.

I. INTRODUCTION

Compressed sensing (CS) [1] is a novel signal compression and processing theory. Compared with the Nyquist sampling theory, the most important of CS theory is that it can be randomly sampled by standards that are far less than Nyquist's, and the original signal can be recovered under small distortion rates. CS theory differs from the traditional Nyquist sampling theory, which includes three aspects: sparse representation of the signal, design of the measurement matrix, and design of the reconstruction algorithm. A crucial step to implement is design of the reconstruction algorithm. The CS framework is attractive, as it implies that x can be faithfully recovered from only $M = O(K \log N)$ samples [2], suggesting the significant cost reduction in digital data acquisition.

At present, many reconstruction algorithms have been proposed to obtain the sparse signal and approximated sparse signal from compressed measurements. There are two major classes of reconstruction algorithms: convex optimization methods and matching pursuit methods. Convex optimization methods include basis pursuit (BP) [3], gradient projection for sparse reconstruction (GPSR) [4], iterative threshold(IT) [5], iterative hard threshold (IHT) [6], interior-point method [7], Bergman iteration (BT) [8], and total variation (TV) [9]. Although it requires fewer measurements, its computational complexity is higher, which leads to slower convergence. Matching pursuit algorithms that are based on the idea of iterative greedy pursuit are popular sparse reconstruction algorithms. The earliest ones include the matching

The associate editor coordinating the review of this manuscript and approving it for publication was Kezhi Li.

pursuit (MP) [10]. Based on the MP algorithm, the orthogonal matching pursuit (OMP) [11] algorithms was proposed to optimize the MP via orthogonalization of the atoms of the support set. Their successors include the stage-wise OMP (StOMP) [12], regularized OMP (ROMP) [13], subspace pursuit (SP) [14], compressive sampling matching pursuit (CoSaMP) [15], [16] and sparsity adaptive matching pursuit(SAMP) [17] algorithms. The StOMP algorithm selects multiple atoms (columns) of the measurement matrix via a threshold parameter. The ROMP algorithm filters the selected atoms of the measurement matrix with regularized rule to reduce the running time of the OMP. The SP and CoSaMP algorithm was a similar algorithm. Both of these algorithms are proposed with the idea of backtracking, and the differences are that SP selects the most relevant K columns from the measurement matrix in each iteration, while CoSaMP selects the most relevant $2K$ columns from the measurement matrix. The CoSaMP method requires the sparsity information, which limits its application. We are inspired by the SAMP method to deduce a sparsity adaptive CoSaMP method. Furthermore, we use two threshold strategies to control the stop condition, to reduce reconstruction error. Recently, the multiple measurement vectors (MMV) model is extended to compressed sensing theory to solve the jointly sparse recovery problem [18]–[22]. Therefore, we also extend the proposed algorithm to the MMV scenario.

The layout of this paper is as follows. Section II introduces the CS theory, including the sparse signal reconstruction for Single Measurement Vector (SMV) model and the joint sparse signal reconstruction for MMV model. The CoSaMP algorithm is described in Section III. The proposed methods with SMV model and MMV model are deduced in Section IV. The simulation results and the discussion are provided in Section V, and the conclusion is drawn in Section VI.

II. COMPRESSED SENSING THEORY

A. SINGLE MEASUREMENT VECTOR MODEL

The compressed sensing is a powerful framework for signal acquisition, which asserts that we can recover signals from far fewer measurements than traditional methods. In Single Measurement Vector (SMV) model, it supposes that a signal $x \in R^N$ is K -sparse or compressible, where K is the sparsity level. The signal x can be well-approximated using a sparse orthogonal basis $\Psi = \{\psi_1, \psi_2, \ldots \psi_N\} \in R^{N\times N}$, with notes $K << N$. According to the CS theory, such a signal x can be defined as

$$x = \sum_{i=1}^{N} \psi_i \theta_i = \Psi\theta \tag{1}$$

where, $\theta = \Psi^T x$ is the $N \times 1$ column vector of projection coefficients, $\theta_i = \langle x, \psi_i \rangle = \psi_i^T x$ is the projection coefficient and $(\cdot)^T$ denotes the transpose operation. Therefore, we can discovery that x and θ are equivalent representations of the original signal. x is the representation of the signal in the time domain and θ is the representation of the signal in the Ψ domain. Generally, we consider this step a sparse representation of the signal. Next, we need to design a measurement matrix Φ that is unrelated to the sparse orthogonal basis Ψ, obtaining an observing vector y, expressed as

$$y = \Phi x + w \tag{2}$$

where $y \in R^{M\times 1}$, $\Phi \in R^{M\times N} (M << N)$ represents the observing vector and a random measurement matrix, respectively. $w \in R^{M\times 1}$ denotes that the additive noise vector, and which is usually Gaussian white noise. This reconstruction problem, approximately estimating x from (2) using the measurement matrix Φ and the observing vector, is known as the SMV problem. Obviously, equation (2) can be regarded as a linear projection of the signal x on the Φ. Now, we consider reconstruction of x from y. However, we can see that the dimension of $y$is much lower than the dimension of x, meaning (2) has infinitely many solutions. That is, (2) is an underdetermined equation, and it is difficult to obtain an accurate reconstruction of the original signal x using the conventional 'inverse' transforms of Φ. Whereas, it is well known that, with prior information on the signal sparsity K and Φ meeting certain conditions, x can be reconstructed by solving the l_p-minimization problem:

$$\min \ \|x\|_0 \quad \text{subject to } \Phi x = y \tag{3}$$

$$\min \ \|x\|_1 \quad \text{subject to } \Phi x = y \tag{4}$$

where $\|.\|$ represents that the l_0 -norm and l_1 -norm of the vector. In formulation (3), the problem is an l_0 -minimization problem, which can be reconstructed by a greedy pursuit algorithm. In formulation (4), the problem is an l_1 -minimization problem, which can be solved using a convex optimization. A condition of Φ ensuring the exact recovery of x is called the restricted isometric property (RIP) [2], [23].

Definition1: For each integer $K = 1, 2, \ldots,$ define the isometry constant δ_K of a measurement matrix Φ as the smallest number such that

$$(1 - \delta_K) \|x\|_2^2 \le \|\Phi x\|_2^2 \le (1 + \delta_K) \|x\|_2^2 \tag{5}$$

holds for all K -sparse vector x and a vector is said to be K -sparse if it has at most K -nonzero entries, namely, $\|x\|_0 \le K$.

B. MULTIPLE MEASUREMENT VECTORS MODEL

Recently, the reconstruction problem of finding sparse representation of multiple measurement vectors (MMV) in a redundant dictionary was motivated by EEG/MEG source localization and DOA estimation, where sequences of measurement vectors are available. Obviously, the reconstruction problem of MMV can be regard as how to simultaneously reconstruct multiple one dimensional sparse signals from the single measurement vector.

In SMV problem, we say that, a vector $x \in R^{N\times 1}$ is K -sparse if it has at most K non-zeros components, the sparsity level and the support of the x can be represented by the index set of the non-zeros components and their number, respectively. Similarly, in MMV model, we let $\{x_i\}_{i=1}^N \in R^N$

be the jointly K -sparse (i.e. the union of the support set of x_i has at most K non-zeros elements). Therefore, the joint-sparse signal X has no more than K non-zero rows, and we also call as the row K -sparse. Where, the $X \in R^{N\times l}$, that is, X has l one-dimensional K -sparse signal with the same nonzero positions. Besides, it is noted that the support of X represents the set of indices of non-zero rows. Therefore, such jointly sparse signal can be expressed as

$$y^{(l)} = \Phi x^{(l)} + w^{(l)} \tag{6}$$

where, $\Phi \in R^{M\times N}$ represents that the measurement matrix, $x^{(l)} \in R^{N\times 1}$ represents that the l-th source vector, and $y^{(l)}$ is the measurement vector corresponding to the source vector $x^{(l)}$. It is noted that, the size of support of $x^{(l)}$ is $\left|\text{supp}\left(x^{(l)}\right)\right| \leq K$ and $x^{(l)}$ have a common support set for $l = 1, 2, \ldots L$. $w^{(l)} \in R^{M\times 1}$ denotes the additive noise vector to the $y^{(l)}$. For simplicity, the equation (6) can be rewrite as

$$Y = AX + W \tag{7}$$

where, $X = \left[x^{(1)}, x^{(2)}, \ldots, x^{(L)}\right]$, $W = \left[w^{(1)}, w^{(2)}, \ldots, w^{(L)}\right]$ and $Y = \left[y^{(1)}, y^{(2)}, \ldots, y^{(L)}\right]$. From the equation (6) and (7), we can known that the measurement process has a common measurement matrix Φ and the noise matrix W.

We described the minimize formulation of MMV model. From the existing equations (3) and (4) in SMV scenario, similarly, we give the solutions to the following optimization problem in noiseless and MMV scenarios:

$$\min \ \|X_l\|_0 \quad \text{subject to } \Phi X_l = Y_l \tag{8}$$

$$\min \ \|X_l\|_1 \quad \text{subject to } \Phi X_l = Y_l \tag{9}$$

where, $X \in R^{N\times L}$ and X_l is the l-th column vector in matrix X. Here, $l = 1, 2, \ldots, L$, L is the number of the columns of X. Y_l is the l-th column vector of matrix Y. From the equations (3) and (4) in SMV model and the equations (8) and (9) in MMV model, we can known that the above optimization problem in equations (8) and (9) can equivalent as the optimization problems in equations (3) and (4) when the number of column is equal to 1, that is, $L = 1$.

III. CoSaMP ALGORITHM

This section presents a summary of existing matching pursuit algorithms for CS. Recently, the matching algorithms are widely used for signal reconstruction due to the simple structure and low computational complexity. The algorithms can be divided into top-down and bottom-up categories. Bottom-up methods such as OMP assume a possible initial solution before the iteration to develop the final solution. Top-down methods such as SP and CoSaMP use the backtracking strategy to more accurately determine the true support set of atoms, and the final solution of the signal is obtained by the least squares methods. Although the computational complexity of backward tracking methods is higher, they can be more accurate in many cases. We introduce the CoSaMP.

To improve the convergence rate and efficiency of the algorithm, in the preliminary stage of the algorithm, it chooses $2K$ atoms (columns) that are the most relevant to the residual of the last iteration from the measurement matrix to form the preliminary atomic set, and rejects the irrelevant atoms. Then, we merge into the support atomic set of the last iteration and current preliminary atomic to update the candidate atomic set. Next, the transition solution of the signal is obtained by the least square method. Finally, the support set of the signal is obtained by prune the candidate atomic set. The specific algorithm procedure is displayed in Algorithm 1.

The entire procedure is as shown in Algorithm 1.

Algorithm 1 CoSaMP Algorithm

Input: matrix Φ, observation signal y, sparsity level$K$
Output: K-sparse approximation $\hat{x}$ to original signal x
Initialize: $\hat{x} \leftarrow 0$, $r \leftarrow y$, $k \leftarrow 0$
repeat

$k \leftarrow k + 1$	Loop index
$e \leftarrow \Phi^T \times r$	Form signal proxy
$\Omega \leftarrow \text{supp}\left(\lvert e\rvert, 2K\right)$	Identify large $2K$ components
$T \leftarrow \Omega \cup \Gamma_{k-1}$	Merge supports
$b\lvert_T \leftarrow \Phi_T^{\dagger} y$	Signal estimation by least square
$\Gamma \leftarrow \text{supp}\left(\lvert b\rvert, K\right)$	Prune to obtain current supports
$\hat{x}_k \leftarrow b_{k\Gamma}$	Update the final approximation
$r \leftarrow y - \Phi\hat{x}_k$	Computing current residual

Until halting criterion true; return $\hat{x} \leftarrow \hat{x}_k$

IV. PROPOSED ALGORITHM

The CoSaMP algorithm requires that the sparsity of the signal is known, and uses the size of the sparsity to set the number of selected atoms. However, the sparsity of the signal is usually unknown in practice, which limits the practical application of the algorithm. To solve the problem, we first randomly set the estimated sparsity as $\hat{K}$, and step-size $s(\hat{K} = s)$. In the following, we deduce the proposed method for SMV model and MMV model, respectively.

A. PROPOSED ALGORITHM FOR SMV

1) SELECTION OF ATOMIC

In SMV model, we firstly select the $2\hat{K}$ atoms to expand the set of atoms, then calculate the atomic coefficient C_k and select $2\hat{K}$ maximum atomic correlations to determine their corresponding atomic index and form an atomic index set Ω. The values of C_k and Ω can be expressed as

$$C_k = \left\{c_j | c_j = \left\langle r_{k-1}, \phi_j \right\rangle\right\} \tag{10}$$

$$\Omega = \text{supp}\left(|C_k|, 2\hat{K}\right) \tag{11}$$

where $j = 1, 2, \ldots, N$, k is the number of iterations, r_{k-1} is the residual of $(k-1)th$ iterations, ϕ_j is the j-th column of the measurement matrix Φ. $< . >$ represents the inner product of the previous residual and the atomic of the measurement matrix. C_k is the correlation coefficients at k-th iteration, and the length of the C_k is N. In the first iteration, we let r_k equal y. $\text{supp}\left(|\cdot|, 2\hat{K}\right)$ is used to determine the atomic index

corresponding to the $2\hat{K}$ maximal value from e, and forms a preliminary atoms index set Ω.

In the CoSaMP algorithm, if the known sparsity is K, it will select $2K$ atoms to add into the candidate support sets. This will consume much more time to select K better atoms to former new support sets. To reduce complexity, we use the fuzzy threshold method to control the number of selected atoms that will be added into the candidate atomic set.

In the atomic index set Ω, the atoms corresponding to the index are expressed as τ, $\tau = [\tau_1, \tau_2, \cdots \tau_{2\hat{K}}]$. We select the atoms that are larger than the fuzzy threshold in the atomic set τ, to form a new atomic set, and the index of the new atomic set forms the new atomic index set Ω^*. This process is described as

$$\omega_k = \alpha_k \times (c_{k1} + c_{k2}) \times t \tag{12}$$

$$C_k^* = |C_k|_{2\hat{K}} \tag{13}$$

$$\Omega^* = \text{supp}\left(C_k^*, \omega_k\right) \tag{14}$$

where, ω_k is the fuzzy threshold at the k-th iteration, c_{k1} and c_{k2} correspond to the first and second components after sorting C_k in descending order in (11), the specify value of the $\alpha_k = c_{k2}/c_{k1}$, and the variable range of α_k is [0.60 0.90] according our experiments. The α_k is not fixed value that we set. The specify value of α_k is determined by the coefficients c_{k1} and c_{k2}. In (12), we can known that the fuzzy threshold ω_k is control by the α_k, the sum of the coefficients c_{k1} and c_{k2}, and the threshold t. Besides, due to the atomic relation coefficients c_{k1} and c_{k2} is different at each iteration, the specify value of ω_k is also not the same at each iteration, which ensure that the proposed method select more atoms with higher relevant at each iterations. In terms of the parameter α_k, we set it as $\alpha_k = c_{k2}/c_{k1}$, mainly because that the first two atomic relevant coefficients can more accurately measure the atomic reliability at each iteration, thereby searching the atoms with higher relevant to add it to the new preliminary atomic set.

For the threshold t, if the threshold is too larger, the number of the atoms with higher relevant will reduce in the new preliminary atomic index set via fuzzy threshold selection strategy selected. This will increase the computation complexity of the algorithm and reduce the reconstruction accuracy of original signal. Meanwhile, if the size of the threshold is too smaller, it will make the redundant atoms and lower relevant atoms cannot be effectively rejected. Specifically, if we select a too large threshold t and take it as the input parameter of the proposed algorithm, such as $t = 0.8$, it will not only eliminate too many preliminary atoms with high reliability in preliminary atomic set, but also cause the scale of the preliminary atomic set to be too small. These defects will lead to the inaccuracy of the transition signal estimated by the least square method and the inaccuracy of the support atom set estimated by pruning the transition signal, which causes the algorithm need more iteration to complete the reconstruction of the algorithm. Therefore, if we choose a larger threshold, the computational complexity of the proposed method will increase and the reconstruction precise of the proposed method will decline. On the contrary, when we select a too smaller threshold t, such as $t = 0.01$, it will not only effectively reject the preliminary atoms with lower reliability and higher redundantly in preliminary atomic set, but also cause the size of the preliminary atomic set to be too larger. These shortcomings will lead to the inaccuracy of the transition signal and the final support atomic set, which causes the algorithm need more iteration to complete the reconstruction of the algorithm. Therefore, if we choose a too small threshold, the computational complexity of the proposed method will increase and the reconstruction precise of the proposed method will decline. Therefore, based on the above analysis on the threshold t, we should select a suitable threshold that is neither small nor large. According the experiments in the section V for different source signals, we test the proposed method performance with $t \in [0.3\ 0.5]$. It shows the proposed method has better performance for different source signals with $t = 0.4$, $t = 0.45$ and $t = 0.5$, and the performance has little difference. Therefore, we randomly select the threshold $t = 0.5$.

Combining the equation (10) and (13), we can known that the new atomic coefficient C_k^* is sub-set of the initial atomic coefficient set C_k, that is, C_k^* is belongs to C_k. Note that, the length of C_k^* is equal to $2\hat{K}$. Besides, supp $((\cdot), \omega_k)$ represents that searches the corresponding index that satisfies the fuzzy-threshold condition and forms a new atomic index set Ω^*.

After the fuzzy threshold selection strategy is completed for SMV model, we need to merge the atomic set Ω^* and the support set Γ_{k-1} to update the current candidate atom set T, described as:

$$T = \Omega^* \cup \Gamma_{k-1} \tag{15}$$

where T, Ω^*, Γ_{k-1} represent the current candidate atomic index set, the current atomic index set, and the support atomic index set of the previous iteration, respectively.

Next, we use the least squares method to obtain the transition estimation of original signal for SMV model, which can be described as in follow:

$$b_{kT} = \Phi_T^{\dagger} y \tag{16}$$

where, b_{kT} is the transition estimation of the sparse signal x by the least square method at the k-th iteration and the position of the non-zeros is consistent with the atomic candidate index set. This is known as the transition signal. $\Phi_T^{\dagger}$ represents that the pseudo inverse matrix of the candidate atomic matrix Φ_T (or set), and y is a observing vector, which is used to storage the information of original signal.

Finally, we update the support index set F, described as:

$$\Gamma = \text{supp}\left(|b_{kT}|, \hat{K}\right) \tag{17}$$

where $|\cdot|$ represents is the absolute value of transition signal b_{kT} at k-th iteration. supp $\left(|\cdot|, \hat{K}\right)$ is used to determines the

atomic(or column) index of the measurement matrix corresponding to the maximum $\hat{K}$ value from transition signal b_{kT} and consist of the support atomic index set Γ. Using the atoms corresponding to the support set index set, we construct the new support set Φ_Γ.

Next, we update the final estimation of sparse signal x at k-th iteration, which is expressed as

$$x_k = b_{k\Gamma} \tag{18}$$

where x_k is the final approximation estimation of the original signal at the k-th iteration, and $b_{k\Gamma}$ is the recovery signal corresponding to the support atomic index set Γ.

The fuzzy threshold strategy accords the atomic correlation degree to secondly select the preliminary atoms. This can improve the reliability of the preliminary atoms and eliminate the atoms with low correlation degree in the pre-selected atoms to make the supporting atom set more be accuracy. Therefore it can reduce the reconstruction error. In addition, it can faster select the useful atoms, so it reduces the computation time for convergence.

2) TERMINATION CONDITION

In the termination condition of the CoSaMP algorithm, one of the iterative halt conditions is that the maximum iteration number is the sparsity value. That is, it stops until the iteration reaches the maximum iteration number. However, in practical, the true sparsity level is unknown. Therefore, the termination condition is unsatisfied if we still use the sparsity value as the maximum iteration number. To overcome this problem, we propose a double threshold control method and multiple stage variable step-size method to control the convergence condition of the algorithm and adjust the estimated sparsity, thereby getting rid of the dependency of sparsity information of original signal. In particular, we set the maximum number of iteration is set as the number of measurements M to ensure the reconstruction accuracy when the estimated sparsity is much lower than the real sparsity of original signal.

First, we compute the residual by

$$r_c = y - \Phi x_k \tag{19}$$

where r_c is the current residual. Next, we use the difference between the current residual value and the iterative halt error to decide how to adjust the estimated sparsity L. If

$$\|r_c\|_2 \leq \varepsilon_2 \quad \text{or } k > M \tag{20}$$

then the algorithm stops the iteration and outputs the final estimation of the original signal $\hat{x}$. $\|.\|_2$ denotes that the l_2 -norm of residual vector r_c. ε_2 and M are the iteration stopping error threshold of the algorithm and the number of measurement values, respectively. It is noted that, if the current residual r_c is smaller than the second threshold ε_2, this means that the estimated sparsity is better approach the real sparsity of the original signal and the reconstruction accuracy can satisfy that we require. Besides, to avoid the excessive estimation of the sparsity, we set that the number of the loop index k is greater than the number of measurement values M. If the condition in equation (20) is not satisfied, the other condition should be judge, that is, If

$$\varepsilon_2 \leq \|r_c\|_2 \leq \varepsilon_1 \tag{21}$$

then

$$s = s \tag{22}$$
$$J = J + 1 \tag{23}$$
$$\hat{K} = J \times s \tag{24}$$

where ε_1 is another iteration stopping error threshold of the proposed algorithm ($\varepsilon_1 > \varepsilon_2$), and s, J and $\hat{K}$ are the fixed step-size, stage index and estimated sparsity, respectively. If the condition is satisfy, it is indicates that the estimated sparsity is close to the real sparsity, but it not approach to the real sparsity. So, we still need to use the smaller step-size to complete the approximately approach of the real sparsity of original sparsity, and enhancing the reconstruction accuracy and capacity. Here, the double halt error threshold ε_1 and ε_2 that we set according experiment and some relevant reference literatures [13], [15], [16]. We set the double thresholds that is mainly to more accurate approximately approach the true sparsity of the original signal, and enhance the reconstruction capacity in unknown sparsity environment. When the 2-norm of the current residual r_c is smaller than the first threshold ε_1 but greater than the second threshold ε_2, the estimated sparsity is gradually close to the true sparsity but need to achieve a more accurately approach by utilized the smaller iterative step-size, which will enhance the accuracy of the final estimation signal and the support atomic set at next iteration. If

$$\|r_c\|_2 > \varepsilon_1 \quad \text{and } \|r_c\|_2 \geq \|r_{k-1}\|_2 \tag{25}$$

then

$$s = 2 \times s \tag{26}$$
$$J = J + 1 \tag{27}$$
$$\hat{K} = J \times s \tag{28}$$

where, r_c and r_{k-1} are the current and the previous residuals. It is noted that, when the current residual is greater than the threshold ε_1, it indicates that the estimated sparsity is smaller than the real sparsity and the reconstruction accuracy is not satisfy that we can be acceptable value. Therefore, we utilizes the larger step size to speedy approximate the real sparsity, thereby reducing the adjusted runtime of estimated sparsity, which will lower the computational complexity. If

$$\|r_c\|_2 > \varepsilon_1 \quad \text{and } \|r_c\|_2 < \|r_{k-1}\|_2 \tag{29}$$

then continue to iterate and update the parameters:

$$r_k = r_c \quad \text{and } \Gamma_k = \Gamma \tag{30}$$

where r_k and Γ_k are the current residual and the support index set, which will be used to compute atomic correlation

coefficients and the expansion of the candidate atomic set in the next iteration.

The entire procedure of the proposed method for SMV model is shown in Algorithm 2.

Algorithm 2 Proposed Algorithm for SMV Model

Input: matrix Φ, observation signal y, step-size s
Initialize: $\hat{x} \leftarrow 0, r \leftarrow y, k \leftarrow 0, J \leftarrow 0, \hat{K} \leftarrow s$
Output: K-sparse estimation $\hat{x}$ to original signal x
repeat
  $k \leftarrow k+1$ loop index
  $C \leftarrow \Phi^T \times r$ form signal proxy for SMV
  $\Omega \leftarrow \text{supp}\left(|C|, 2\hat{K}\right)$ Identify large $2\hat{K}$ components
  $\omega \leftarrow \alpha \times (c_1 + c_2)/2$ computing fuzzy threshold
  $C^* \leftarrow |C|_{2L}$ obtain the maximum $2\hat{K}$ correlations
  $\Omega^* \leftarrow \text{supp}(|C^*|, \omega)$ obtain the new preliminary set
  $T \leftarrow \Omega^* \cup \Gamma_{k-1}$ Merge to form current candidate set
  $b|_T \leftarrow \Phi_T^{\dagger} y$ Transition estimation
  $\Gamma \leftarrow \text{supp}\left(|b|, \hat{K}\right)$ Prune to obtain current supports
  $\hat{x}_k \leftarrow b_{k\Gamma}$ Update the final estimation
  $r_c \leftarrow y - \Phi \hat{x}_k$ Computing current residual
 If halting criterion 1 true **then**
  If halting criterion 2 true **then**
   quit iteration;
  else
   $s \leftarrow s$ Keeping step-size
   $J \leftarrow J+1$ Stage index
   $\hat{K} \leftarrow J \times s$ Update the sparsity estimation
  end
 else if $(\|r_c\|_2 \geq \|r_{k-1}\|_2)$
  $s \leftarrow 2 \times s$ Increasing step size
  $J \leftarrow J+1$ Stage index
  $\hat{K} \leftarrow J \times s$ Update the sparsity estimation
 else
  $r_k \leftarrow r_c$ Update current residual
  $\Gamma_k \leftarrow \Gamma$ Update current support set
 end if
until halting criterion true; return $\hat{x} \leftarrow \hat{x}_k$

The purpose of setting two thresholds is to make estimated sparsity morel close to the real sparsity, which will improve the accuracy of support set estimation and the reconstruction performance. If the residual error is larger than the first threshold (larger threshold), it means that the estimated sparsity is far away from the real sparsity, and then large step size is used to adjust the estimated sparsity to make the estimated sparsity faster close to the real sparsity. If the residual error is between the two thresholds, it means that the estimated sparsity is not far away from the real sparsity, and then small step size is used to adjust the estimated sparsity to avoid overestimating the estimated sparsity. If the residual error is smaller than the second threshold (smaller threshold), it means that the estimated sparsity is enough close to the real sparsity, and it will not adjust the estimated sparsity.

The two thresholds are used to control the step size of sparsity estimation. When the estimated sparsity is far away from the real sparsity (it is evaluated by residual error), it will use larger step size to adjust the estimated sparsity. This can make the estimated sparsity faster close to the real sparsity. When the estimated sparsity is close to the real sparsity, it will use smaller step size to adjust the estimated sparsity. This can avoid overestimating the sparsity.

The values of two thresholds are just used to indirectly evaluate the proximity between the estimated sparsity and the real estimated sparsity. They have nothing to do with the sparsity of source signals. The different value of two thresholds may affect the performance of proposed method.

In the following experiments, we set the two thresholds as $\varepsilon_1 = 1 \times e^{-5}$ and $\varepsilon_2 = 1 \times e^{-7}$ for different source signals, bases on analysis of simulation results, the proposed method has better performance than others for different source signals. Therefore, we can use the two values as the two thresholds.

B. PROPOSED ALGORITHM FOR MMV

1) SELECTION OF ATOMIC

In MMV model, we also select the $2\hat{K}$ atoms to expand the preliminary atomic set, then calculate the atomic coefficient E_k and select the $2\hat{K}$ maximum atomic correlations to identify their corresponding atomic index set Ω, which can be expressed as in follow.

$$E_k = \left\{ e_j \middle| e_j = norm\left(\phi_j^T \times R_{k-1}, p\right) \right\} \tag{31}$$

$$\Omega = \text{supp}\left(E_k, 2\hat{K}\right) \tag{32}$$

where, $j = 1, 2, \ldots .N$, k is the number of iterations in algorithm. R_{k-1} represents that the residual of the previous iteration and the size of residual matrix is $M \times l$. l is the number of source signal, which is joint sparse signal. ϕ_j is the j-th column of the sensing matrix Φ, and $\phi_j^T \times R_{k-1}$ denotes the matrix multiplication of the column atoms of matrix Φ and the previous residual, and it's a vector with $1 \times l$. We name $\phi_j^T \times R_{k-1}$ as the inner product vector, and for simplicity, which can be expressed as Λ_j. $norm(|\cdot|, p)$ represents the l_p -norm operation of inner product vector Λ_j. Here, $p = 2$. E_j is the atomic correlation coefficients for MMV model at k-th iteration. In the first iteration of algorithm, we use observation signal Y as the initial residual R_0. The $\text{supp}\left(E_k, 2\hat{K}\right)$ represents the preliminary atomic index set corresponding to the largest $2\hat{K}$ value in the atomic correlation coefficient E_k at k-th iteration.

After the preliminary stage for MMV model, the preliminary atomic set still exists the redundant atoms, which will reduce the estimation accuracy of the support set. This eventually affects the reconstruction capacity, the computation complexity and the reconstruction precision. Therefore, we also use the fuzzy threshold selection strategy to reduce the number of the lower relevant and redundant atoms that will be added into the follow candidate atomic set. This

process is mainly described as

$$\omega_k = \alpha_k \times (e_{k1} + e_{k2}) \times t \tag{33}$$
$$E_k^* = (E_k)_{2L} \tag{34}$$
$$\Omega^* = \text{supp}\left(E_k^*, \omega_k\right) \tag{35}$$

where, The α_k is not fixed value that we set, and with variability at each of iterations until the proposed method can achieve reconstruction. The variable range of α_k is [0.55 0.95] for MMV model. The specify value of the α_k is determined by the coefficients e_{k1} and e_{k2}, that is, $\alpha_k = e_{k2}/e_{k1}$. The e_{k1} and e_{k2} correspond to the first and second components after sorting E_k in descending order in (32). The t is threshold that we set, and the rang of is usually set as $t \in [0.3\ 0.5]$. It is noted that, if the threshold is greater than 0.5, then ω_k is the fuzzy threshold at the k-th iteration. Combining the equation (31) and (34), we can known that the new atomic coefficient E_k^* is sub-set of the initial atomic coefficient set E_k, that is, E_k^* is belongs to E_k. Note that, the length of E_k^* is $2\hat{K}$. Besides, $\text{supp}\left((\cdot), \omega_k\right)$ represents that searches the corresponding index that satisfies the fuzzy-threshold condition and forms a new atomic index set Ω^*.

After the fuzzy threshold selection strategy is completed for MMV model, we need to merge the atomic set Ω^* and the support set Γ_{k-1} to update the current candidate atom set T, described as:

$$T = \Omega^* \cup \Gamma_{k-1} \tag{36}$$

where T, Ω^*, Γ_{k-1} represent the current candidate atomic index set, the current atomic index set, and the support atomic index set of the previous iteration, respectively.

Next, we use the least squares method to obtain the transition estimation of original signal for MMV model, which can be described as in follow:

$$B_{kT} = \Phi_T^{\dagger} Y \tag{37}$$

where, B_{kT} is the transition estimation of the sparse signal X by the least square method at the k-th iteration and the position of the non-zeros is consistent with the atomic candidate index set. The transition estimation B_{kT} can be considered as matrix with $N \times l$. The number of the rows is $|T|$ and the $|T|$ is the length of candidate atomic index set. For simplicity, B_{kT} is usually named as the transition signal B corresponding to the candidate atomic index set T at k-th. $\Phi_T^{\dagger}$ represents that the pseudo inverse matrix of the candidate atomic matrix Φ_T (or set), and Y is a observing vector with $M \times l$ which is used to storage the information of original signal.

Before pruning the support index set, we need to operate l_2 -norm of each row vector in transition signal B to obtain a transition estimation vector U at each iteration, this process can be described as

$$U_k = \left\{ u_j \middle| \, u_j = norm\left(B_j, 2\right) \right\} \tag{38}$$

where, the U_k call as the estimation vector of transition estimation matrix B at k-th iteration. $j = 1, 2, \ldots..N$, B_j represents that the j-th row of the matrix B. u_j is the estimation value corresponding to the B_j after l_2 -norm operation on the row vector B_j.

Next, we update the support index set F, described as:

$$\Gamma = \text{supp}\left(U_k, \hat{K}\right) \tag{39}$$

where $\hat{K}$ is the estimated sparsity of the original signal at k-th iteration. $\text{supp}\left((\cdot), \hat{K}\right)$ is used to determines the support atomic (or column) index of the measurement matrix corresponding to the maximum $\hat{K}$ value from transition estimation vector U_k and consist of the support atomic index set Γ. Using the atoms corresponding to the support set index set, we construct the new support set Φ_Γ.

Finally, we update the final estimation of joint random sparse signal X at k-th iteration, which is expressed as

$$\hat{X}_k = B_{k\Gamma} \tag{40}$$

where $\hat{X}_k$ is the final approximation estimation of the original signal at the k-th iteration, and $B_{k\Gamma}$ is the recovery signal corresponding to the support atomic index set Γ.

2) TERMINATION CONDITION

In the termination condition of the CoSaMP algorithm, one of the iterative halt conditions is that the maximum iteration number is the sparsity value. That is, it stops until the iteration reaches the maximum iteration number. However, in practical, the true sparsity level is unknown. Therefore, the termination condition is unsatisfied if we still use the sparsity value as the maximum iteration number. To overcome this problem, we propose a double threshold control method and multiple stage variable step-size method to control the convergence condition of the algorithm and adjust the estimated sparsity, thereby getting rid of the dependency of sparsity information of original signal. In particular, we set the maximum number of iteration is set as the number of measurements M to ensure the reconstruction accuracy when the estimated sparsity is much lower than the real sparsity of original signal.

First, we computing the residual by

$$R_c = Y - \Phi X_k \tag{41}$$

where R_c is the current residual. Next, we use the difference between the current residual value and the iterative halt error to decide how to adjust the estimated sparsity $\hat{K}$. If

$$\|R_c\|_2 \leq \varepsilon_2 \quad \text{or } k > M \tag{42}$$

then the algorithm stops the iteration and outputs the final estimation of the original signal $\hat{X}$. $\|.\|_2$ denotes that the l_2 -norm of residual vector R_c. ε_2 and M are the iteration stopping error threshold of the algorithm and the number of measurement values, respectively. It is noted that, if the current residual R_c is smaller than the second threshold ε_2, this means that the estimated sparsity is better approach the real sparsity of the original signal and the reconstruction accuracy can satisfy that we require. Besides, to avoid the excessive estimation of the sparsity, we set that the number of

the loop index k is greater than the number of measurement values M. If the condition in equation (41) is not satisfied, the other condition should be judge, that is, If

$$\varepsilon_2 \leq \|R_c\|_2 \leq \varepsilon_1 \tag{43}$$

then

$$s = s \tag{44}$$
$$J = J + 1 \tag{45}$$
$$\hat{K} = J \times s \tag{46}$$

where ε_1 is another iteration stopping error threshold of the proposed algorithm ($\varepsilon_1 > \varepsilon_2$), and s, J and $\hat{K}$ are the fixed step-size, stage index and estimated sparsity, respectively. If the condition is satisfy, it is indicates that the estimated sparsity is close to the real sparsity, but it not approach to the real sparsity. So, we still need to use the smaller step-size to complete the approximately approach of the real sparsity of original sparsity, and enhancing the reconstruction accuracy and capacity. If

$$\|R_c\|_2 > \varepsilon_1 \quad \text{and} \quad \|R_c\|_2 \geq \|R_{k-1}\|_2 \tag{47}$$

then

$$s = 2 \times s \tag{48}$$
$$J = J + 1 \tag{49}$$
$$\hat{K} = J \times s \tag{50}$$

where, R_c and R_{k-1} are the current and the previous residuals. It is noted that, when the current residual is greater than the threshold ε_1, it is indicated that the estimated sparsity is much smaller than the real sparsity and the reconstruction accuracy is not satisfy that we acceptable maximum iterative threshold. Therefore, we utilizes the larger to speedy approximate the real sparsity, thereby reducing the adjusted runtime of estimated sparsity, which will lower the computational complexity. If

$$\|R_c\|_2 > \varepsilon_1 \quad \text{and} \quad \|R_c\|_2 < \|R_{k-1}\|_2 \tag{51}$$

then continue to iterate and update the parameters:

$$R_k = R_c \quad \text{and } \Gamma_k = \Gamma \tag{52}$$

where R_k and Γ_k are the updated current residual and the update support index set, which will used the calculation of atomic correlation coefficients and the expansion of the candidate atomic set in the next iteration.

The entire procedure of the proposed method for MMV model is shown in Algorithm 3

Algorithm 3 Proposed Algorithm for MMV Model

Input: matrix Φ, observation signal Y, step-size s
Initialize: $\hat{X} \leftarrow 0, R \leftarrow Y, k \leftarrow 0, J \leftarrow 0, \hat{K} \leftarrow s$
Output: K-sparse estimation $\hat{X}$ to original signal X
repeat
  $k \leftarrow k + 1$ loop index
  for iteration j until $j > N$ do
    $E_j \leftarrow norm\left(\Phi_j^T \times R, 2\right)$ form signal proxy
  end for
  $\Omega \leftarrow \text{supp}\left(E, 2\hat{K}\right)$ Identify large $2\hat{K}$ components
  $\omega \leftarrow \alpha \times (E_1 + E_2)/2$ Computing fuzzy threshold
  $E^* \leftarrow E_{2\hat{K}}$ obtain the maximum $2\hat{K}$ correlations
  $\Omega^* \leftarrow \text{supp}\,(E^*, \omega)$ obtain the new preliminary set
  $T \leftarrow \Omega^* \cup \Gamma_{k-1}$ Merge to form current candidate set
  $B|_T \leftarrow \Phi_T^{\dagger} Y$ Transition estimation
  for iteration j until $j > N$ do
    $U_j \leftarrow norm\left(B_j, 2\right)$
  end for
  $\Gamma \leftarrow \text{supp}\left(U_j, \hat{K}\right)$ Prune to obtain current supports
  $\hat{X}_k \leftarrow B_{k\Gamma}$ Update the final estimation
  $R_c \leftarrow Y - \Phi\hat{X}_k$ Computing current residual
  If halting criterion 1 true **then**
    If halting criterion 2 true **then**
      quit iteration;
    else
      $s \leftarrow s$ Keeping step-size
      $J \leftarrow J + 1$ stage index
      $\hat{K} \leftarrow J \times s$ Update the sparsity estimation
    end
  else if $(\|r_c\|_2 \geq \|r_{k-1}\|_2)$
    $s \leftarrow 2 \times s$ Increasing step size
    $J \leftarrow J + 1$ Stage index
    $\hat{K} \leftarrow J \times s$ Update the sparsity estimation
  else
    $R_k \leftarrow R_c$ Update current residual
    $\Gamma_k \leftarrow \Gamma$ Update current support set
end if
until halting criterion true; return $\hat{X} \leftarrow \hat{X}_k$

V. SIMULATION AND DISCUSSION

In this section, we use one dimensional sparse signal, joint sparse signals and two dimensional image signal as original signal. All obtained reconstruction performance is an average performance after running 1000 times using the computer with a quad-core 64-bit processor and 4G memory. We use the average approximation error (*Aae*) and mean square error (*MSE*) as indicators to evaluate the reconstruction precision for noiseless condition and noise condition, respectively. The average approximation error and mean square error are expressed as followings:

$$Aae = \frac{1}{n}\sum_{i=1}^{n}\left\|x - \hat{x}_i\right\|_2 \tag{53}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}E\{(x - \hat{x}_i)^2\} \tag{54}$$

where, x, $\hat{x}_i$ and n represents that the original signal, estimation signal for the *ith* and the number of experiments, respectively. $E\{(x - \hat{x}_i)^2\}$ is the expectation of $(x - \hat{x}_i)^2$. $\left\|x - \hat{x}_i\right\|_2$ is the 2 Forbenius of $(x - \hat{x}_i)$.

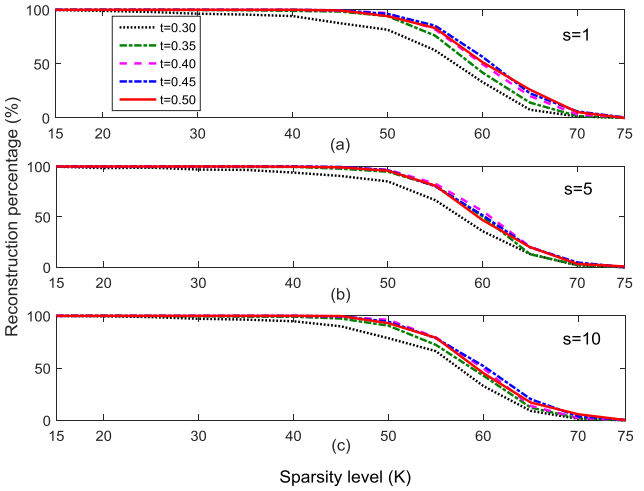


FIGURE 1. Reconstruction percentage of the proposed algorithms with different sparsity levels at different thresholds and step-sizes conditions ($M = 130$, $N = 256$, $t \in [0.3\ 0.5]$).

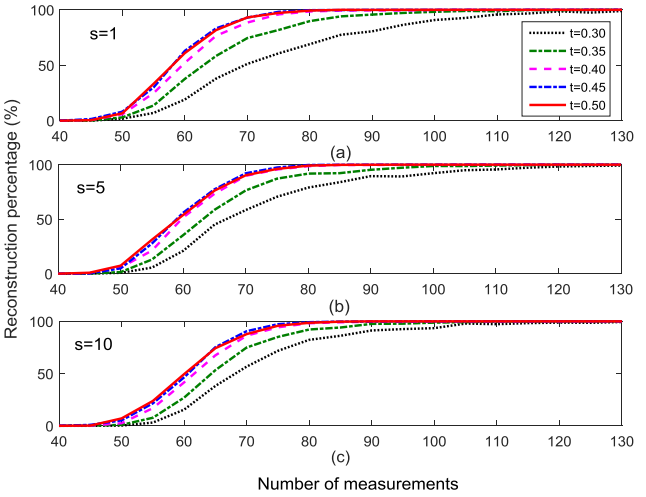


FIGURE 2. Reconstruction percentage of the proposed algorithms with different measurements at different thresholds and step-sizes ($K = 20$, $N = 256$).

A. 1-D SPARSE SIGNAL RECONSTRUCTION FOR SMV MODEL

In this sub-section, we use the random signal with K -sparse and Gaussian distribution as original signal. The measurement matrix is randomly generated with a Gaussian distribution. We set the iteration step-size as $s = 1, 5, 10$. The reconstruction error of all algorithms is $1 \times e^{-6}$. Besides, we set the iterative halt error of the proposed algorithm as $\varepsilon_1 = 1 \times e^{-5}$ and $\varepsilon_2 = 1 \times e^{-7}$, respectively.

In Figure 1, we compare the reconstruction percentage of the proposed algorithm with different sparsity level at different thresholds and step-sizes conditions. In SMV model, we set the sparsity level as $K \in [15\ 75]$. The threshold set is $t \in [0.3\ 0.5]$. The iterative step-size of the proposed method is set as $s \in [1\ 5\ 10]$. Besides, we set the number of measurements (sampling points) as $M = 130$. From Figure 1 (a)-(c), we can see that, for different iterative step-sizes, the reconstruction percentages of the proposed method with $t = 0.4$, $t = 0.45$ and $t = 0.5$ are very close, and are almost no difference, except for $t = 0.3$ and $t = 0.35$. This means that the reconstruction percentages of the proposed method with large threshold are almost the same for different sparsity levels.

In Figure 2, we compare the reconstruction percentage of the proposed algorithm with different measurements in different thresholds and step-sizes conditions. In the Figure 1, it shows that when the sparsity level is smaller than 20 and the number of measurement is equal to 130, the reconstruction percentage of the proposed method is 100% with different thresholds and iterative step-sizes. Therefore, we set the sparsity level as 20 and the range of measurement is from 40 to 130 in Figure 2. The others parameters are same with the parameters in Figure 1.

From Figure 2, we can see that the reconstruction percentages of the proposed method with $t = 0.4$, $t = 0.45$ and $t = 0.5$ are very close, and are almost no difference, except for $t = 0.3$ and $t = 0.35$. This means that the reconstruction percentages of the proposed method with large thresholds are almost the same for different sparsity levels. The proposed method with larger thresholds has higher reconstruction percentage than with smaller threshold.

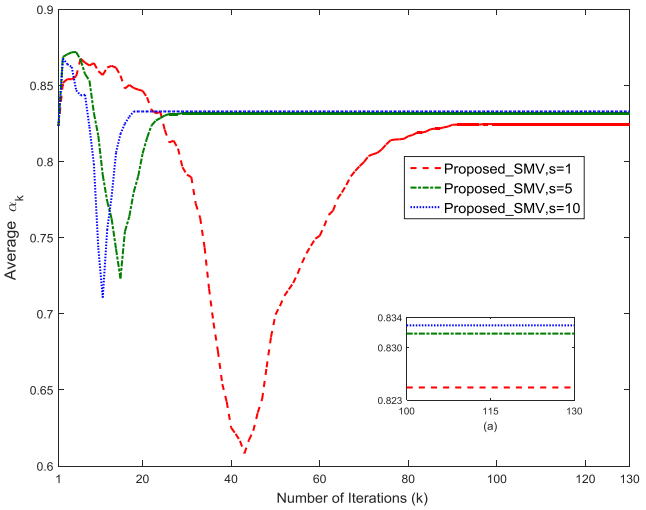


FIGURE 3. The average α_k of the proposed algorithms with different iterations at different step-sizes ($K = 20$, $N = 256$, $M = 130$, $t = 0.5$, Gaussian signal).

Based on the analysis of Figure 1 to Figure 2, we can conclude that, the reconstruction ability of proposed method with larger thresholds is almost identical. Therefore, we randomly select threshold $t = 0.5$ as the default threshold parameter of the proposed algorithm in the following simulations.

Figure 3 shows the relationship between number of iteration and α_k with different step-sizes. In the Figure 1, it shows that when the sparsity level is smaller than 20 and the number of measurement is equal to 130, the reconstruction percentage of the proposed method is 100% for different thresholds and iterative step-sizes. Therefore, we set the sparsity level,

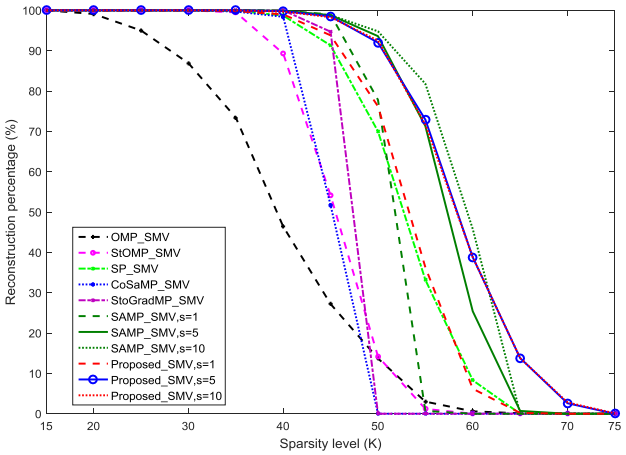


FIGURE 4. Reconstruction percentage of different algorithms with different sparsity at SMV model ($M = 130$, $N = 256$, $t = 0.5$ Gaussian signal).

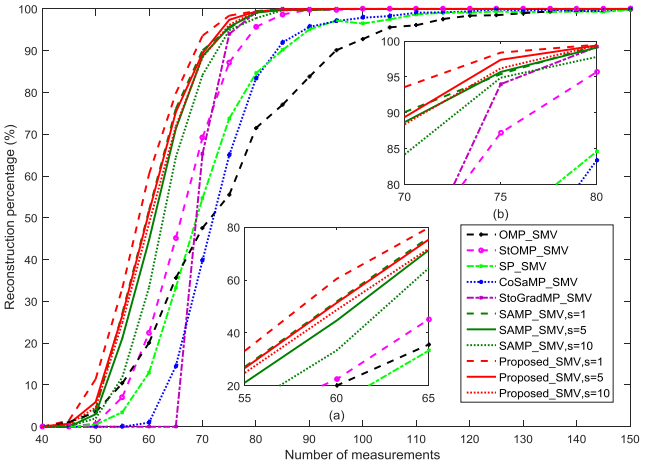


FIGURE 5. Reconstruction percentage of different algorithms with different measurements at SMV model ($K = 20$, $N = 256t = 0.5$ Gaussian signal).

number of measurements and threshold as 20, 130 and 0.5 in the simulation of the Figure 3, respectively. From Figure 3, we can clearly see that, the man range of threshold α_k is from 0.6 to 0.9. In particular, from detail views sub-figure (a) in Figure 3, we can see that, when the number of iterations $100 \leq k \leq 130$, the average α_k of the proposed method is nearly identical for different iterations. Besides, from Figure 3, we can see that, the average α_k of the proposed method has instability for different step-size when the number of iterations is from 1 to 100. This is attributing to the residual instability of the proposed method in reconstruction process. Based on the analysis of Figure 3, we can see that the mean variable range of α_k is 0.6 to 0.9. The explanation about α_k is in (12). It is just a ratio in the iterative procession, and is not fixed value that we set.

In Figure 4, we compare the reconstruction percentage of the proposed algorithm with the OMP [11], StOMP [12], SP [14], CoSaMP [15], StoGradMP [16] and SAMP [17] algorithms for different sparsity levels at the SMV model. We set the length of signal, the number of measurement and the range of sparsity level as $N = 256$, $M = 130$ and $K \in [15\ 75]$, respectively. From Figure 4, we see that, when the sparsity level $K \leq 20$, all of the methods almost achieve higher reconstruction percentages. When $20 \leq K \leq 35$, the reconstruction percentage of the OMP algorithm begins to decline, from approximately 99.1% to 46.4%, while other algorithms still have higher reconstruction percentages. When $35 \leq K$, the reconstruction percentages of all algorithms begin to decrease, except for the StoGradMP and the proposed algorithm with $s = 5$ and $s = 10$. In particular, for $35 \leq K < 40$, the proposed algorithm, SP and CoSaMP algorithms begins to decline, which are from 99.8% to 99%, 99.6% to 98.7% and 99.7% to 98.4%, respectively. When $35 \leq K \leq 50$, the SAMP method with $s = 1$ has higher reconstruction percentage than proposed method with $s = 1$. However, When $50 < K$, the proposed method with $s = 1$ has higher reconstruction percentage than SAMP method with $s = 1$. When $45 \leq K \leq 60$, the SAMP with $s = 5$ has higher reconstruction percentage than proposed method with $s = 5$. However, When $60 < K$, the proposed method with $s = 5$ has higher reconstruction percentage than SAMP method with $s = 5$. When $45 \leq K \leq 65$, the SAMP with $s = 10$ has higher reconstruction percentage than proposed method with $s = 10$. However, When $60 < K$, the proposed method with $s = 10$ has higher reconstruction percentage than SAMP method with $s = 10$. Therefore, we can approximatively regard the performances in reconstruction percentage for the proposed method and SAMP are same, and they have better performances in reconstruction percentage than others.

In Figure 5, we compare the proposed methods with others algorithms for different measurement values under the SMV model. In Figure 4, we see that the reconstruction percentage of all algorithms is 100% when the sparsity is 20. Therefore, we set the sparsity level as 20 in the simulation of Figure 5, and the length of the original signal is keep consistent with the figure 4. From figure 5, we can see that, the proposed method with $s = 1$ has a higher reconstruction percentage than other methods for different measurement sizes, until all methods have 100% reconstruction. From the detail views that are sub figure (a) and sub figure (b) in figure 5, we can see that the proposed method has higher reconstruction percentage than SAMP method with same step size s and other methods.

Base on the analysis of figure 4 to figure 5, the proposed method has better performance in reconstruction percentage than others on the whole.

In Figure 6, we compare the average running time of the proposed algorithms with the OMP, StOMP, SP, CoSaMP, StoGradMP and SAMP algorithms for different sparsity levels at the SMV model. In Figure 4 and Figure 5, we can see that, when the number of the measurements is equal to 130 and the sparsity level is less than or equal to 20, the reconstruction percentage of the OMP algorithm is almost approach to 100%. When the sparsity level $K \leq 40$ and the number of measurements is equal to 150, the reconstruction percentage of the other algorithms are close to 100%. Therefore, we set the sparsity range of the OMP algorithm and the

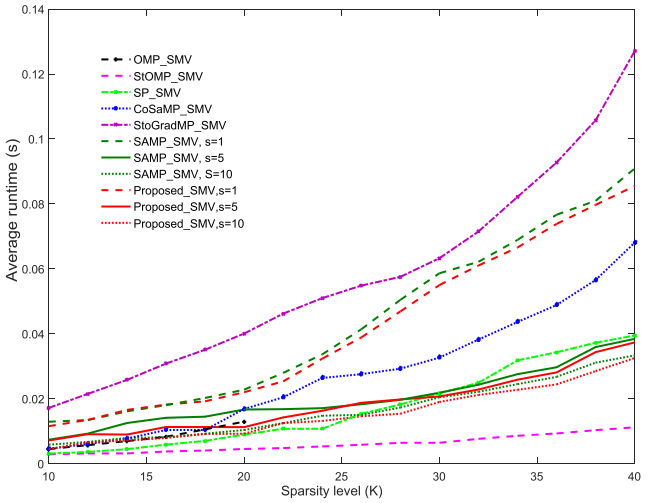


FIGURE 6. The average runtime of different algorithms with different sparsity levels at SMV model and fully reconstruction conditions ($M = 150$, $N = 256$, $t = 0.5$, Gaussian signal).

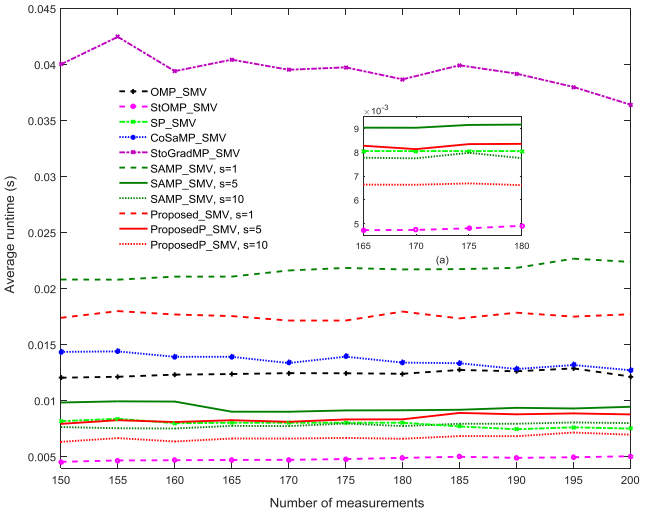


FIGURE 7. The average runtime of different algorithms with different measurements at SMV model ($K = 20$, $N = 256$, Gaussian signal).

other methods as [10 20] and [10 40] in the simulation of the figure 6, and the number of measurements and samples of the signal are 150 and 256, respectively.

From figure 6, the running time of the StoGradMP algorithm is longest, and the next is followed by the SAMP with $s = 1$, the proposed method with $s = 1$ and CoSaMP method. The StOMP method has the shortest running time. The proposed method has shorter running time than SAMP method for the same step-size s. However, the running times of SP, SAMP with $s = 5$ and $s = 10$, proposed method with $s = 5$ and $s = 10$, and OMP method have little difference and are almost the same. Besides, for the proposed method, when $s = 10$ the running time is the shortest, and the next is $s = 5$ and $s = 1$. This means that the larger step-size can reduce the running time of the proposed algorithm with the same sparsity level setting when the original signal is fully reconstructed.

In Figure 7, we compare the average running time of different algorithms for different measurements under the SMV model. From figure 4 and figure 5, we see that the reconstruction percentage is 100% when the sparsity is equal to 20 and the number of the measurements is greater than or equal to 150. Therefore, we set the range of the measurements as [150 200] and sparsity level as 20 in the simulation of the figure 7. The other parameters are consistent with the figure 5.

From figure 7, we can see that the StOMP method has the shortest running time, and the next is followed by is proposed method with $s = 10$ and the SAMP with $s = 10$ and SP method. The StoGradMP has the longest running time. The proposed method has shorter running time than SAMP method for the same step size s. Besides, for the proposed method, when $s = 10$ the running time is the shortest, and the next is $s = 5$ and $s = 1$.

Base on the analysis of figure 6 and 7, the proposed method with $s = 10$ has shorter running time than other methods except for the StOMP. The the proposed methods have shorter running time than SAMP with same step size.

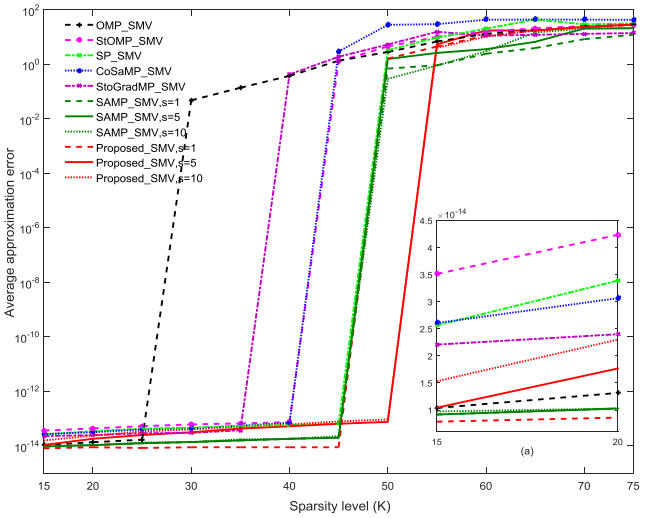


FIGURE 8. The average approximation error of different algorithms with different sparsity at SMV model ($M = 130$, $N = 256$, Gaussian signal).

In Figure 8, we compared the average approximation error of different algorithm with different sparsity levels under the SMV model. In Figure 4, we regard the reconstruction percentage that is higher than 95% as the higher probability reconstruction stage for SMV model. Otherwise, we regard as the lower reconstruction probability stage. The simulation parameters in figure 8 are same in figure 4.

From the sub-figure (a) in figure 8, we can see that, when $15 \leq K \leq 20$, the average approximation error of the proposed method with $s = 1$ is smaller than other algorithms in higher reconstruction probability stage. The average approximation error of the StOMP algorithm is the maximum, which means that the reconstruction accuracy is lower than other algorithms. It attribute to the inaccuracy of the atomic selection in the StOMP algorithm. Besides, in terms of the proposed method, we can see that the reconstruction error of the proposed with $s = 1$ is the lowest, the next

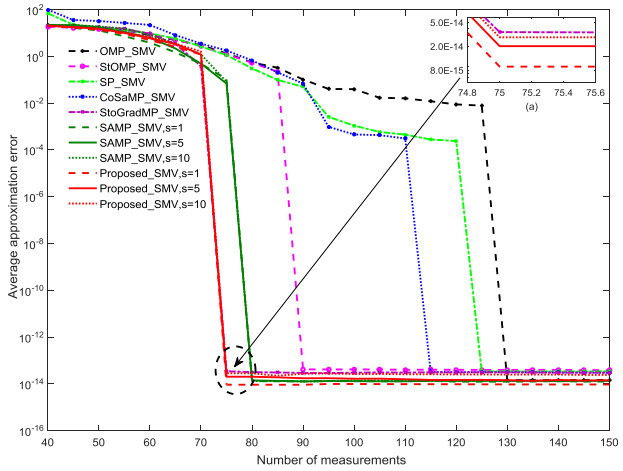


FIGURE 9. The average approximation error of different algorithms with different measurements at SMV model ($K = 20$, $N = 256$, Gaussian signal).

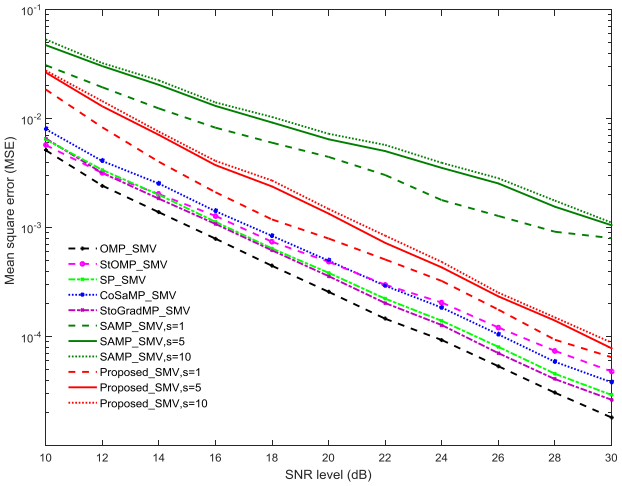


FIGURE 10. The Mean square error (MSE) of different algorithms with different SNR levels ($K = 20$, $N = 256$, $M = 150$, $t = 0.5$, Gaussian signal).

are the proposed method with $s = 5$ and the proposed method with $s = 10$. This indicates that the reconstruction accuracy of the proposed method with smaller step-size is higher than the proposed method with larger step-size. This is mainly because that the smaller step-size can more accurately approach the real sparsity of original signal than the larger step-size, which makes the proposed algorithm have higher reconstruction accuracy. When the sparsity level than 40, all methods cannot reconstruct the signal that can be seen from figure 4, it is no reason to compare the error. Therefore, the proposed method with $s = 1$ has better performance in error than others. The proposed methods with $s = 5$ and $s = 10$ have worse performance in error than SAMP and OMP, and better than the others except for proposed method with $s = 1$.

In Figure 9, we compared the average approximation error of different algorithms with different measurements under the SMV model. The simulation parameters in figure 9 are same in figure 5. From Figure 9, we can see that the proposed method with $s = 1$ has smaller average approximation error than other algorithms. From the sub-figure (a) in figure 10, we can see that, when $74.8 \leq m \leq 75.6$, the reconstruction error of the proposed with $s = 1$ is the smallest, followed by the proposed with $s = 5$, the proposed with $s = 10$ and the StoGradMP algorithm, respectively. This means that the reconstruction accuracy the proposed method with smaller step-size is higher than the other algorithms and the proposed method with larger step-size. With the increasing of measurements, the average approximation errors are almost the same for proposed method with $s = 5$, SAMP and OMP, and are higher than the proposed method with $s = 10$.

Based on the analysis of the figure 8 and 9, the proposed method with $s = 1$ has smaller error than other methods for different sparsity level and measurements. Although OMP has smaller error than proposed method with $s = 10$, it requires larger number of measurements at the same sparsity level. The SAMP methods with different s have almost the same error with proposed method with $s = 5$, and has smaller error with $s = 10$ than proposed method with $s = 10$ for lager measurements, but SAMP with $s = 10$ has larger error than the proposed method with $s = 10$ when $75 \leq M \leq 80$.

In Figure 10, we compared the average square error (*MSE*) of different algorithms with different signal-noise ratio (*SNR*) levels at SMV model. From figure 4 and figure 5 we can see that, when the sparsity level is equal to 20 and the number of measurements is greater than or equal to 150, all of algorithms can achieve fully reconstruction. This is mainly originate from the reconstruction error of the algorithm is unstable during incomplete reconstruction stage, which will affect the reconstruction error of the algorithm in the noise condition. Therefore, we set the sparsity level and the number of measurements as 20 and 150, respectively. We used Gaussian white noise as the noise signal. The range of the *SNR* level is from 10dB to 30dB.

In figure 10, the OMP has the lowest MSE, followed by other methods except for the proposed methods and SAMP methods with different step size. The proposed methods have lower MSE than SAMP methods. These mean that the OMP, StoGradMP, SP, CoSaMP and StOMP have stronger capability against noise disturbance than proposed methods and SAMP methods, and the proposed methods have stronger capability against noise disturbance than SAMP methods. The proposed method with smaller step size has stronger capability against noise disturbance than with larger step size. The reason for proposed method and SAMP method have weaker capability against noise disturbance than other methods, is mainly because that the presence of the noise causes the inaccuracy of the sparsity estimation, which affects the reconstruction accuracy of the proposed algorithm. However, in terms of the practical application of the algorithm, the proposed method is undoubtedly more advantageous. It is because that, except for the SAMP, the other algorithms both requires the sparsity of the original as the prior information.

However, this prior information is unknown in practical application. Although the OMP algorithm can also achieve the reconstruction in an unknown sparsity environment, the number of iteration is equal to the numerical value of the sparsity level. That is the number of iteration of the OMP algorithm is constrained by sparsity level. If we set the iteration stop condition of the OMP algorithm to that when the residual is less than the iteration stop threshold, the iteration is stopped. This may cause an increase in the number of OMP algorithm cycles and the size of the support set, which affects the computational complexity and reconstruction accuracy of the algorithm.

According to the analysis in the section A, on the whole, the proposed method has higher reconstruction percentage. The proposed method with $s = 1$ has smaller error than the others. Although the SAMP method has smaller error than proposed method with $s = 5$ and $s = 10$, the proposed method with $s = 1$ has smaller error than SAMP. The difference is about 1×10^{-14} and is very small and can be ignored. Therefore, we can regard they have the same error. In the aspect of running time, the proposed method and SAMP method have longer running time than some methods. This is because that the proposed method and SAMP method require estimating sparsity, it will consume much time. There is a difference between the estimated sparsity and real sparsity, it also will affect the convergence time. In the noise condition, the proposed method and SAMP method has larger error than other methods. This is because the noise affects the accuracy of estimated sparsity. Besides, the errors are also large in lower SNR for other methods, all method have not realized successfully reconstruction. This is because that the normal compress sensing method has weaker capability against noise disturbance. Therefore, we can conclude that the proposed method have better performance in reconstruction and weaker capability against noise than others for SMV model and practical application.

B. JOINT SPARSE SIGNAL RECONSTRUCTION FOR MMV MODEL

In this subsection, we use the random signal with K -sparse and Gaussian distribution as original signal. The number of original signal is equal to $l = 4$, and the size of the source signal is $N \times l = 256 \times 4$, which also can be regard as a matrix with 256×4. In addition, we ensure that the position of the non-zeros elements of the source signal is the same, that is , the rows of matrix is sparse, and the number of the non-zeros rows are equal to K. The measurement matrix is randomly generated with a Gaussian distribution. We set the iteration step-size of the proposed and SAMP algorithm as $s = 1, 5, 10$. The reconstruction error of all algorithms is $1 \times e^{-6}$. Besides, we set the double iterative halt error of the proposed algorithm as $\varepsilon_2 = 1 \times e^{-5}$ and $\varepsilon_2 = 1 \times e^{-7}$, respectively. It is also noted that these errors are obtained using Frobenius norm operation on the reconstruction error matrix and residual matrix, respectively.

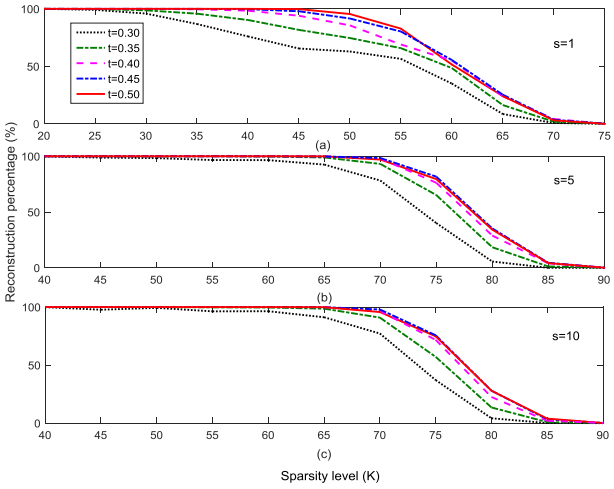


FIGURE 11. Reconstruction percentage of the proposed algorithms with different measurements under the different thresholds and step-sizes ($M = 120$, $N = 256$, $l = 4$, Joint sparse random Gaussian signal).

In Figure 11, we compare the reconstruction percentage of the proposed algorithm with different sparsity under the different thresholds and step-sizes conditions. In MMV model, the range of the sparsity levels is from 20 to 90. The threshold set and the iterative step-size of the proposed algorithm are stills set as $t \in [0.3\ 0.5]$ and $s \in [1, 5, 10]$, respectively. Besides, we set the number of measurements as $M = 120$.

From the figure 11 an figure 12, we can see that the proposed methods with thresholds $t = 0.4$, 0.45 and 0.5 have higher reconstruction percentage than with threshold $t = 0.3$ and 0.35 under the same sparsity level, measurement and step-size s. Due to that proposed methods with thresholds $t = 0.4$, 0.45 and 0.5 have little difference in the performance of reconstruction percentage, we randomly select the threshold $t = 0.5$ as the threshold used in MMV model. The selected threshold is the same with threshold that is used in SMV model.

In Figure 12, we compare the reconstruction percentage of the proposed algorithm with different measurements in different thresholds and step-sizes conditions. We set the sparsity level and the number of measurements as $K = 20$ and $M \in [35\ 100]$, respectively. The other experiment parameters are same with the figure 12.

Figure 13 shows the relationship between number of iteration and α_k with different step-sizes. From figure 11 and figure 12, we can see that when the sparsity level is smaller or equal to 20 and the number of measurements is greater than or equal to 100, the proposed method with different step-sizes can achieve the fully reconstruction. Besides, we also discover that the reconstruction percentage of the proposed with lager step-sizes is very identical. Therefore, we set the sparsity level, number of measurements and threshold as 20, 100 and 0.5 in the simulation of the figure 13.

From Figure 13, we can clearly see that, the variable range of α_k is from 0.6 to 0.95. In particular, from detail views sub-figure (a) in figure 13, we can see that, when the number of iteration $60 \leq k \leq 100$, the average α_k of the

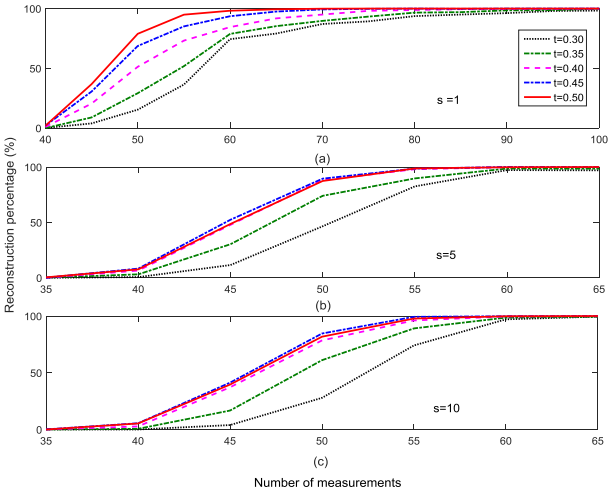


FIGURE 12. Reconstruction percentage of the proposed algorithms with different measurements under the different thresholds and step-sizes ($K = 20$, $N = 256$, $l = 4$, Joint sparse random Gaussian signal).

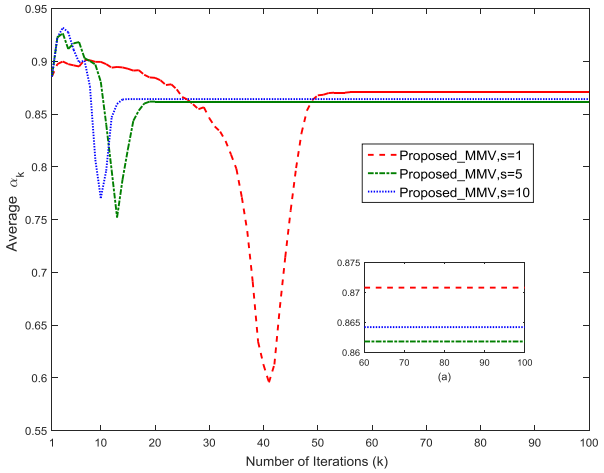


FIGURE 13. The average α_k of the proposed algorithms with different step-sizes at different iterations ($K = 20$, $l = 4$, $N = 256$, $M = 130$, $t = 0.5$, Gaussian signal).

proposed method is nearly identical for different iterations. Besides, from figure 13, we can see that, the average α_k of the proposed method has some instability for different step-size before convergence. This is attributing to the residual instability of the proposed method in reconstruction process.

Based on the analysis of figure 11 and figure 12, we can conclude that, the reconstruction ability of proposed method with larger thresholds is almost identical. Therefore, we randomly select threshold $t = 0.5$ as the default threshold parameter of the proposed algorithm in the following simulations. Based on the analysis of figure 13, we can see that the mean variable range of α_k is 0.6 to 0.95 in MMV model. The explanation about α_k is in (33). It is just a ratio in the iterative procession, and is not fixed value that we set.

In Figure 14, we compare the reconstruction percentage of different algorithms with different sparsity levels for the MMV model. From figure 14, we can see that, when $K < 20$, all of the algorithms almost can achieve higher

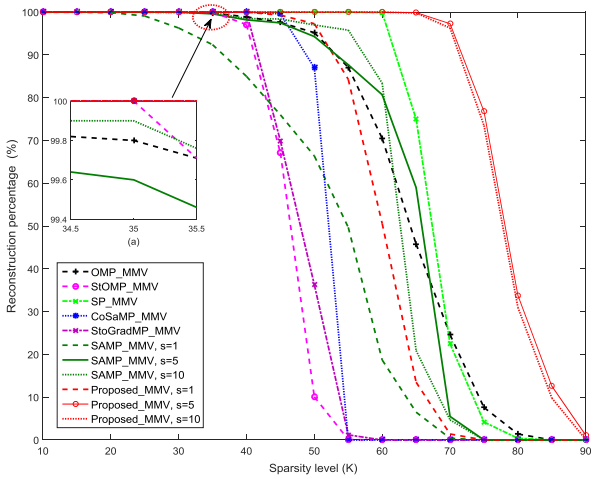


FIGURE 14. Reconstruction percentage of different algorithms with different sparsity levels at the MMV model ($M = 120$, $N = 256$, $l = 4$, $t = 0.5$, Joint sparse random Gaussian signal).

reconstruction percentages. When $20 \leq K \leq 35$, the reconstruction percentage of the SAMP algorithm with $s = 1$ begins to reduce from approximately 99.9% to 92.4%. When $35 \leq K$, the reconstruction percentage of all algorithm begins to reduce except for the StoGradMP, CoSaMP and the proposed algorithms. When $40 \leq K \leq 65$, all algorithm begins to decline, except for the proposed algorithm with $s = 5$ and $s = 10$. When $40 \leq K \leq 50$, the reconstruction percentage of proposed method with $s = 1$ reduces from 100% to 97.3%, but its reconstruction percentage is stills higher than the other algorithms, except for the SP and the proposed method with $s = 5$ and $s = 10$. Besides, for $50 \leq K \leq 65$, although the reconstruction percentage of the SAMP with lager step-sizes ($s = 5$ and $s = 10$) is higher than the proposed algorithm with smaller step-size ($s = 1$), but reconstruction percentage of the proposed method with lager step-sizes is stills higher than the SAMP algorithm with different step-sizes and the other greedy algorithms. When $80 \leq K \leq 90$, the reconstruction percentage of the other algorithms is almost approach to 0%, that is to say, their algorithms are not complete reconstruction of signal. However, the proposed method with larger step-size is still can achieve the reconstruction of signal. Furthermore, in terms of proposed algorithm, we can see that, the reconstruction percentage of the proposed method with larger step-size is higher than the smaller step-sizes under the same sparsity and measurement condition. This means that the proposed method with lager step-size have a better reconstruction capacity than the other algorithms. Therefore, we can conclude that the proposed method with $s = 5$ and $s = 10$ has better performance in reconstruction percentage than others with the same measurements and different sparsity levels.

In figure 15, we compared the reconstruction percentage of different algorithms with different measurements at the MMV model. From figure 14, we can see that the reconstruction percentage of all algorithms can achieve the fully reconstruction when the sparsity level and the number of

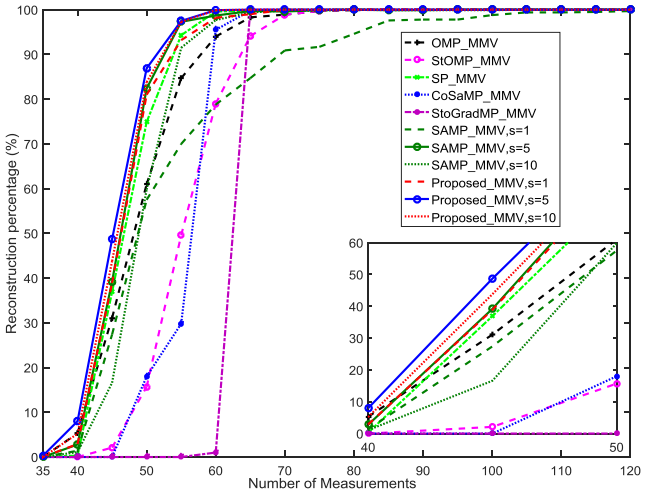


FIGURE 15. Reconstruction percentage of different algorithms with different measurements with the MMV model ($K = 20$, $N = 256$, $l = 4$, $t = 0.5$, Joint sparse random Gaussian signal).

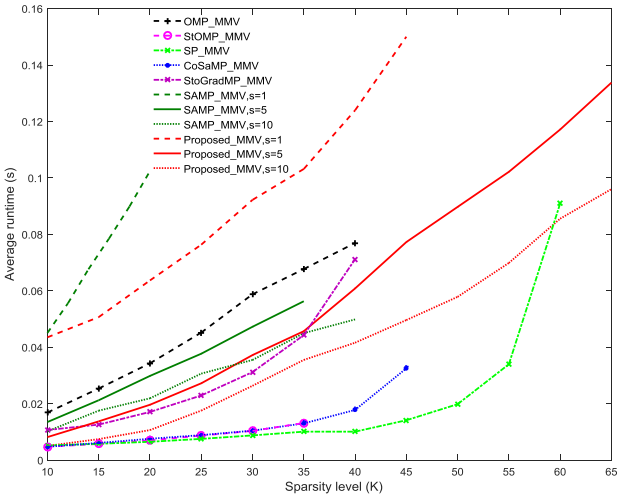


FIGURE 16. Average runtime of different algorithms with different sparsity levels under the fully reconstruction environment ($M = 120$, $N = 256$, $l = 4$, Joint sparse random Gaussian signal).

measurements are equal to 20 and 120, respectively. Therefore, we set the sparsity level and the maximum number of measurement as 20 and 120 in the simulation of figure 15, respectively. The other simulation parameters of figure 15 are same with figure 14.

In figure 15, it shows that the proposed method with $s = 5$ has the highest reconstruction percentage, followed by proposed method with $s = 10$, SAMP with $s = 5$, proposed method with $s = 1$ and others. We can also see that, when the number of measurements is greater than 80, all algorithms can achieve the fully reconstruction, while the SAMP with $s = 1$ is still not complete the fully reconstruction, and until the number of measurements is greater than 120, the SAMP with $s = 1$ can achieve the fully reconstruction. It shows that the SAMP with $s = 1$ requires more measurements to complete the higher probability reconstruction. Therefore, we can conclude that the proposed method with $s = 5$ and $s = 10$ has better performance in reconstruction percentage than others with the same sparsity and different measurements.

Base on the analysis of figure 14 and figure 15, the proposed method with $s = 5$ and $s = 10$ has higher reconstruction percentage than others under the same condition.

In Figure 16, we compare the average runtime of the proposed algorithm with the other greedy algorithms for different sparsity levels under the completed reconstruction environments. From figure 14, we can see that, when the number of measurement is equal to 120 and the sparsity level are less or equal to 20, 35, 40, 45, 60 and 65, the reconstruction percentage of the SAMP with $s = 1$, the SAMP with $s = 5$ and StOMP algorithms, the OMP, StoGradMP and SAMP algorithms with $s = 10$, the proposed with $s = 1$ and CoSaMP algorithms, the SP algorithm, and the proposed method with $s = 5$ and $s = 10$ are almost to approach 100%, respectively. That is, these algorithms can achieve the fully reconstruction. Therefore, we set the different sparsity range to measure the average runtime of different algorithms. The detailed sparsity range can be seen in figure 16.

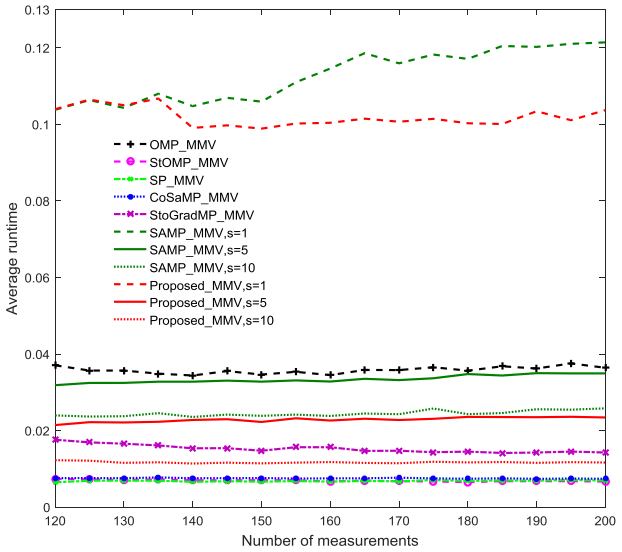


FIGURE 17. Average runtime of different algorithms with different measurements under the fully reconstruction environment ($K = 20$, $N = 256$, $l = 4$, Joint sparse random Gaussian signal).

In figure 17, we compared the average runtime of different greedy algorithms with different measurements under the completed reconstruction environments. From figure 15, we can see that, when the sparsity level is equal to 20 and the number of measurements is greater than 120, the reconstruction percentages of all algorithms can achieve the fully reconstruction. Therefore, we set the sparsity levels and the range of measurements as 20 and [120 200] in the simulation of the figure 17, respectively.

From figure 16 and figure 17, we can see that the proposed method with $s = 10$ has shorter running time than other methods except for SP, StOMP and CoSaMP methods. The proposed method also has lower running time than SAMP methods with the same step size s. We can also see that, the average runtime of the proposed method with $s = 1$ is the

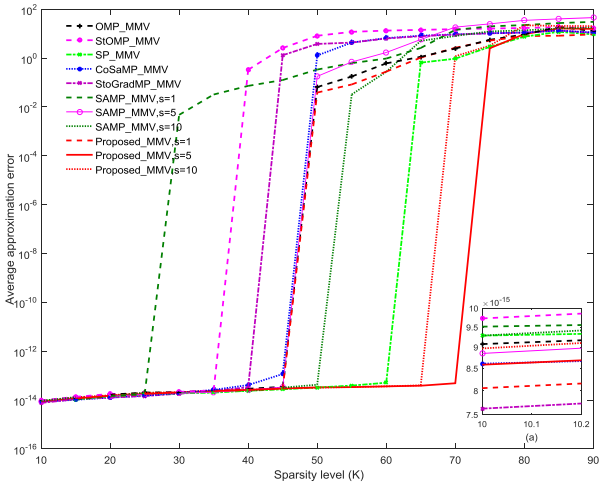


FIGURE 18. The average approximation error of different algorithms with different sparsity levels under the MMV model ($K = 20$, $N = 256$, $l = 4$, Joint sparse random Gaussian signal).

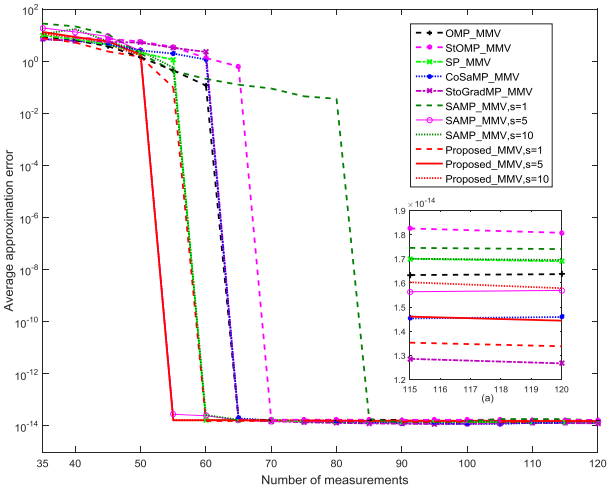


FIGURE 19. The average approximation error of different algorithms with different measurements under the MMV model($K = 20$, $N = 256$, $l = 4$, Gaussian signal).

longest, the next is with $s = 5$ and $s = 10$, respectively. It is indicated that the computation complexity of the proposed with smaller step-size is higher than the proposed with lager step-size under the completed reconstruction condition.

In Figure 18, we compare the average approximation error of different algorithms with different sparsity levels in multiple measurement vectors (MMV) model. Here, we also regard the reconstruction percentage that is higher than 95% as the higher probability reconstruction stage for MMV model. Otherwise, we regard it as the lower probability reconstruction stage. The simulation parameters are the same with the parameters in figure 14. In Figure 19, we compare the average approximation error of different algorithms with different measurements in MMV model. We set the experiment parameters of the figure 19 same as in figure 15.

From figure 18 and 19, when they have smaller error that means they have higher reconstruction percentage, we can see

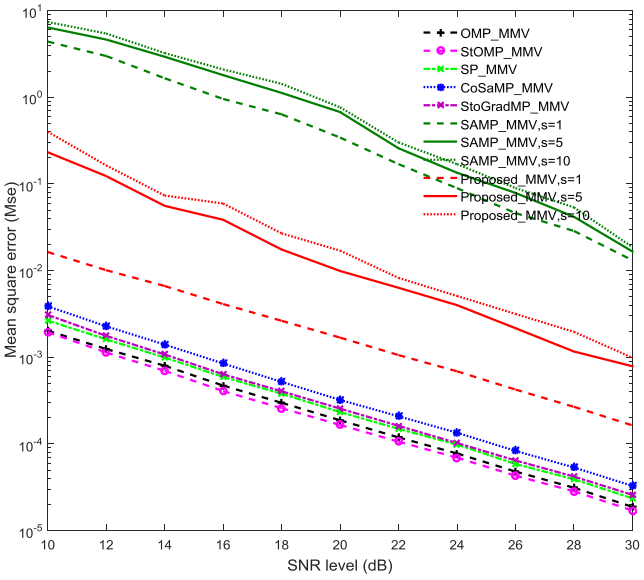


FIGURE 20. Reconstruction percentage of different algorithms with different measurements under the MMV model ($K = 20$, $N = 256$, $l = 4$, $M = 120$, $t = 0.5$ Joint sparse random Gaussian signal).

that the proposed method with $s = 1$ has smaller error than other methods except for the StoGradMP method, and the proposed with $s = 5$ has almost the same error with CoSaMP. The proposed methods with different step size have smaller error than SAMP methods with different step size.

In Figure 20, we compare the mean square error of different algorithms with different *SNR* levels in multiple measurement vectors (MMV) model. From figure 14 and figure 15, we can see that, when the sparsity level is equal to 20 and the number of measurements is greater than or equal to 120, all of algorithms can achieve fully reconstruction. Therefore, we set the sparsity level and the number of measurements as 20 and 120, respectively. We used Gaussian white noise as the noise signal. The range of the *SNR* level is from 10 to 30, and the sample gap is 2.

From Figure 20, we can see that proposed method with different step size has smaller error than SAMP method with different step size, and they have larger error than other methods for different SNR. That means that the proposed method and SAMP method have weaker capability against noise disturbance than others. The reason is that the other methods suppose the sparsity level is known and do not require to estimate sparsity. However, the proposed method and SAMP need estimate sparsity and the noise causes the inaccuracy of the sparsity estimation, which affects the reconstruction accuracy of the proposed algorithm and SAMP method.

According to the analysis in the section B, the proposed method with $s = 5$ and $s = 10$ has higher reconstruction percentage than others under the same sparsity level or measurement, and they require smaller number of measurements to realize 100% reconstruction than the others. Although the StOMP method has smaller error than proposed method, the mean difference between the StOMP method and proposed method with $s = 5$ and $s = 10$ is just about 2×10^{-15}.

The difference is very small and can be ignored. Beside, the difference in reconstruction percentage between the proposed method with $s = 5$ and $s = 10$ and StOMP is very large, and the StOMP almost has the lower reconstruction percentage than other methods. In the aspect of running time, the SP, StOMP and CoSaMP methods have smaller running time than proposed methods and SAMP methods with different step size, and the proposed method has smaller running time than SAMP method. This is because that the proposed method and SAMP method require to estimate sparsity, it will consume much time. There is a difference between the estimated sparsity and real sparsity, it also will affect the convergence time. In the noise condition, the proposed method and SAMP method has larger error than other methods. This is because the noise affects the accuracy of estimated sparsity. Besides, the errors are also large in lower SNR for other methods, all method have not realized successfully reconstruction. This is because that the normal compress sensing method has weaker capability against noise disturbance. Therefore, we can conclude that the proposed method with $s = 10$ and $s = 5$ have better performance in reconstruction and weaker capability against noise disturbance than others for MMV model.

C. 2-D IMAGE SIGNAL RECONSTRUCTION FOR SMV MODEL

In this subsection, we use two test images of size 256×256 as the original images. We regard the image signal as the two-dimensional signal. The test images include the following: Peppers and Cameraman. The sparse basis is a wavelet basis with sparse representation capability, and the size is 256×256. The measurement matrix is randomly generated with a Gaussian distribution, and the size is 180×256. Additionally, we assume that the numerical value of the sparsity level is equal to 30. The iteration halt error of the algorithm set as $1 \times e^{-4}$. The maximum number of iterations is set as the numerical value of the sparsity level, that is, 30. The iteration step-size of the proposed method set as $s = 1, 5, 10$. We use the same threshold that is used in the other simulations, and the ratio of compression is $\delta = 0.7$.

According to the relevant literatures, such as [18]–[22], we can known that when we used the MMV model, we must ensure that the position of the non-zeros elements of joint sparse original signal are the same. The number of the non-zeros rows and its positions are identical in joint sparse signal. However, in the two dimensional image signals, we cannot guarantee the position of the sparse coefficients is the same after the original image signal is sparse represented by sparse basis or over-completed dictionary. Therefore, in this section, we stills use the SMV model to reconstruction 2-D image signal.

We use the Peak Signal to Noise Ratio *PSNR* as an indicator to evaluate the reconstruction quality, which can be expressed as follows:

$$ME = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left| \hat{x}(i,j) - x(i,j) \right|^2 \quad (55)$$

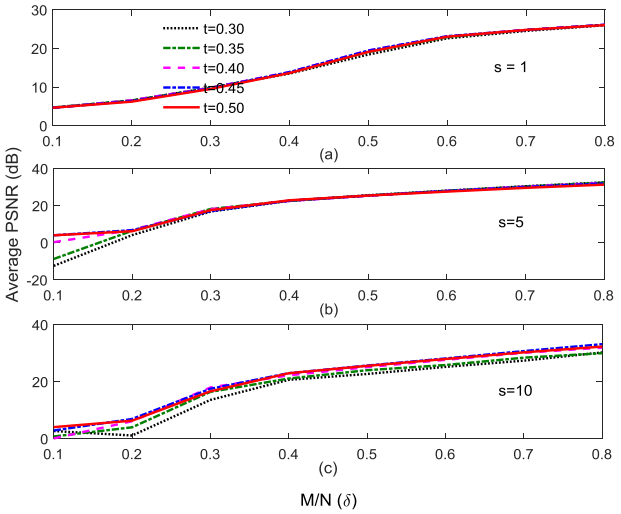


FIGURE 21. The average *PSNR* of proposed algorithm with different thresholds and step-sizes for pepper image ($N = 256$, $K = M/6$).

$$PSNR = 10 \times \lg_{10}\left(\frac{MAX_{\hat{x}}^2}{ME}\right) = 20 \times \lg_{10}\left(\frac{MAX_{\hat{x}}}{\sqrt{ME}}\right) \quad (56)$$

where $M = N = 256$, $\hat{x}(i,j)$ and $x(i,j)$ represents the reconstruction value and original value of the correspondence position, *MSE* is the mean square error and $MAX_{\hat{x}}$ represents the maximum value of the color of the image point. In this paper, because each sample point is represented by 8 bits, $MAX_{\hat{x}} = 255$. The larger the *PSNR*, the higher the reconstruction image quality.

In Figure 21 and Figure 22, we compared the average *PSNR* of the proposed algorithm with different thresholds and step-sizes under the noiseless condition and SMV model. We used the Pepper and Cameraman images as the 2-D original image signal, respectively. We still set the thresholds as $t \in [0.3\ 0.5]$. The iterative step-sizes were set as $s \in [1\ 5\ 10]$. Besides, we used the size of δ to represent the number of measurements. Here, we call the δ as the ratio of compression, that is, the number of measurements vs. the length of original signal. It is represented as M/N. We set the range of the ratio of compression as $\delta \in [0.1\ 0.8]$ and the gap between of the ratio compression is 0.1. The size of the sparsity level was set as the one-sixth of the number of measurements.

From Figures 21 and 22, we can see that the average PSNR is almost the same for $t \in [0.4\ 0.5]$, and the difference in PSNR between them is very little under the same condition. Therefore, we randomly select $t = 0.5$ as threshold in the following experiments.

Figure 23 and figure 24 show that the original image, reconstruction image by different algorithms, respectively. From the figure 23 and figure 24, we can recognize the original image from reconstruction image except for the sub figure (h) and (l). This means that the SAMP with $s = 5$ and $s = 10$ are not successfully to reconstruct the original image, and the others are effective for reconstructing image

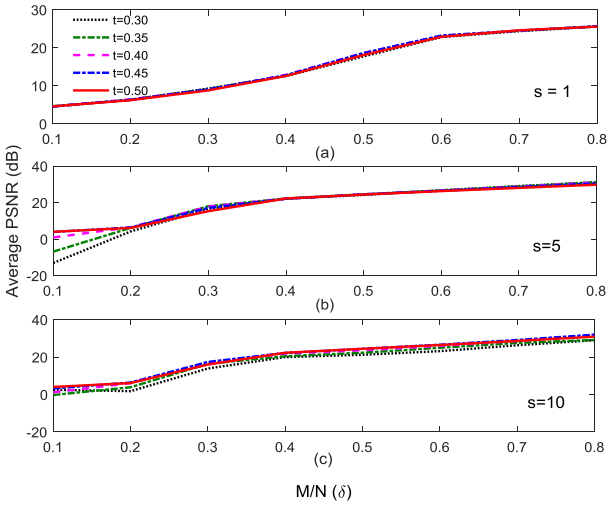


FIGURE 22. The average *PSNR* of proposed algorithm with different thresholds and step-sizes for cameraman image ($N = 256$, $K = M/6$).

TABLE 1. The average *PSNR* of different algorithms for different images signal reconstruction.

Algorithm	Peppers	Cameraman
OMP	27.19/dB	26.10/dB
StOMP	30.08/dB	28.43/dB
SP	28.01/dB	26.97/dB
CoSaMP	28.23/dB	27.24/dB
StoGradMP	23.73/dB	23.02/dB
SAMP, s=1	29.86/dB	28.19/dB
SAMP, s=5	5.72/dB	5.72/dB
SAMP, s=10	5.72/dB	5.72/dB
Proposed, s=1	28.74/dB	27.28/dB
Proposed, s=5	30.45/dB	30.73/dB
Proposed, s=10	31.85/dB	30.96/dB

In Table 1, we compare the average *PSNR* of the proposed algorithm and other algorithms in different test images conditions. From Table I, we can see that the proposed method with $s = 5$ and $s = 10$ have higher *PSNR* than others for Peppers and Cameraman images reconstruction. In particular, the *PSNR* of the proposed method with $s = 10$ are higher 2dB and 2.8dB than SAMP algorithm with $s = 1$, and higher 3.6dB and 3.7dB than CoSaMP algorithm for Peppers and Cameraman images, respectively.

In Table 2, we compare the average running time of different algorithms for Peppers and Cameraman images. From Table 2, we can see that the proposed method with $s = 10$ has the smallest running time. The running time of SAMP with $s = 1$ is seven times more than proposed method with $s = 10$. The running time of CoSaMP is three times more than proposed method with $s = 10$.

FIGURE 23. The reconstruction images of different algorithms under Peppers image condition. (a) Original image; (b) OMP; (c) StOMP; (d) SP; (e)CoSaMP; (f) StoGradMP; (g) SAMP, $s = 1$; (h) SAMP, $s = 5$; (i) SAMP, $s = 10$; (j) Proposed, $s = 1$; (k) Proposed, $s = 5$; (l) Proposed, $s = 10$.

TABLE 2. The average runtime of different algorithms for different images signal reconstruction.

Algorithm	Peppers	Cameraman
OMP	18.57/s	17.56/s
StOMP	13.65/s	14.56/s
SP	21.72/s	21.72/s
CoSaMP	32.27/s	31.74/s
StoGradMP	57.56/s	54.32/s
SAMP, s=1	79.95/s	80.48/s
SAMP, s=5	24.59/s	24.69/s
SAMP, s=10	13.75/s	14.72/s
Proposed, s=1	21.35/s	22.29/s
Proposed, s=5	17.38/s	17.64/s
Proposed, s=10	11.12/s	10.45/s

Base on the analysis of Table 1 and Table 2, the proposed method with $s = 10$ has the highest PSNR and smallest running time than others. The proposed methods with

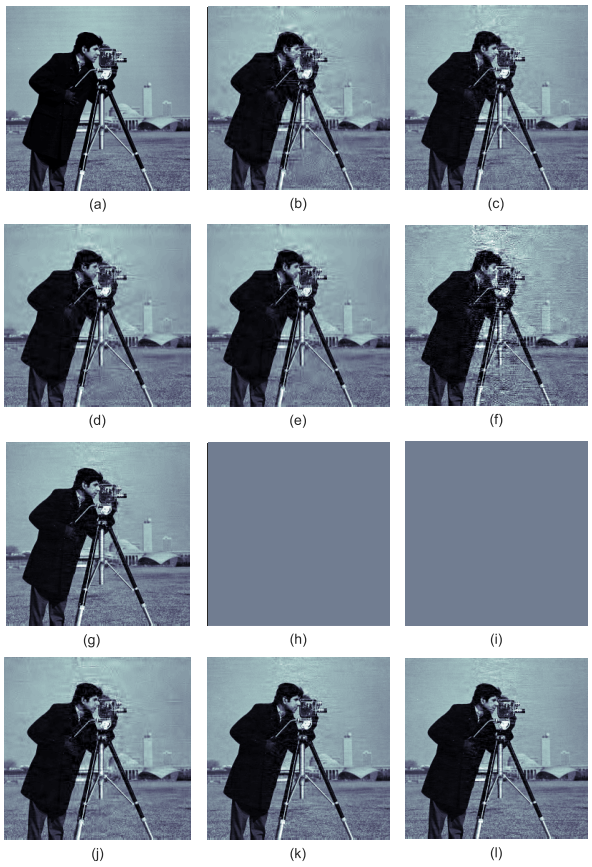


FIGURE 24. Reconstruction images of different algorithms under Cameraman image condition. (a) Original image; (b) OMP; (c) StOMP; (d) SP; (e) CoSaMP; (f) StoGradMP; (g) SAMP, $s = 1$; (h) SAMP, $s = 5$; (i) SAMP, $s = 10$; (j) Proposed, $s = 1$; (k) Proposed, $s = 5$; (l) Proposed, $s = 10$.

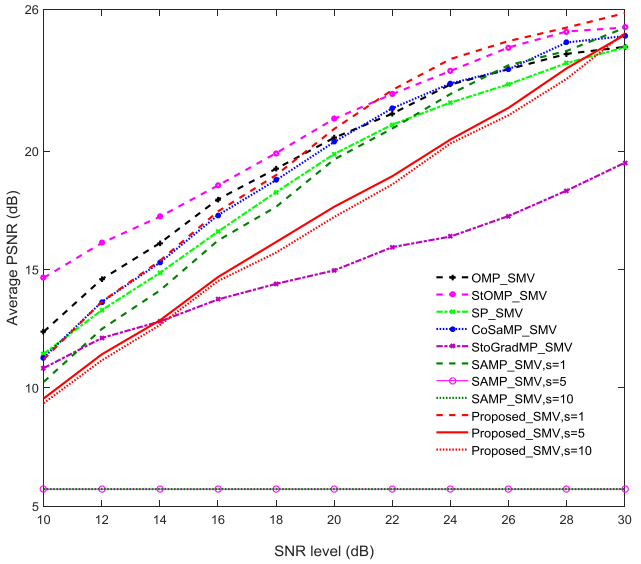


FIGURE 25. Reconstruction percentage of different algorithms with different measurements under the Peppers image condition ($N = 256$, $t = 0.5$, $\delta = M/N = 0.7$, $K = M/6$).

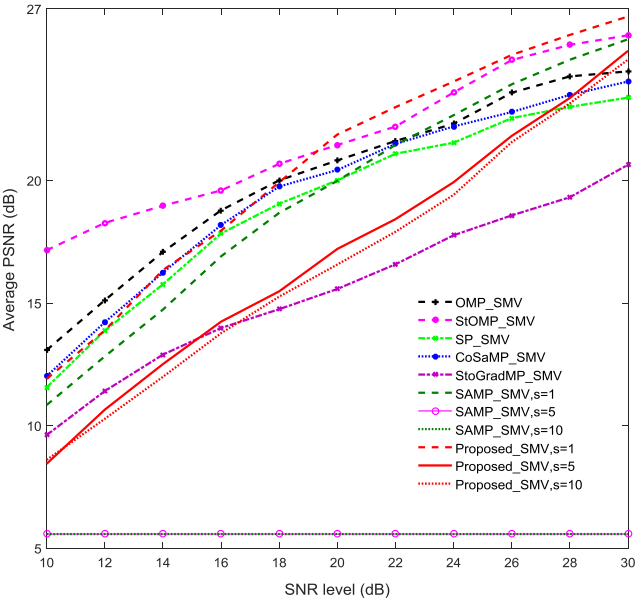


FIGURE 26. Reconstruction percentage of different algorithms with different measurements under the Cameraman image condition ($N = 256$, $t = 0.5$, $\delta = M/N = 0.7$, $K = M/6$).

different step sizes all can reconstruct the two images. However, the SAMP method with $s = 10$ and $s = 5$ cannot reconstruct the two images. Therefore, this shows that the propose method with $s = 10$ has better performance than others for Peppers and Cameraman images reconstruction in noiseless environment and SMV model.

In figure 25 and figure 26, we compare the performance of different algorithms with different SNR levels in single measurement vector (SMV) model and noise environment. We use Gaussian white noise as the noise signal. The range of the *SNR* level is from 10 to 30. The other parameters are the same with the parameters in figure 23 and 24.

From figure 25, we can see that the proposed method with $s = 1$ has larger PSNR than other methods except StOMP and OMP method when the SNR is lower than 22dB. When SNR is higher than 22 dB, the proposed method with $s = 1$ has the highest PSRN. From figure 26, we can see that, the proposed method with $s = 1$ has larger PSNR than other methods except StOMP, CoSaMP, and OMP method when the SNR is lower than 20dB. When SNR is higher than 20 dB, the proposed method with $s = 1$ has the highest PSRN. From the two figures, we can also see that proposed method with $s = 10$ and $s = 5$ have smaller PSNR than others with smaller SNR. However, the proposed method with different step size has faster increase in PSNR than other methods with the increase of SNR. In figure 25, when the SNR is 30dB, the difference in PSNR between proposed method with different step size and StoMP, SAMP with $s = 1$ and CoSaMP is very little. In figure 23, when the SNR is 30dB, the difference in PSNR between proposed method with different step size and StoMP and SAMP with $s = 1$ and is very little.

Base on the analysis of figure 25 and 26, the proposed method with $s = 1$ has better performance in error than other methods except for the StOMP, CoSaMP, and OMP method in

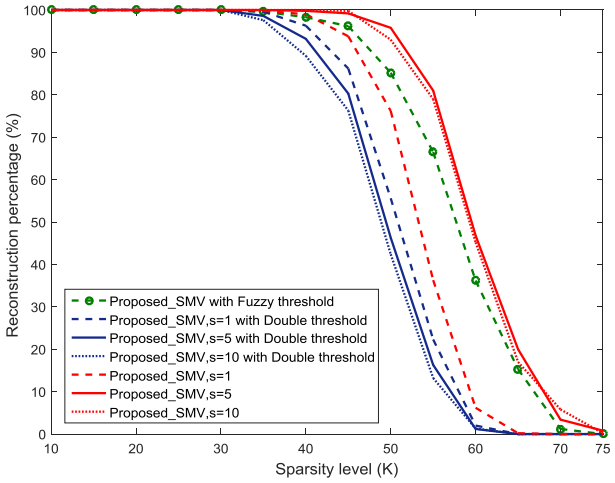


FIGURE 27. Reconstruction percentage of proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with fuzzy threshold and proposed algorithm only with double threshold for different sparsities in the SMV model ($M = 130$, $N = 256$, and $t = 0.5$, Gaussian signal).

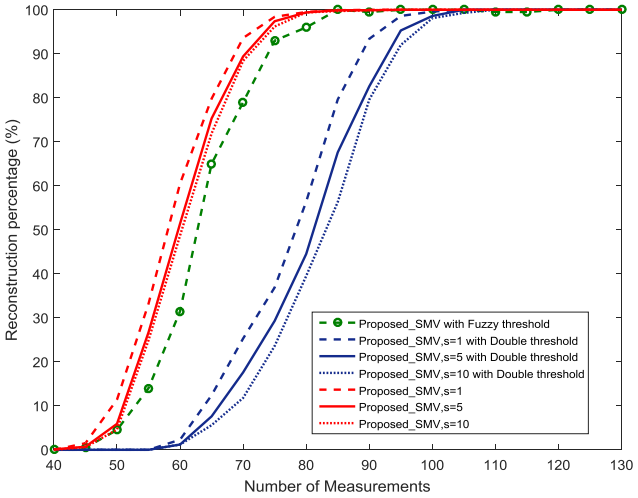


FIGURE 28. Reconstruction percentage of proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with fuzzy threshold and proposed algorithm only with double threshold for different measurements in the SMV model ($K = 20$, $N = 256$, $t = 0.5$ Gaussian signal).

lower SNR, and has the best performance in error than others in higher SNR. The reason that the proposed method has not better performance in lower SNR, is the estimated sparsity is easily affected by the noise. The other methods except for the SMAP method suppose the sparsity is known, they do not require estimated the sparsity. However, the sparsity prior information is unknown in practical application. The proposed method is more suitable for practical application.

D. COMPARISON OF PROPOSED METHOD ONLY WITH FUZZY THRESHOLD AND ONLY WITH DOUBLE THRESHOLD

In figure 27 and figure 28, we compare the reconstruction percentage of the proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with

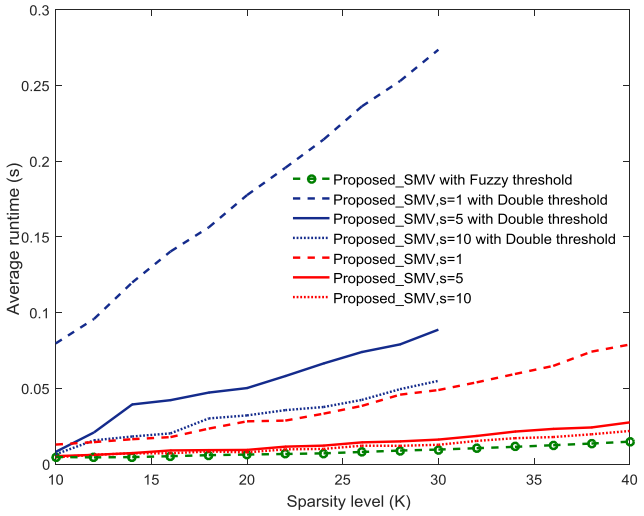


FIGURE 29. The average runtime of proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with fuzzy threshold and proposed algorithm only with double threshold for different sparsity levels in the SMV model under the successful reconstruction environment ($M = 130$, $N = 256$, $t = 0.5$, Gaussian signal).

fuzzy threshold and proposed algorithm only with double threshold in the SMV model, for different sparsity levels and measurements respectively. We set the length of the random signal, the number of measurements and the range of sparsity levels as $N = 256$, $M = 130$ and sparsity from 10 to 75, respectively.

From figure 27 and figure 28, we can see that the proposed method only with fuzzy threshold has higher reconstruction percentage than only with double threshold, and the proposed method both with fuzzy threshold and double threshold has the highest in reconstruction percentage, under the same condition. These show that the fuzzy threshold method has larger contribution in improving reconstruction percentage than the double threshold method in the SMV model.

In figure 29 and figure 30, we compare the average running time of the of the proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with fuzzy threshold and proposed algorithm only with double threshold in the SMV model, for different sparsity levels and measurements respectively.

From the figure 29 and figure 30, we can see that the proposed method only with fuzzy threshold has smallest running time than others. If we only use the double threshold method, it will consume more time to convergence and it cannot reconstruct the original signal successfully when the sparsity is larger than 30 under the conditions that are used in figure 29. These show that the fuzzy threshold method can reduce the computation time and improve the reconstruction performance in the SMV model.These show that the fuzzy threshold method has larger contribution in improving reconstruction percentage than the double threshold method.

In figure 31 and figure 32, we compare the reconstruction percentage of the proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with

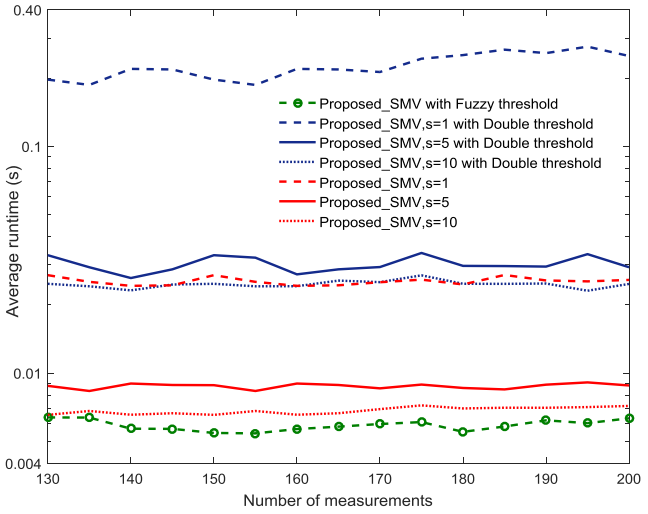


FIGURE 30. The average runtime of proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with fuzzy threshold and proposed algorithm only with double threshold for different measurements in the SMV model under the successful reconstruction environment ($K = 20$, $M = [130\ 200]$, $N = 256$, Gaussian signal).

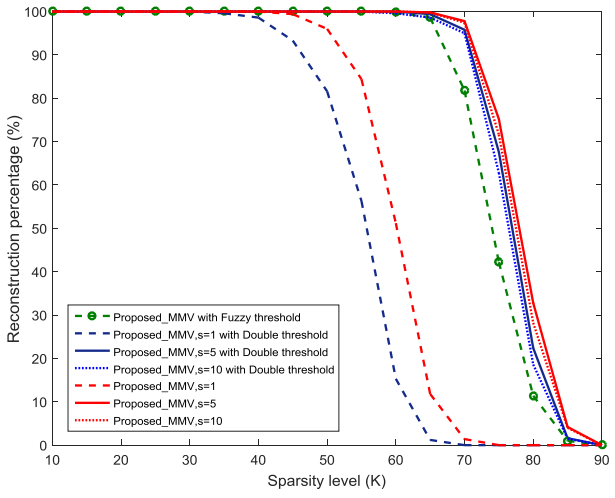


FIGURE 31. Reconstruction percentage of proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with fuzzy threshold and proposed algorithm only with double threshold for different sparsity levels in the MMV model ($M = 120$, $N = 256$, $l = 4$, $K \in [10\ 90]$, $t = 0.5$, joint sparse random Gaussian signal).

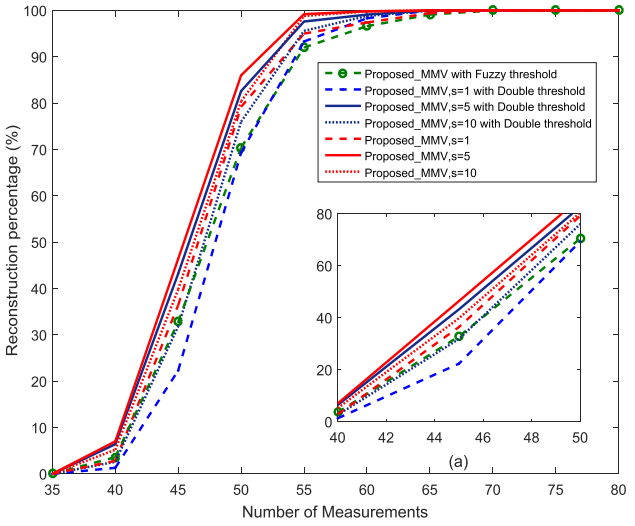


FIGURE 32. Reconstruction percentage of proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with fuzzy threshold and proposed algorithm only with double threshold for different measurements in the MMV model ($K = 20$, $N = 256$, $l = 4$, $M \in [35\ 80]$, $t = 0.5$, Joint sparse random Gaussian signal).

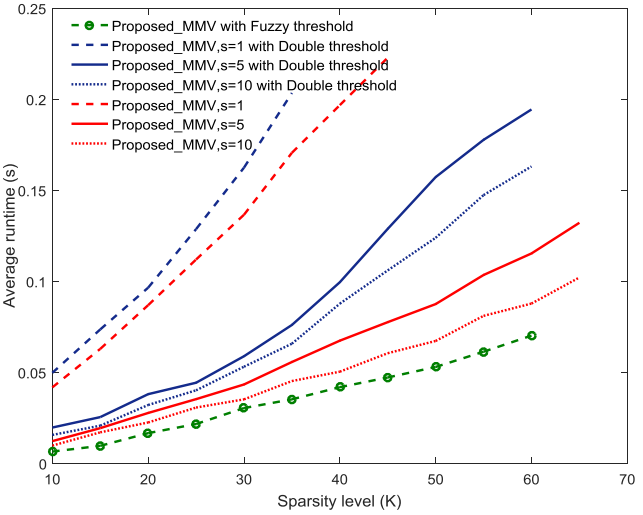


FIGURE 33. Average runtime of proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with fuzzy threshold and proposed algorithm only with double threshold for different measurements under the successful reconstruction environment ($M = 130$, $N = 256$, $l = 4$, $K \in [10\ 70]$, joint sparse random Gaussian signal).

fuzzy threshold and proposed algorithm only with double threshold in the MMV model, for different sparsity levels and measurements respectively. In figure 33 and figure 34, we compare the running time of the proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with fuzzy threshold and proposed algorithm only with double threshold in the MMV model, for different sparsity levels and measurements respectively.

From figure 31 and 32, we can see that the proposed method only with fuzzy threshold has smaller contribution in improving the reconstruction percentage in the MMV model. From the figure 33 and 34, we can see that proposed method only with fuzzy threshold has smallest running time than others. If we only use the double threshold method, it will consume more time to convergence. These show that the fuzzy threshold method can reduce the computation time in the MMV model.

In figure 35 and figure 36, we use the Peppers image as original signal and compare the PSNR and running time of the proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with fuzzy threshold and proposed algorithm only with double threshold, respectively. From the figure 35, we can see that the proposed method only with fuzzy threshold has higher PSNR than only with double threshold, and the proposed method both with fuzzy threshold and double threshold has the highest PSNR most of the time. From the figure 36, we can see that the proposed

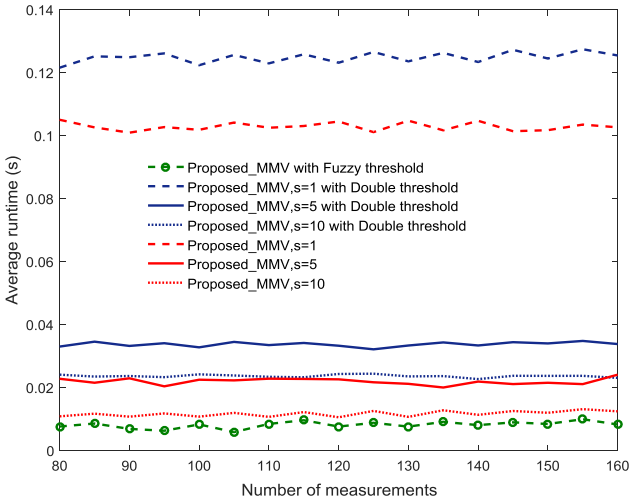


FIGURE 34. Average runtime of proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with fuzzy threshold and proposed algorithm only with double threshold for different measurements under the successful reconstruction environment ($K = 20$, $N = 256$, $l = 4$, $t = 0.5$, $M \in [80\ 160]$, joint sparse random Gaussian signal).

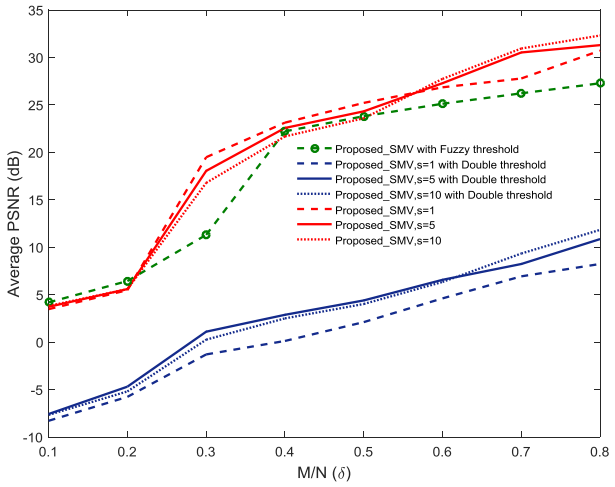


FIGURE 35. The average *PSNR* of proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with fuzzy threshold and proposed algorithm only with double threshold for different δ under the Peppers image and SMV model ($N = 256$, $t = 0.5$, $K = M/6$).

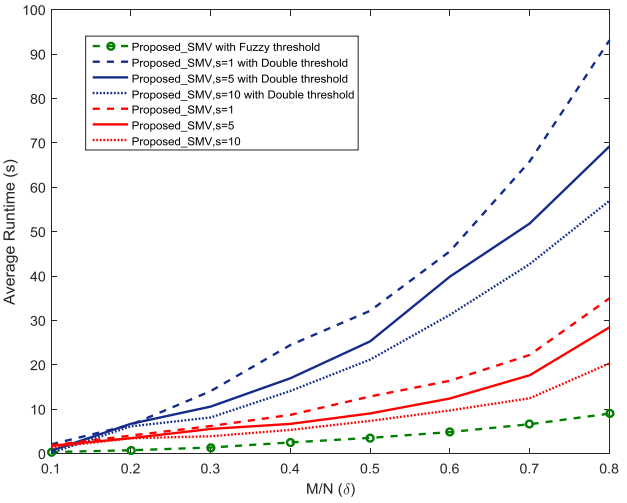


FIGURE 36. The running time of proposed algorithm both with the fuzzy threshold and double threshold, proposed algorithm only with fuzzy threshold and proposed algorithm only with double threshold for different δ under the Peppers image and SMV model ($N = 256$, $t = 0.5$, $K = M/6$).

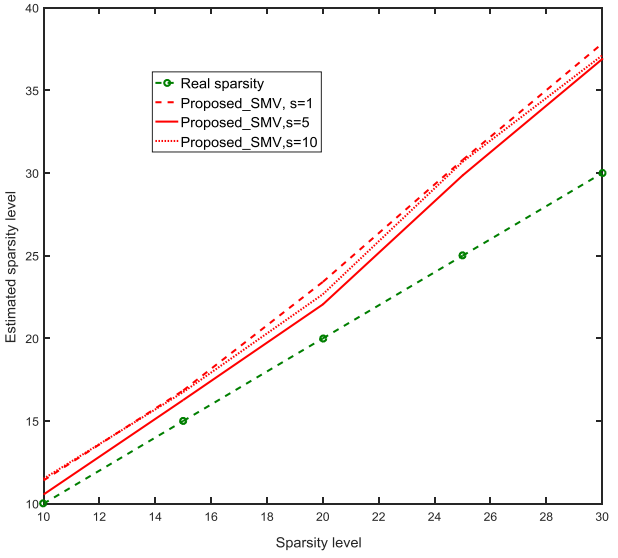


FIGURE 37. Estimated sparsity of proposed method and real sparsity with the same measurements in the SMV model ($M = 130$, $N = 256$, $K \in [10\ 30]$, $s \in [1\ 5\ 10]$, $t = 0.5$ Gaussian signal).

method only with fuzzy threshold has smallest running time than others. If we only use the double threshold method, it will consume more time to convergence. These show that the fuzzy threshold method can reduce the computation time and improve the reconstruction performance for image.

Based on the analysis of section D, it shows that the fuzzy method can reduce the computation time of proposed method.

E. COMPARISON OF ESTIMATED SPARSITY AND REAL SPARSITY

In figure 27, when the sparsity level is smaller than 30, the reconstruction percentages of proposed methods with different step size are all 100%. Therefore, in figure 37, we only compare the estimated sparsity of proposed method and real sparsity with the sparsity range that is 10 to 30. The other parameters are the same with figure 27. For the same reason, based on figure 32, we set the sparsity range that is 10 to 40 in figure 38, and the other parameters are the same with figure 32.

Based on figure 37 and figure 38, we can see that the difference between the estimated sparsity and real sparsity increases with the increase of real sparsity for the same measurements. The difference is smaller for the smaller sparsity. This means the estimated sparsity is more close to the real sparsity. When the value of sparsity is larger, the difference is also larger. If we want to reduce the difference, we can increase the value of measurement. When the measurement is fixed, the larger sparsity is, the larger error is. This is proved by the experiments in section A to section D.

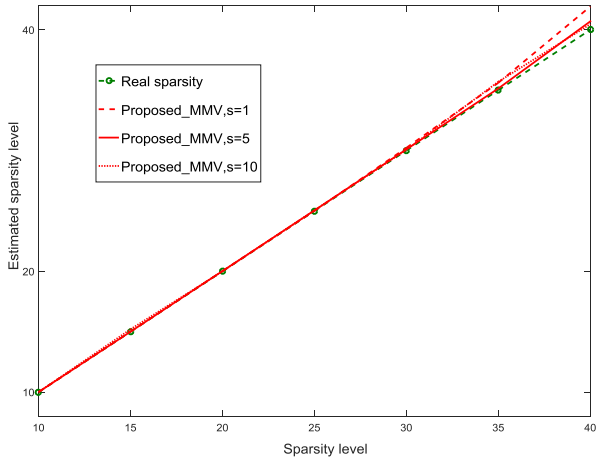


FIGURE 38. Estimated sparsity of proposed method and real sparsity with the same measurements for MMV model ($M = 120$, $N = 256$, $l = 4$, $t = 0.5$, joint sparse random Gaussian signal).

VI. CONCLUSION

In this study, an improved reconstruction method was proposed. The proposed algorithm firstly utilizes the fuzzy threshold method to improve the selections of atoms, avoiding the blind enlargement of the preliminary atomic sets and reducing the computational complexity of the final atomic set selection, thereby improving the reconstruction precision. In addition, the proposed algorithm adopts a double threshold iterative method and a variable step-size method to control convergence condition and adjust the estimated sparsity, which makes the algorithm more accurate to approximate the signal sparsity. This effectively solves the sparsity evaluation problem, and improves the reconstruction precision. Moreover, we extended the proposed method to the MMV case for joint sparse signal reconstruction. The simulation results show that the proposed method has better performance in reconstruction percentage than other greedy methods and does not require sparsity information in SMV and MMV model.

However, the proposed methods will consume more running time than other methods except for SAMP. This is because that the proposed method and SAMP method require estimating and adjusting sparsity, it will consume much time. There is a difference between the estimated sparsity and real sparsity, it also will affect the convergence time. Although the proposed method has weaker capability against noise disturbance, the errors are larger for all methods with lower SNR and the original signal is not successfully reconstructed by all methods. In the future, we will research how to improve the capability against noise disturbance for compressed sensing.

REFERENCES

[1] H. E. A. Laue, "Demystifying compressive sensing [lecture notes]," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 171–176, Jul. 2017. doi: 10.1109/MSP.2017.2693649.

[2] M. Brajović, I. Orović, M. Daković, and S. Stanković, "Gradient-based signal reconstruction algorithm in Hermite transform domain," *Electron. Lett.*, vol. 52, no. 1, pp. 41–43, Aug. 2016. doi: 10.1049/el.2015.1700.

[3] T. Sun, B. Roberto, and H. Jiang, "Local linear convergence of a primal-dual algorithm for the augmented convex models," *J. Sci. Comput.*, vol. 69, no. 3, pp. 1301–1315, Dec. 2016. doi: 10.1007/s10915-016-0235-4.

[4] M. Wang, X. Wu, W. Jing, and X. He, "Reconstruction algorithm using exact tree projection for tree-structured compressive sensing," *IET Signal Process.*, vol. 10, no. 5, pp. 566–573, Jun. 2016. doi: 10.1049/iet-spr.2015.0351.

[5] S. Huang and T. D. Tran, "Sparse signal recovery via generalized entropy functions minimization," *IEEE Trans. Signal Process.*, vol. 67, no. 5, pp. 1322–1337, Mar. 2019. doi: 10.1109/TSP.2018.2889951.

[6] H. Li and G. Liu, "Perturbation analysis of signal space fast iterative hard thresholding with redundant dictionaries," *IET Signal Process.*, vol. 11, no. 4, pp. 462–468, Jun. 2017. doi: 10.1049/iet-spr.2015.0366.

[7] H. Gan, S. Xiao, and Y. Zhao, "A novel secure data transmission scheme using chaotic compressed sensing," *IEEE Access*, vol. 6, pp. 4587–4598, 2018. doi: 10.1109/ACCESS.2017.2780323.

[8] Q. Shi, H. Lu, and Y.-M. Cheung, "Rank-one matrix completion with automatic rank estimation via l1-norm regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4744–4757, Oct. 2018. doi: 10.1109/TNNLS.2017.2766160.

[9] K. Li, T. C. Chandrasekera, Y. Li, and D. J. Holland, "A non-linear reweighted total variation image reconstruction algorithm for electrical capacitance tomography," *IEEE Sensors J.*, vol. 18, no. 12, pp. 5049–5057, Jun. 2018. doi: 10.1109/JSEN.2018.2827318.

[10] B. Bouchhima, R. Amara, and M. T. Hadj-Alouane, "Perceptual orthogonal matching pursuit for speech sparse modelling," *Electron. Lett.*, vol. 53, no. 21, pp. 1431–1433, Dec. 2017. doi: 10.1049/el.2017.1608.

[11] S. K. Sahoo and A. Makur, "Signal recovery from random measurements via extended orthogonal matching pursuit," *IEEE Trans. Signal Process.*, vol. 63, no. 10, pp. 2572–2581, May 2015. doi: 10.1109/TSP.2015.2413384.

[12] V. Meena and G. Abhilash, "Robust recovery algorithm for compressed sensing in the presence of noise," *IET Signal Process.*, vol. 10, no. 3, pp. 227–236, May 2016. doi: 10.1049/iet-spr.2015.0067.

[13] L. Pei, H. Jiang, and M. Li, "Weighted double-backtracking matching pursuit for block-sparse reconstruction," *IET Signal Process.*, vol. 10, no. 8, pp. 930–935, Oct. 2016. doi: 10.1049/iet-spr.2016.0036.

[14] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009. doi: 10.1109/TIT.2009.2016006.

[15] M. A. Davenport, D. Needell, and M. B. Wakin, "Signal space CoSaMP for sparse recovery with redundant dictionaries," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6820–6829, Oct. 2013. doi: 10.1109/TIT.2013.2273491.

[16] N. Nguyen, D. Needell, and T. Woolf, "Linear convergence of stochastic iterative greedy algorithms with sparse constraints," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 6869–6895, Nov. 2017. doi: 10.1109/TIT.2017.2749330.

[17] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," in *Proc. 42nd Asilomar Conf. Signals, Syst. Comput.*, Oct. 2008, pp. 581–587. doi: 10.1109/ACSSC.2008.5074472.

[18] J. Chen and X. Ho, "Theoretical results on sparse representations of multiple-measurement vectors," *IEEE Trans. Signal Process.*, vol. 54, no. 12, pp. 4634–4643, Dec. 2006. doi: 10.1109/TSP.2006.881263.

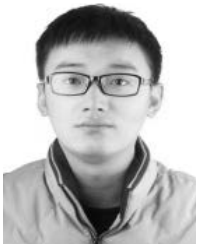
[19] K. Lee, Y. Bresler, and M. Junge, "Subspace methods for joint sparse recovery," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3613–3641, Jun. 2012. doi: 10.1109/TIT.2012.2189196.

[20] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 912–926, Sep. 2011. doi: 10.1109/JSTSP.2011.2159773.

[21] J. D. Blanchard, M. Cermak, D. Hanle, Y. Jing, "Greedy algorithms for joint sparse recovery," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1694–1704, Apr. 2014. doi: 10.1109/TSP.2014.2301980.

[22] A. Besson, D. Perdios, Y. Wiaux, and J.-P. Thiran, "Joint sparsity with partially known support and application to ultrasound imaging," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 84–88, Jan. 2019. doi: 10.1109/LSP.2018.2880571.

[23] H. Li and J. Wen, "A new analysis for support recovery with block orthogonal matching pursuit," *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 247–251, Feb. 2019. doi: 10.1109/lsp.2018.2885919.



HU YUNFENG was born in Wuhu, Anhui, China, in 1993. He received the bachelor's degree in electronic information engineering from the Anhui Institute of Information Technology, Wuhu, in 2016. He is currently pursuing the master's degree with the School of Electrical Engineering, Northeast Electric Power University, Jilin, China. His research interests include compressed sensing, convex optimization, signal processing, and wireless communication.

ZHAO LIQUAN was born in Heilongjiang, in 1982. He received the B.S. degree in electrical and information engineering and the Ph.D. degree in communication and information system from the Harbin University of Science and Technology, Harbin, China, in 2005 and 2009, respectively. Since 2009, he has been an Associate Professor with Northeast Electric Power University, Jilin, China. His research interests include compressed sensing and blind source separation.