



ARTICLE

A High-Quality Adaptive Video Reconstruction Optimization Method Based on Compressed Sensing

YanJun Zhang¹, Yongqiang He², Jingbo Zhang¹, Yaru Zhao³, Zhihua Cui^{1,*} and Wensheng Zhang⁴

¹School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, 030024, China

²Department of Big Data and Intelligent Engineering, Shanxi Institute of Technology, Yangquan, 045000, China

³School of Computer Science, Beijing University of Technology, Beijing, 100124, China

⁴The Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, 100049, China

*Corresponding Author: Zhihua Cui. Email: cuizhihua@tyust.edu.cn

Received: 01 August 2022 Accepted: 21 December 2022

ABSTRACT

The video compression sensing method based on multi hypothesis has attracted extensive attention in the research of video codec with limited resources. However, the formation of high-quality prediction blocks in the multi hypothesis prediction stage is a challenging task. To resolve this problem, this paper constructs a novel compressed sensing-based high-quality adaptive video reconstruction optimization method. It mainly includes the optimization of prediction blocks (OPBS), the selection of search windows and the use of neighborhood information. Specifically, the OPBS consists of two parts: the selection of blocks and the optimization of prediction blocks. We combine the high-quality optimization reconstruction of foreground block with the residual reconstruction of the background block to improve the overall reconstruction effect of the video sequence. In addition, most of the existing methods based on predictive residual reconstruction ignore the impact of search windows and reference frames on performance. Therefore, Block-level search window (BSW) is constructed to cover the position of the optimal hypothesis block as much as possible. To maximize the availability of reference frames, Nearby reference frame information (NRFI) is designed to reconstruct the current block. The proposed method effectively suppresses the influence of the fluctuation of the prediction block on reconstruction and improves the reconstruction performance. Experimental results show that the proposed compressed sensing-based high-quality adaptive video reconstruction optimization method significantly improves the reconstruction performance in both objective and supervisor quality.

KEYWORDS

Compressed sensing; OPBS; block-level search window; nearby reference frame information; evolutionary algorithm

1 Introduction

The conventional video codecs use compression standards such as H.264/AVC, HEVC, etc. Their complexity at the encoding end is generally about 5~10 times higher than at the decoding end.



However, wireless video terminals such as portable terminals, wireless video detection heads, and wireless multimedia sensor networks (WMSN) usually encode video *in situ*. These encoding devices have minimal processing ability. Thus, an encoding algorithm is required to simplify the encoder and have good compression efficiency.

Compressed sensing (CS) [1] is a theory of signal processing. And the emergence of the theory alleviates the huge pressure of signal sampling [2], transmission [3,4], and storage caused by people's high demand for information. It points out that for a compressible signal, collecting only a small number of observations contains enough information to recover the original data. The CS theory can effectively avoid the huge waste of sampling resources. Simultaneously, it greatly reduces the encoder complexity and enables low-cost video acquisition [5] and compression [6,7] in resource-constrained environments. Therefore, CS theory has great application prospects.

At present, the theory has been applied in image transmission [8], image compression [9], fault diagnosis [10], particularly in video encoding [11,12] and decoding [13]. Because of its "compression-as-sampling" feature, CS has unparalleled advantages over traditional video signal acquisition, compression and transmission. Compared to image signals, video signals are essentially a collection of image signals. Therefore, video frames not only have all the characteristics of image signals, but also have a spatial correlation between frames. Furthermore, the Internet is changing people's behavioral habits. Particularly, online video is taking on an increasingly dominant role in the new era of media. The user demand for high-definition video image continues to grow. Hence, the improvement of reconstruction quality is a steady-state topic in the field of video codecs.

The overall procedure of video codecs consists of an encoder and a decoder. The operating principle of the end-to-end video codec is shown in Fig. 1. First, the video is compressed and encoded by the content provider and transmitted over the wireless network to the server; Second, encoded video is stored and transmitted by the server and sent to the users. Finally, the consumer decodes it and then it is ready to play.

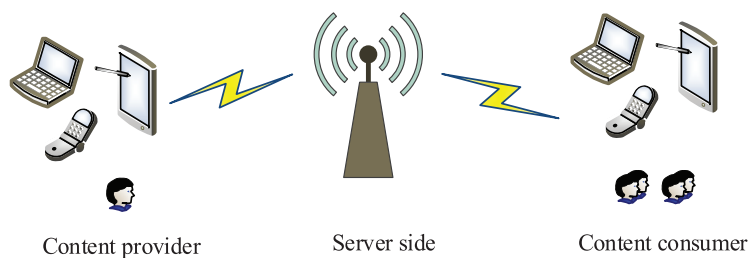


Figure 1: The architecture of end-to-end video codecs

Generally, innovations on the coding side are investigating sampling methods. The research on how to reduce the amount of data transmitted is the focus of the coding side. However, the best possible reconstruction quality is a stable topic on the decoding side. Hence, to enhance the reconstruction performance of video frames, this paper proposes a CS-based high-quality adaptive video reconstruction (CS-HAVR) optimization method. The following are the contributions of the paper:

1. A novel single objective OPBS-PSO model is constructed to realize the optimization of prediction blocks. This paper combines the high-quality optimization reconstruction of foreground block with the multiple hypothesis prediction reconstruction of the background block to improve the overall reconstruction effect of the video sequence.

2. To maximize the usability of reference frames, nearby reference frame information is designed to improve the precision of prediction blocks. Furthermore, block-level search window is proposed to cover the position of the optimal hypothesis block as much as possible.
3. Extensive experimental results prove that our designed model improves reconstruction performance in both objective and subjective quality.

The organizational framework for the remaining sections of this paper is set out below. The research works of relevant scholars are presented in [Section 2](#); [Section 3](#) describes the proposed CS-HAVR optimization method in detail; Extensive experiments are conducted in [Section 4](#), which compares the paper's approach with three methods to prove the validity of the proposed strategy; [Section 5](#) gives a conclusion.

2 Related Work

The high-quality reconstructed image has been a hot topic in video reconstruction. Hence, a wide variety of methods are created. It mainly includes the adjustment of the sampling strategy and the design of the reconstruction method. They all affect the quality and performance of the video reconstruction.

As a sampling operation in CS, various method is usually designed to decrease the number of sampling and increase the reconstruction precision. For this reason, Li [14] presented a method which is block-based, with different measurement matrix and sampling rate for each block. To further improve reconstruction performance, an improved block-based compression sensing (BCS) technology was proposed in [15] to perform operations such as rotation and weighting on the perceptual matrix corresponding to each block. Belyaev et al. [16] conducted the study on block size and proposed a heuristic approach. It randomly selected block size in each iteration. Similarly for block size adjustment, Heng et al. [17] used a fuzzy logic system (FLS) to resize each corresponding block. Xu et al. [18] proposed the idea of combining the adaptive sample rate allocation method with background subtraction to improve the overall reconstruction of video sequences. The algorithm has good reconstruction quality and efficiency in the underground coal mine application scenario. Unde et al. [19] utilized the high redundancy between consecutive video frames to set different sample rates for different blocks in non-key frames, thereby reducing transmission costs. In addition, greedy iterative reconstruction methods, such as pre-GOMP [20], which find the optimal match by iteratively solving the least squares problem. However, the method suffers from iterative, complex, and more time-consuming drawbacks. Therefore, a non-iterative recovery method was developed by Poovathy et al. [21] based on splitting, processing and merging techniques, and it reduces the complexity of the algorithm to a certain extent. Das [22] presented a method which adopted the sparse Tucker tensor decomposition in the same year. In response to this issue, plenty of scholars have studied different sampling strategies [23]. Thus, a uniform sampling rate is used for each block in this paper to reduce complexity on the coding side.

Reconstruction methods [24,25] based on multiple hypothesis prediction [26] have attracted the attention of scholars. However, how to assign the weights precisely has been a headache for academics. In response to the problem, a new reconstruction method was presented by Zheng et al. [27] to get a better prediction block. It mainly consists of two parts: obtaining the hypothesis set and predicting the weights. However, it is still to be further investigated to optimize the hypothesis set. Video Compression Sensing (VCS) aims to perceive and recover scene video in a space-time aware manner. To make better use of temporal and spatial correlations, a hybrid three-dimensional residual block consisting of

pseudo- and true three-dimensional sub-blocks is proposed by Zhao et al. [28]. Ebrahim et al. [29] used the redundancy of keyframes and non-keyframes to generate side information in the reconstruction of non-keyframes and proposed the CS-based Joint Decoding (JD) framework. Similarly, in order to explore interframe correlation, Shi et al. [30] proposed a multi-level feature compensation method using convolutional neural networks (CNN). It allows better exploration of intra-frame and inter-frame correlations. To better use picture information in the spatial-temporal domain, distributed compressed video sensing (DCVS) was proposed [31]. It divides frames into key frames and non-key frames, and uses block-based compressed sensing (BCS) to sample each frame. Cloud-based video image uploads are more common in our daily life. The upload speed is related to the quality of the communication channel, where the amount of data uploaded can be higher if the quality of the communication channel is high and lower if not. Zheng et al. [32] developed a terminal-to-cloud video upload system. They determined the amount of data to be uploaded based on the quality of the channel. Furthermore, a multi-reference frame strategy is proposed to improve the reconstruction. Zhao et al. [33] proposed a new algorithm for efficient reconstruction of video from CS measurements based on the idea of predictive-residual reconstruction. However, it did not consider the effect of the predicted blocks on the reconstruction. The blocks that are more similar to the current block are simply treated as hypothetical blocks. Thus, we conducted studies on the optimization of prediction blocks and designed block-level search window and nearby reference frame information in our paper. The experimental results demonstrate the influence of the prediction block on the final reconfiguration. Our previous research has been on hyperspectral image reconstruction methods [34], while video reconstruction is addressed in this study.

3 The Proposed CS-HAVR Method

The proposed method of CS-HAVR is illustrated in Fig. 2. It consists of the encoding process and the decoding process. First, video sequences are grouped. There are one key frame and seven non-key frames in each group. Second, a group of pictures enter the coding process. Key frame uses the BCS method. Similarly, non-key frames also use BCS to complete the encoding process. The difference is that the sample ratio of key frame is larger compared to the sample ratio of non-key frame. Particularly, Nearby reference frame information is designed to provide more useful information during the reconstruction of non-key frames. Third, the encoded image enters the decoder side via the wireless channel. For key frames, BCS-smoothed projected landweber based on multi-hypothesis (MH-BCS-SPL) is used to obtain the reconstructed values. In the reconstruction process of non-keyframes, a two-stage reconstruction strategy is used because of the low sampling rate for non-keyframes. It is divided into initial reconstruction and secondary reconstruction. For non-key frames, block-level search window and optimization of prediction blocks are constructed to complete secondary reconstruction in this paper.

3.1 Encoding Process

During the encoding stage, original frames are divided into several group of pictures (GOPs). In general, there is a key frame in the GOP, and the others are non-key frames. The structure of the GOP is shown in Fig. 3.

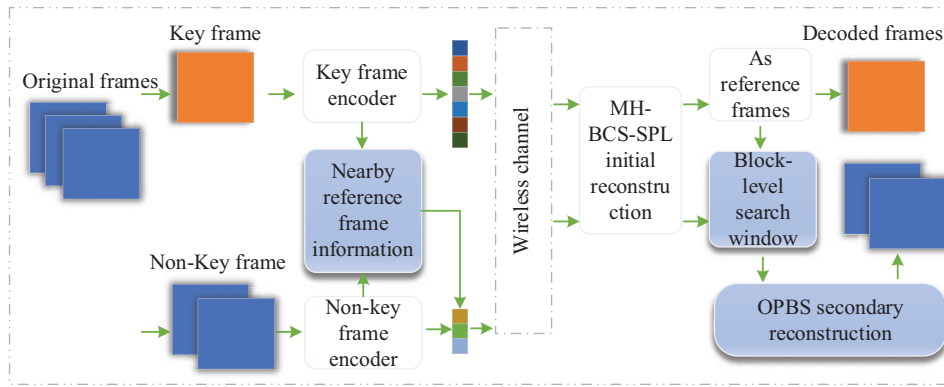


Figure 2: The overall framework of the CS-HAVR method

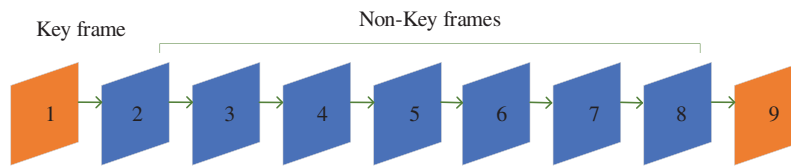


Figure 3: The structure of the GOP

Fig. 3 indicates that the first frame (orange noted) is key frame, and the second to eighth frames (blue noted) are non-key frames. Next, all frames are compressed. Key frames and non-key frames use different compression rates. Generally, the compression rate of key frames is higher due to the fact that the key frame is used as a reference frame for other frames during the reconstruction. In addition, the block compression sensing (BCS) is used in encoding stage. The calculation method of the compression process is shown in Eq. (1).

$$y_{h,j} = \Phi t_{h,j} \tag{1}$$

where, $\Phi \in R^{P \times Q} (P \ll Q)$ represents the observation matrix, the random Gaussian observation matrix is applied in this study. P represents the number of sampling values in y_h . t_h represents the h -th frame. $t_{h,j}$ represents the j -th block of the h -th frame. Accordingly, $y_{h,j}$ represents the projection of the j -th block of the h -th original frame. Through Eq. (1), each frame completes its encoding process.

3.1.1 Nearby Reference Frame Information (NRFI)

The multi-hypothesis prediction phase plays a key role in prediction-residual reconstruction, which refers to finding the best set of matching blocks in the reference frame. The selection of the reference frame has a big impact on the reconstruction of the current block, and this study uses the near-neighbor information so that the current frames all have the opportunity to use the information of the neighboring frames.

For Non-key frame encoder, NRFI is designed to make full use of near-neighbor frame information in our paper. The similarity between neighboring frames is relatively higher due to video sequence consisting of consecutive frames. It can be expressed in Table 1. The horizontal coordinate represents the frame number of the reference frame. The vertical coordinate represents the frame number of the current frame. The bolded font represents the most similar frame to the current frame. For example, the two frames most similar to the 6th frame are the 5th and 7th frames with SSIM values of 0.68

and 0.72, respectively. Furthermore, structural similarity (SSIM) is used to calculate the similarity of two images. It measures image similarity from lightness, contrast and structural aspects respectively. The higher value of the SSIM index represents the higher similarity of the two images. The calculation method of SSIM is shown in Eq. (2).

$$SSIM = l(x, y) c(x, y) s(x, y) \quad (2)$$

where, $l(x, y)$ denotes the comparison of the lightness of x and y . $c(x, y)$ denotes the comparison of the contrast between x and y . $s(x, y)$ denotes the comparison of the structures of x and y .

Table 1: The similarity between the frames in a GOP of Foreman datasets

Frame No.	1	2	3	4	5	6	7	8	9
1	1.00	0.63	0.43	0.36	0.33	0.32	0.29	0.26	0.23
2	0.63	1.00	0.65	0.51	0.43	0.37	0.35	0.33	0.28
3	0.43	0.65	1.00	0.67	0.51	0.40	0.39	0.41	0.35
4	0.36	0.51	0.67	1.00	0.66	0.50	0.48	0.52	0.40
5	0.33	0.43	0.51	0.66	1.00	0.68	0.61	0.57	0.35
6	0.32	0.37	0.40	0.50	0.68	1.00	0.72	0.55	0.31
7	0.29	0.35	0.39	0.48	0.61	0.72	1.00	0.64	0.35
8	0.26	0.33	0.41	0.52	0.57	0.55	0.64	1.00	0.55
9	0.23	0.28	0.35	0.40	0.35	0.31	0.35	0.55	1.00

This paper combines the key frame and neighbor information to reconstruct the current block. In order to use as much nearest near-neighbor information as possible, this study first reconstructs the third and seventh frames using the first and ninth frames. Then reconstructs the fifth frame with the already reconstructed third and seventh frames. Finally, reconstructs the other frames in turn as shown in Fig. 4.

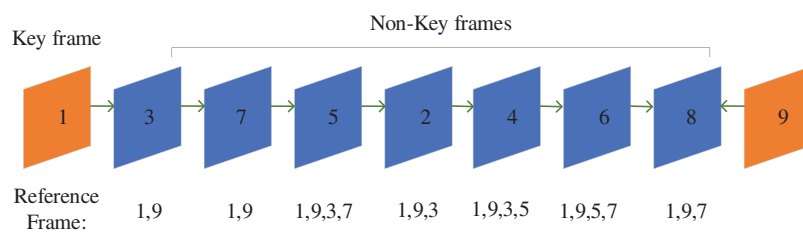


Figure 4: An example of NRFI in the GOP

As shown in Fig. 4, this study sets the GOP size to 8. It consists of frames one to eight. In addition, the first frame and the ninth frame are keyframes in the two GOPs, respectively. Meanwhile, the arrows between the video frames indicate the reconstruction order, and the corresponding reference frame number is below the video frames. Moreover, the maximum number of reference frames is 4. Taking a frame as an example, the reference frame of the fourth frame consists of the third and fifth frames already reconstructed near-neighbor frame plus two key frames.

3.2 Decoding Process

The encoded block comes to the decoder via the wireless channel. At the decoding end, BCS-SPL based on multi-hypothesis is used for initial reconstruction. For key frames, there is no more secondary reconstruction. For non-key frames, block-level search window and optimization of prediction blocks are constructed to complete secondary reconstruction in this paper.

3.2.1 Block-Level Search Window (BSW)

The search window finds a hypothetical block set similar to the current block. Fig. 5 shows a search process where the blue represents a subblock of the current non-key frames, orange represents the hypothetical block set of the current block, and the search process is carried out in key frames. The common search window is the same size whether high or low motion video sequences. For low motion video sequences, it is relatively good to use a small size of the search window. The distance between the optimal hypothetical block and the current block is small in the low motion frames. For high motion video sequences, it is better to use a larger search window. Otherwise, the search window cannot cover the position of the optimal hypothesis block. Therefore, block-level search window is designed in this study to automatically adjust the search window for each block.

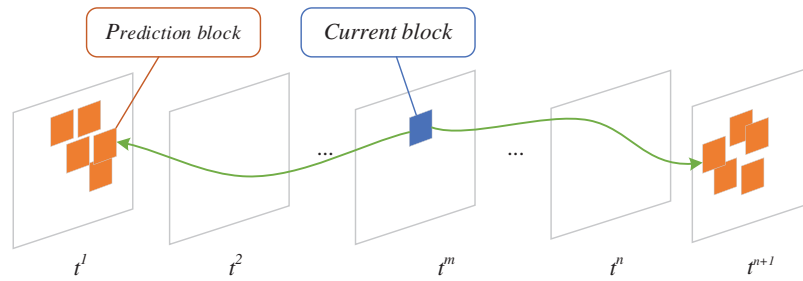


Figure 5: An illustration of the search process

The pixel difference between the current block and the corresponding block in the reference frame is analyzed in this paper. Fig. 6 shows partial block difference between the current frame and the reference frame. The horizontal coordinate represents the range where the difference lies. Vertical coordinate represents the number of blocks within a certain difference range. It can be seen from Fig. 6 that the difference in pixel values between the current frame and the corresponding block in the reference frame is very different. It ranges from 1 to 100 and beyond. For example, there are twenty-three blocks with a difference less than or equal to 7. However, there are thirty-two blocks with a difference greater than or equal to 65. The pixel difference between the current block and the corresponding block in the reference frame is not fixed. The difference is either too small, almost close to 1, or too large, even exceeding 100. A large search window wastes resources and a small search window is not conducive to finding the best hypothesis blocks. Hence, it is necessary to design a relatively good search window for each block.

First, it calculates the difference between the pixel value of the current block and the corresponding block in the reference frame. The difference can be expressed as shown in Eqs. (3) and (4).

$$v_1 = (|\Phi t_{h,i} - \Phi t_{1,i}|_1) / M \quad (3)$$

$$v_2 = (|\Phi t_{h,i} - \Phi t_{2,i}|_1) / M \quad (4)$$

where, $t_{h,i}$ represents the i -th image block in the current h -th frame t_h . The corresponding i -th block in the first reference frame is denoted by $t_{1,i}$. $t_{2,i}$ denotes the i -th block in the second reference frame. $\Phi t_{h,i}$ represents the measurements of the current block $t_{h,i}$. $\Phi t_{1,i}$ denotes the observations of the corresponding block i in t_1 . The measured value of the corresponding block i in t_2 is represented by $\Phi t_{2,i}$. M represents the number of samples. v_1 and v_2 both denote the difference.

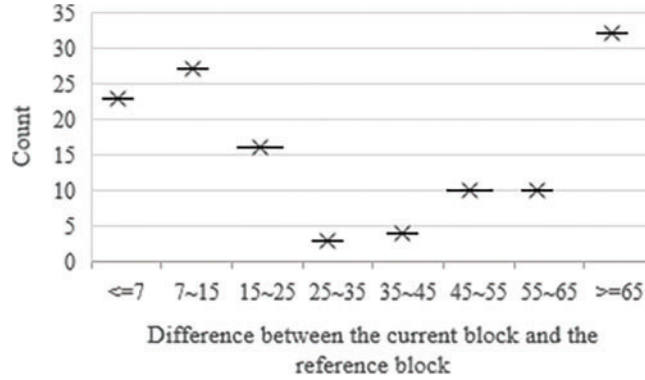


Figure 6: Partial block difference between the current frame and the reference frame

Second, v_1 and v_2 are used to classify blocks into different levels of search windows ($L1, L2, \dots, L7$) based on the difference. The higher the level means the larger the difference. The value of the search window is determined by the difference between blocks. Specifically, when the level of the search window is lower, the search window will be smaller. For example, the search window at the $L6$ level is larger than the search window at the $L5$ level. Each block has a different pixel difference from the corresponding block. It is necessary to set a better search window for each block to find the optimal hypothetical block. Therefore, this paper proposed BSW method to automatically adjust the search window for each block.

3.2.2 Optimization of Prediction Blocks

The relationship between the prediction block and the current block is shown in Fig. 5. The current block is marked with a blue square. It represents a block of the non-key frame. Orange squares are the hypothesis blocks. It consists of the blocks that are most similar to the current block $t_{h,j}$. The prediction block is obtained by assigning different weights to the set of hypothesis blocks. The calculation method of the prediction block is shown in Eq. (5).

$$\tilde{t}_{h,j} = \sum_{i=1}^m \alpha_i t_i^{hyp} \quad (5)$$

where, $\tilde{t}_{h,j}$ represents the prediction block. t_i^{hyp} represents the hypothesis block. α_i represents the corresponding weight of the hypothesis block. The reconstruction methods based on multiple hypothesis prediction does not take into account the effect of the prediction block on the reconstruction. Thus, OPBS is designed in our paper.

① Preliminary preparation for OPBS

The two questions of what blocks are worth optimizing and how to select them are prerequisites for OPBS. Zheng et al. [32] used residual value between current block and reference block to judgement the type of block. However, the method increased complexity on the encoding side. In addition, due

to the sampling rate is different of key frames and non-key frames, it is not appropriate to use the residuals between them. The difference between the current block and block of near-neighbor frame is more suitable because of the same compression rate in this paper, and the difference can be expressed in Eq. (6).

$$v = \|t_{h,j} - t_{adj,j}\|_2 \quad (6)$$

where, $t_{adj,j}$ represents the corresponding block of the current block $t_{h,j}$ in the adjacent frame. v represents the residual between them. In addition, The larger the value of the residual, the more variation in the blocks. The next thing we need to do is to find the block with the larger residual value. The classification of blocks is shown in Eq. (7).

$$t_{h,j} = \begin{cases} 1 & v > T_v \\ 0, & else \end{cases} \quad (7)$$

where, '1' represents prospect block. '0' represents background block. T_v represents the threshold of residual v . When $v > T_v$, the current block is selected as a prospect block. Otherwise, $t_{h,j}$ is chosen as a background block. In addition, the prospect block will perform the optimization operations of the prediction block. The background block will not perform the operation due to its relatively small variation. The distribution of v values is shown in Fig. 7.

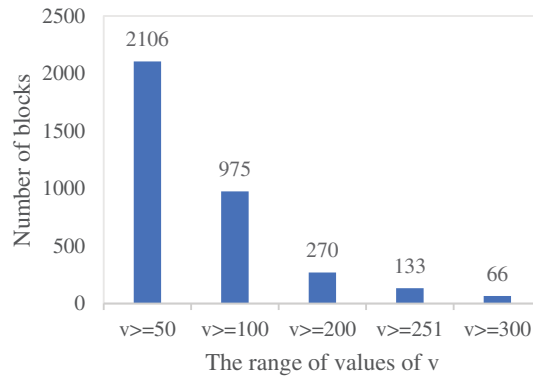


Figure 7: The distribution of v values

There are 6177 subblocks in each block. It is not advisable to optimize all blocks due to its high time complexity. Hence, this paper only optimizes some of the blocks. As can be seen from Fig. 7, when $v \geq 50$, there are 2106 subblocks. It accounts for about three quarters of t_h . Thus, T_v is set to 50.

② The core components of OPBS

I. Single objective OPBS-PSO model realizes optimization of prediction blocks

To enhance the reconstruction performance, the single-objective OPBS model based on PSO is proposed. The distance between the predicted block and the current block is designed as objective function in this paper. The calculation method of objective function is shown in Eq. (8).

$$fitness = \min \|t_{h,j} - \tilde{t}_{h,j}\|_2 \quad (8)$$

where, *fitness* represents finding the minimum distance between the prediction block $\tilde{t}_{h,j}$ and current block $t_{h,j}$. Generally, the smaller the distance between the predicted block and the current block, the greater their similarity. In addition, we also use the peak signal-to-noise ratio (PSNR) to measure the

reconstruction performance between the original frame and the final reconstruction frame as shown in Eq. (9).

$$PSNR = 10 \times \log_{10} \left(\frac{(2^n - 1)^2}{MSE} \right) \quad (9)$$

where, n represents the number of bits per sampled value. MSE represents mean squared deviation.

II. OPBS-PSO evolutionary algorithm

The objective of OPBS is to enhance the reconstruction performance. Moreover, the prediction block consists of multiple hypothesis blocks. However, the number of hypothesis blocks is fixed in the most paper. For example, the number of hypothetical blocks set by Zhao et al. [33] is five. They ignore the effect of the number of hypothetical blocks on the final reconstruction. The blocks that are more similar to the current block are simply treated as hypothetical blocks. Therefore, the number of hypothetical blocks is used as a decision variable for the OPBS-PSO evolutionary algorithm in our paper. The algorithm flow of OPBS-PSO is shown in Table 2.

Table 2: The pseudocode of OPBS-PSO algorithm

Algorithm 1: OPBS-PSO pseudocode

Input: Video sequence dataset T_h , Search window S , Sparsity D , The block size P_b , Number of hypothetical blocks $P = (P_1, P_2, P_3, P_4)$, Number of hypothetical blocks in the reference frame $N = (N_1, N_2, N_3, N_4)$, Number of iterations G_{max}

Output: Reconstructed video image T_f

```

1   The video image  $T_h$  is chunked to obtain  $t_{h,j}$ 
2   Find the comparison block  $t_{adj,j}$  of the current block  $t_{h,j}$  and calculate  $v = \|t_{h,j} - t_{adj,j}\|_2$ 
3   If  $v > T_v$ 
4       Initializing the population  $P$ 
5       While number of iterations  $< G_{max}$ 
6           Calculate fitness value  $F = [fitness]$ 
7           Updating personal best values and global best values.
8       End
9            $N_1 = P_1; N_2 = P_2; N_3 = P_3; N_4 = P_4;$ 
10  Else
11       $N_1 = 5; N_2 = 5; N_3 = 5; N_4 = 5;$ 
12  End
13  Reconstruct the prediction block  $\tilde{t}_{h,j}$  to get the final video frame  $T_f$ .
```

The main steps of OPBS-PSO model are as follows:

Step 1: Calculate the difference between the current block and the adjacent block. Then determine the type of blocks based on the difference.

Step 2: Perform the following loop when the block is prospect block.

Step 3: Initialize the population. The initialization is carried out according to the decision range of the number of hypothetical blocks in the reference frame, and the number of hypothesis blocks is formed for each individual.

Step 4: Perform velocity and position update operations to produce offspring.

Step 5: Solve the objective value according to the designed objective function and the prediction block is obtained.

Step 6: Updating personal best values and global best values.

Step 7: Determine whether the maximum number of iterations is satisfied. If so, terminate the operation. Otherwise, return to the Step 4 and continue iterating until the termination condition is met.

Step 8: When the optimization is completed, a relatively better prediction block is obtained.

4 Simulation Experiment and Analysis

4.1 Experimental Data

This paper adopts six video sequences to conduct extensive experiments. It includes Foreman, Paris, Carphone, Coastguard, News and Highway. These sequences are available for download at <https://media.xiph.org/video/derf/>. Seventeen frames of each video sequence are used for the experiments. Common intermediate format (CIF) is used to conduct the experiments. That is, the resolution of the image is 352 by 288 pixels. Furthermore, the paper is block-based, and all experiments are done with MATLAB R2018b in Windows 10 environment.

4.2 Parameter Settings

There are many parameters in the experiment. It includes the parameters related to the encoding process, decoding process and OPBS-PSO optimization algorithm. The experimental settings of encoding stage and decoding stage are shown in Table 3. Furthermore, the parameter settings of OPBS-PSO are shown in Table 4.

Table 3: Parameters setting of encoding and decoding stage

Parameter	Meaning	Value
S_b	Size of block	32
Φ	Measurement matrix	Gaussian Random
K_{rate}	Sampling rate of key frame	0.7
N_{rate}^k	Sampling rate of non-key frame	0.05, 0.1, 0.2, 0.3, 0.4

Table 4: Parameters setting of OPBS-PSO optimization algorithm

Parameter	Meaning	Value
Pop	Population	50
D	Decision variables	4
I	Iterations	30
F	Objective function	1
V_{min}	Lower bound of decision variables	1
V_{max}	Upper bound of decision variables	11

In this experiment, each frame is divided into blocks with 32 by 32 pixel. Gaussian Random is used as measurement matrix. In addition, the sampling rate of key frame is 0.7 and the sampling rate of non-key frame is 0.05, 0.1, 0.2, 0.3 and 0.4. In the parameters setting of OPBS-PSO, the population is set to 50, D represents the dimensions of decision variables, I represents the number of iterations, F represents the number of objectives. Moreover, the value range of each decision variable is set as [1,11].

4.3 Experimental Analysis

To verify the validity of the presented method, this paper conducts experiments on six datasets, and evaluates the performance with three methods. It includes MC/ME [35], VCSNet [30], and RRS [33]. The comparative results are shown in the figures below.

4.3.1 The Performance of NRFI

As can be seen from Fig. 8, NRFI improves significantly over all three other methods. For Foreman sequences, NRFI improves by an average of 5.07 dB over MC/ME and increases of 4.56 and 0.51 dB over VCSNet and RRS, respectively. Furthermore, it can be seen from Fig. 8 that NRFI is superior to other algorithms on Paris, Carphone and Coastguard datasets.

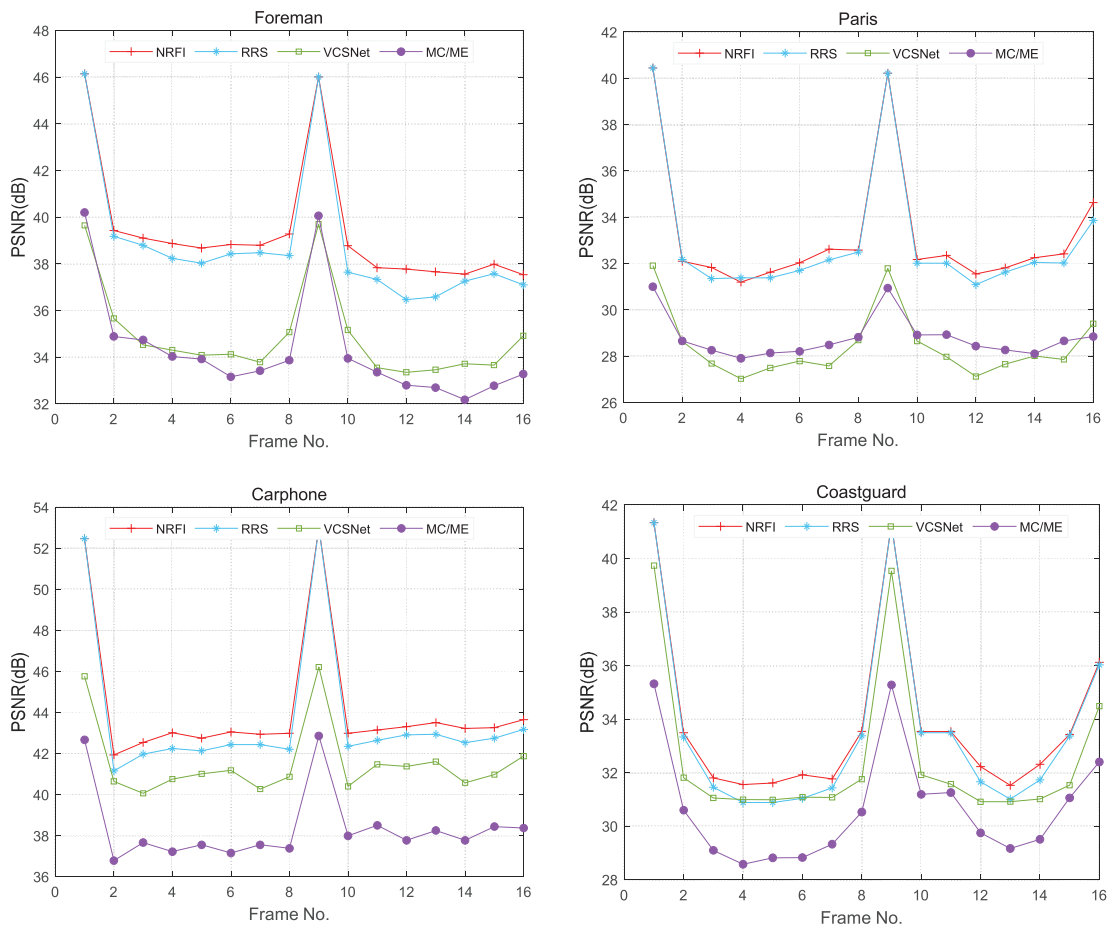


Figure 8: The comparison of the reconstruction performance of NRFI (marked in red) with the three methods

4.3.2 The Performance of BSW

To verify the effectiveness of BSW, we compared BSW with three other methods on five video sequences. The best reconstruction performance is marked in bold. As can be seen from Table 5, BSW demonstrates excellent reconstruction performance on the majority of frames.

Table 5: The comparison of the reconstruction performance of BSW with the three methods on non-key frames

Sequence	Method	2	3	4	5	6	7	8	10	11	12	13	14	15	16
Foreman	BSW	39.26	38.81	38.32	38.11	38.46	38.53	38.46	37.75	37.67	36.70	36.81	37.38	37.66	37.21
	RRS	39.18	38.79	38.23	38.03	38.43	38.48	38.35	37.65	37.33	36.47	36.59	37.26	37.58	37.10
	VCSNet	35.67	34.53	34.31	34.09	34.13	33.79	35.08	35.17	33.55	33.35	33.46	33.72	33.66	34.92
	MCME	34.89	34.74	34.03	33.92	33.16	33.42	33.87	33.95	33.36	32.80	32.70	32.18	32.78	33.28
Paris	BSW	32.23	31.41	31.44	31.44	31.81	32.25	32.58	32.09	32.09	31.20	31.73	32.09	32.11	34.32
	RRS	32.18	31.35	31.38	31.38	31.70	32.16	32.49	32.01	32.01	31.09	31.63	32.05	32.02	33.86
	VCSNet	28.64	27.68	27.03	27.50	27.79	27.58	28.69	28.66	27.98	27.13	27.66	28.02	27.86	29.41
	MCME	28.66	28.26	27.91	28.14	28.21	28.49	28.82	28.92	28.93	28.44	28.27	28.11	28.66	28.85
Carphone	BSW	41.27	42.01	42.26	42.23	42.45	42.46	42.22	42.43	42.66	42.92	42.98	42.55	42.78	43.17
	RRS	41.15	41.97	42.24	42.13	42.43	42.44	42.20	42.35	42.64	42.91	42.94	42.53	42.75	43.17
	VCSNet	40.64	40.06	40.76	41.01	41.18	40.27	40.87	40.40	41.47	41.37	41.60	40.58	40.97	41.87
	MCME	36.78	37.66	37.22	37.55	37.15	37.55	37.38	37.99	38.50	37.77	38.25	37.77	38.44	38.37
Coastguard	BSW	33.35	31.73	31.22	31.22	31.36	31.67	33.40	33.51	33.51	31.88	31.34	31.94	33.32	36.04
	RRS	33.33	31.46	30.89	30.89	31.04	31.43	33.38	33.49	33.49	31.66	31.01	31.73	33.37	36.02
	VCSNet	31.81	31.06	30.99	30.99	31.08	31.08	31.76	31.92	31.58	30.92	30.92	31.02	31.53	34.49
	MCME	30.60	29.10	28.58	28.82	28.83	29.33	30.53	31.19	31.26	29.75	29.17	29.51	31.06	32.40
Highway	BSW	39.32	38.14	38.30	38.21	38.98	38.02	39.32	38.60	39.09	38.80	38.77	38.79	39.01	39.01
	RRS	39.29	37.85	38.16	38.11	38.99	37.76	39.34	38.44	39.02	38.60	38.63	38.67	39.02	39.01
	VCSNet	35.00	34.31	34.01	33.17	33.84	34.05	35.94	35.10	34.02	32.86	33.05	33.90	33.52	34.46
	MCME	34.73	33.77	33.96	34.28	34.75	34.93	35.00	34.03	34.86	34.18	34.08	33.14	35.16	35.16

4.3.3 The Overall Performance of BSW and NRFI

As can be seen in Fig. 9, the proposed method (red noted) has obvious advantages in enhancing reconstruction performance in the five video sequences of Foreman, Paris, Carphone, Coastguard and Highway. Specifically, for Foreman video sequences, the overall improvement of the proposed method over RRS (blue noted) is 0.3–1.66 dB, with an overall average enhancement of 0.63 dB; For Paris video sequences, the improvement effect of the presented method is obviously inferior to that of Foreman video sequences, but it is still an enhancement in a comprehensive view; For Carphone video sequence, the proposed method achieves an average improvement of 0.49 dB over RRS. The proposed method improves 0.41 and 0.24 dB on the Coastguard video sequence and Highway video sequence respectively over RRS.

However, the effect in the News video sequence is poor. After analysis, it may be caused by the transformation speed of frames in the video sequence. It is not difficult to see from the research of other scholars [10,35] that the reconstruction method depends on the data set to a certain extent. Aiming at this reason, hope broad scholars will discuss the problem together with us.

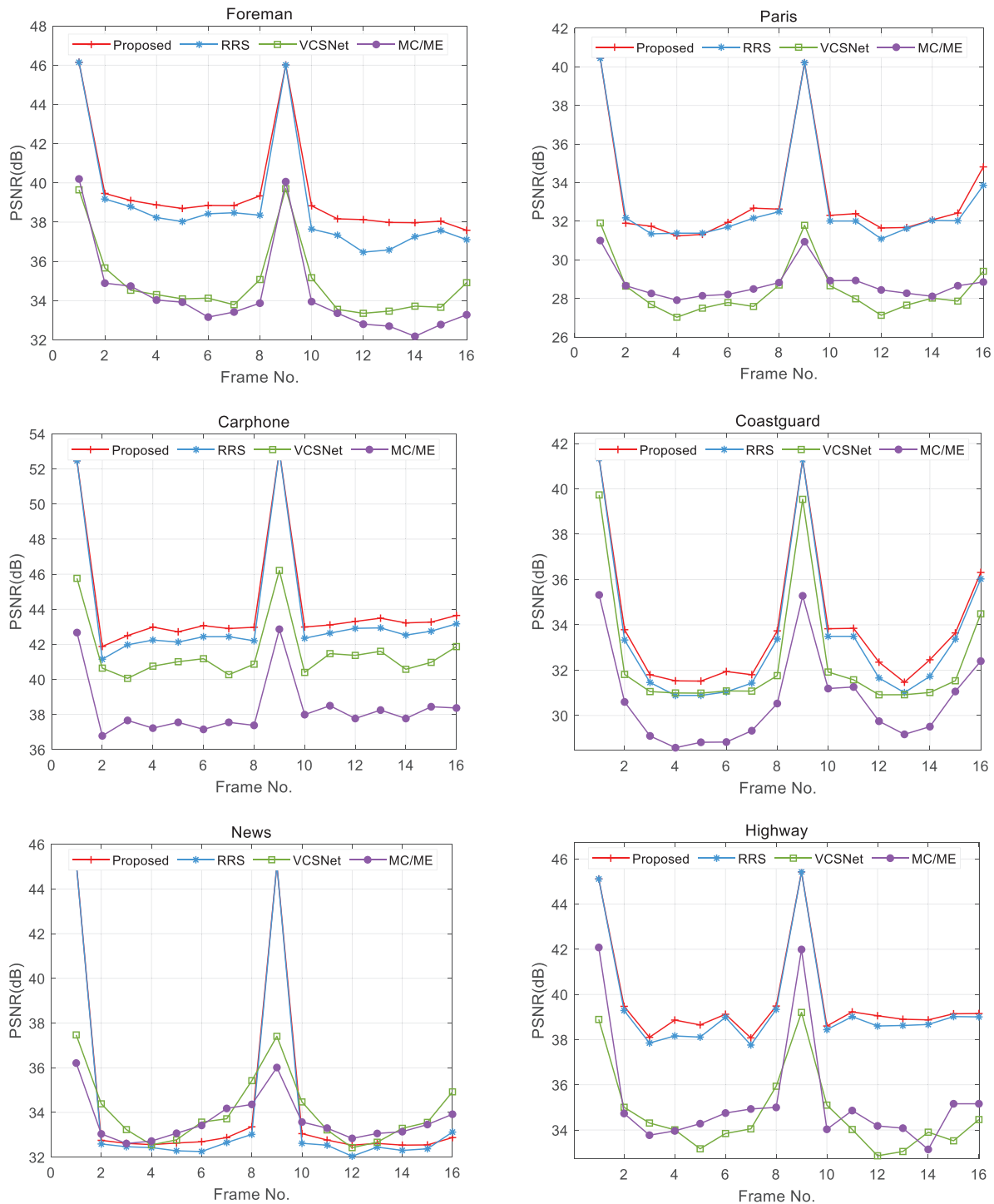


Figure 9: The reconstruction performance of presented algorithm (red noted) and three comparison methods for six video datasets

Table 6 shows the comparison of NRFI, BSW and NRFI+BSW at different sampling rates. It can be seen from Table 6 that NRFI+BSW demonstrates better performance on most datasets.

Table 6: The comparison of reconstruction performance at different sampling rates

Subrate	Sequence	NRFI	BSW	NRFI + BSW
0.05	Foreman	38.12	38.21	38.15
	Paris	31.96	31.91	31.94
	Carphone	43.92	44.07	43.93
	Coastguard	32.73	32.77	32.8
	News	34.42	34.45	34.5
	Highway	39.61	39.69	39.64
0.1	Foreman	38.21	38.28	38.24
	Paris	32.41	32.31	32.5
	Carphone	44.03	44.16	44.04
	Coastguard	32.81	32.83	32.88
	News	34.81	34.73	34.86
	Highway	39.7	39.76	39.73
0.2	Foreman	40.57	40.23	40.59
	Paris	33.85	33.76	33.79
	Carphone	44.96	44.49	44.94
	Coastguard	34.26	34.06	34.3
	News	35.52	35.34	35.53
	Highway	40.21	40.09	40.26
0.3	Foreman	41.79	41.55	41.8
	Paris	35.58	35.44	35.65
	Carphone	46.51	45.91	46.49
	Coastguard	35.71	35.3	35.77
	News	36.59	36.3	36.61
	Highway	41.23	41.08	41.27
0.4	Foreman	42.88	42.67	42.88
	Paris	37.36	37.13	37.46
	Carphone	47.91	47.23	47.88
	Coastguard	37.03	36.41	37.06
	News	38.14	33.45	38.15
	Highway	42.16	42.01	42.18

4.3.4 Proposed OPBS Compared with Other Video Reconstruction Methods

As can be seen from Fig. 10, OPBS (red noted) demonstrates excellent reconstruction performance by comparison of the first five frames. For Highway sequences, it improves by 3.03–4.88 dB over MC/ME and increases by 3.88–6.22 dB and 0.00–0.68 dB over VCSNet and RRS, respectively.

To verify the difference between optimized partial blocks and all blocks, experiment is conducted in this paper. As shown in Fig. 11. The orange dots represent the optimization of all blocks. The blue

dots represent that only the foreground blocks are optimized. Through the validation on the Foreman dataset, it can be seen that the results of OPBS-All blocks are better. However, only foreground blocks are selected for optimization of prediction blocks in our paper. Due to the high time complexity of optimizing all blocks, we choose to optimize only part of the blocks.

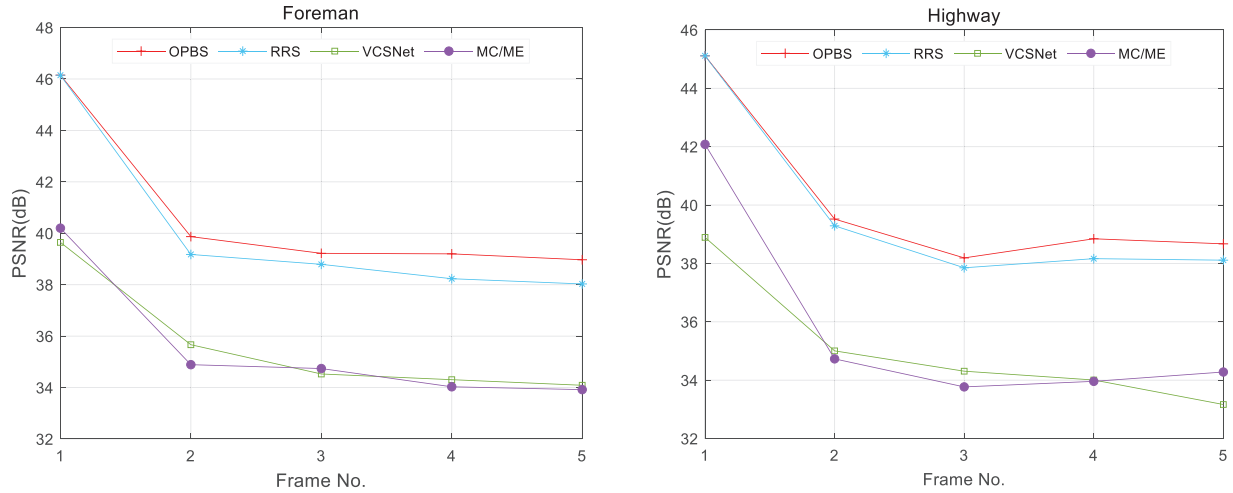


Figure 10: The reconstruction performance of OPBS and three comparison methods

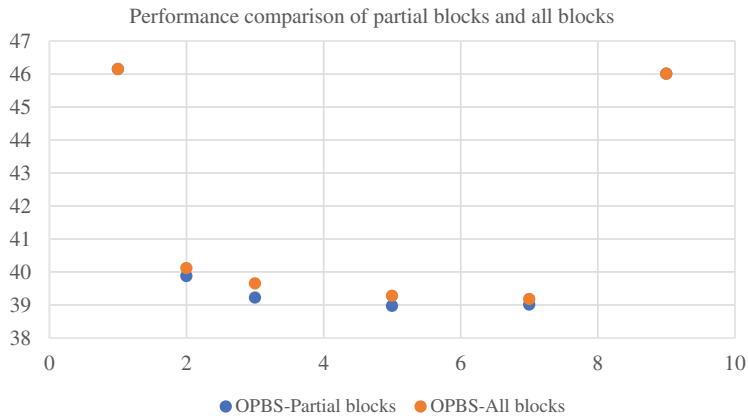


Figure 11: The performance of optimizing all blocks vs. partial blocks on the Foreman dataset

This paper takes out the fifth frame of the Foreman video sequence for visual comparison. As we all know, MC/ME is a representative method, and the proposed method achieves a 5.35 dB improvement over MC/ME as illustrated in Fig. 12; VCSNet uses Convolutional Neural Networks (CNN) for video reconstruction, and this paper’s method enhances 5.19 dB at a sampling rate of 0.2. In addition, RRS is based on prediction-residual reconstruction, and the presented method improves 1.25 dB over RRS.



Figure 12: Visual contrast of the reconstruction results of the 5th frame with diverse ways

To evaluate the robustness of our algorithm, this paper takes out the third frame of the Paris video sequence for visual comparison. It can be seen from Fig. 13 that the proposed method achieves a 4.21 dB improvement over MC/ME; This paper’s method enhances 4.69 dB at a sampling rate of 0.2

over VCSNet. In addition, the presented method improves 1.03 dB over RRS. The experimental results fully demonstrate that OPBS have strong superiority and robustness in VCS.

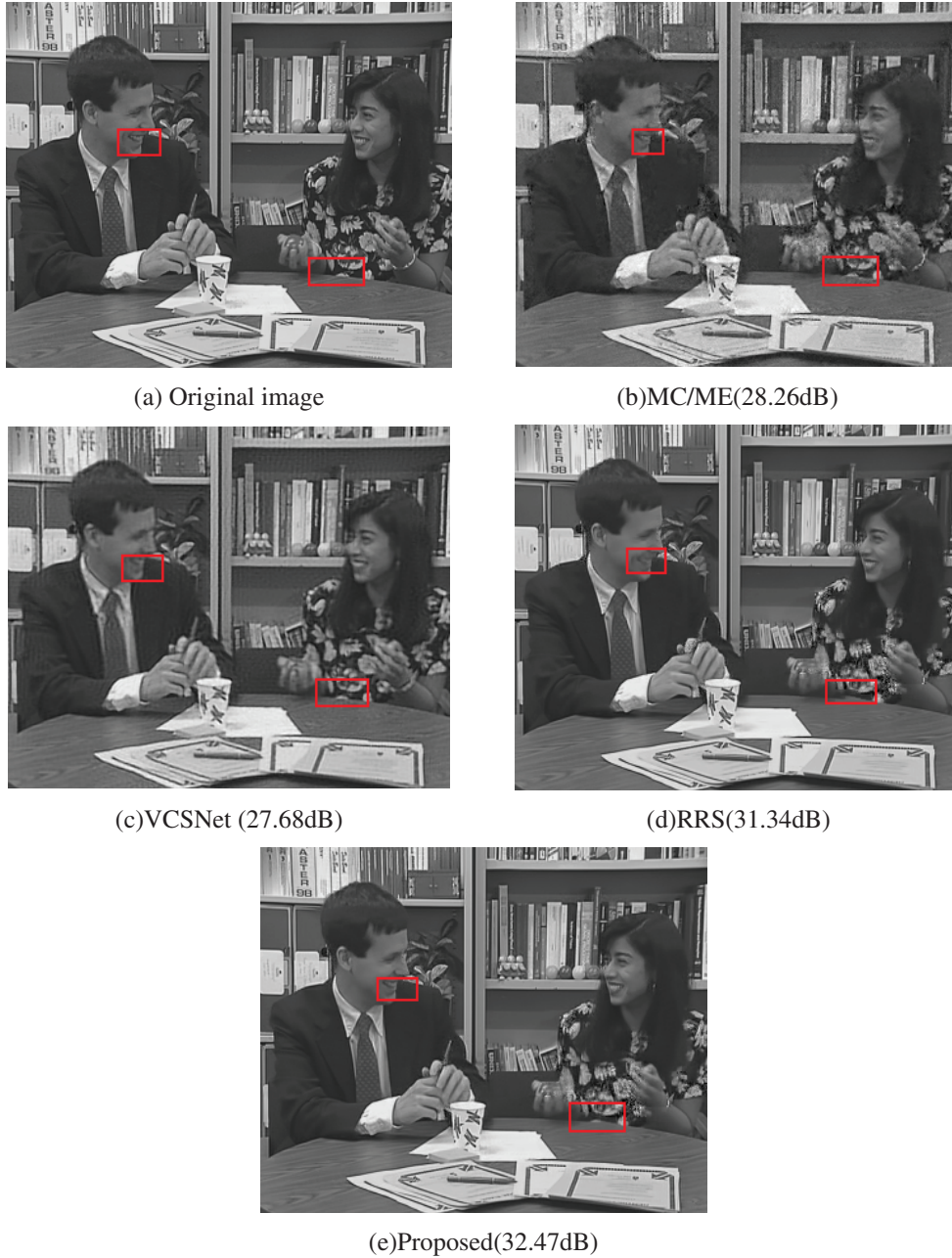


Figure 13: Visual contrast of the reconstruction results of the 3th frame with diverse ways

5 Conclusion and Prospect

This paper proposes a CS-HAVR framework. It consists of two stages: encoding and decoding.

On the encoding side, NRFI is designed to provide more useful information during the reconstruction of non-key frames. First, the similarity between frames is analyzed, and the key frame and near-neighbor information is combined to reconstruct the current block. On the decoding side, a two-stage reconstruction strategy is used because of the low sampling rate for non-keyframes. Simultaneously, OPBS-PSO and BSW are constructed to improve the precision of prediction blocks. In the proposed OPBS-PSO model, this paper combines the high-quality optimization reconstruction of the foreground block with the background block's multiple hypothesis prediction reconstructions to improve the video sequence's overall reconstruction effect.

In the proposed BSW method, this paper sets a better search window for each block to find the optimal hypothetical block. Six video sequences are used to conduct extensive experiments with three other methods in this paper. In addition, the Forman data set is utilized to compare the reconstruction performance of partial block optimization and total block optimization in the experimental part. The experimental results show that optimizing all blocks has obvious advantages, but it brings high time complexity simultaneously.

The improvement of reconstruction quality of multi-hypothesis-based video codecs is influenced by various factors. If these factors are considered simultaneously, it will greatly improve the quality of video reconstruction. In the future, we intend to use intelligent optimization algorithms [36–39] with multiple objectives to jointly optimize the main factors that affect the generation of prediction blocks to create better video decoders.

Funding Statement: This work is supported by the National Natural Science Foundation of China under Grant No. 61806138; Key R&D Program of Shanxi Province (International Cooperation) under Grant No. 201903D421048; National Key Research and Development Program of China under Grant No. 2018YFC1604000; School Level Postgraduate Education Innovation Projects under Grant No. XCX212082.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4), 1289–1306. <https://doi.org/10.1109/TIT.2006.871582>
2. Alhayani, B., Abbas, S. T., Mohammed, H. J., Mahajan, H. B. (2021). Intelligent secured two-way image transmission using corvus corone module over WSN. *Wireless Personal Communications*, 120(1), 665–700. <https://doi.org/10.1007/s11277-021-08484-2>
3. Park, W., Kim, M. (2021). Deep predictive video compression using mode-selective uni- and bi-directional predictions based on multi-frame hypothesis. *IEEE Access*, 9, 72–85. <https://doi.org/10.1109/ACCESS.2020.3046040>
4. Zhang, R. F., Wu, S. H., Wang, Y., Jiao, J. (2020). High-performance distributed compressive video sensing: Jointly exploiting the HEVC motion estimation and the ℓ_1 – ℓ_1 reconstruction. *IEEE Access*, 8, 31306–31316. <https://doi.org/10.1109/ACCESS.2020.2973392>
5. Lalithambigai, B., Chitra, S. (2022). A hybrid adaptive block based compressive sensing in video for IoMT applications. *Wireless Network*, 10(2), 8202. <https://doi.org/10.1007/s11276-021-02847-0>
6. Barrios, Y., Guerra, R., Lopez, S., Sarmiento, R. (2022). Adaptation of the CCSDS 123.0-B-2 standard for RGB video compression. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15, 1656–1669. <https://doi.org/10.1109/JSTARS.2022.3145751>

7. Qiao, M., Meng, Z. Y., Ma, J. W., Yuan, X. (2020). Deep learning for video compressive sensing. *APL Photonics*, 5(3), 030801. <https://doi.org/10.1063/1.5140721>
8. Chai, X. L., Fu, X. L., Gan, Z. H., Zhang, Y., Lu, Y. et al. (2020). An efficient chaos-based image compression and encryption scheme using block compressive sensing and elementary cellular automata. *Neural Computing and Applications*, 32(9), 4961–4988. <https://doi.org/10.1007/s00521-018-3913-3>
9. Chai, X. L., Wu, H. Y., Gan, Z. H., Zhang, Y., Chen, Y. et al. (2020). An efficient visually meaningful image compression and encryption scheme based on compressive sensing and dynamic LSB embedding. *Optics and Lasers in Engineering*, 124, 105837. <https://doi.org/10.1016/j.optlaseng.2019.105837>
10. Hu, Z. X., Wang, Y., Ge, M. F., Liu, J. (2020). Data-driven fault diagnosis method based on compressed sensing and improved multiscale network. *IEEE Transactions on Industrial Electronics*, 67(4), 3216–3225. <https://doi.org/10.1109/TIE.2019.2912763>
11. Zhao, S. P., Liu, R. F., Wang, Y., Wang, Y., Wang, P. et al. (2020). Single-pixel foreground imaging without a priori background sensing. *Optics Express*, 28(18), 26018–26027. <https://doi.org/10.1364/OE.400017>
12. Belyaev, E. (2021). Fast decoding and parameters selection for CS-JPEG video codec. *IEEE 23rd International Workshop on Multimedia Signal Processing*, Tampere, Finland.
13. Mahdaoui, A. E., Ouahabi, A., Moulay, M. S. (2022). Image denoising using a compressive sensing approach based on regularization constraints. *Sensors*, 22(6), 2199. <https://doi.org/10.3390/s22062199>
14. Li, H. G. (2019). Compressive domain spatial-temporal difference saliency-based realtime adaptive measurement method for video recovery. *IET Image Processing*, 13(11), 2008–2017. <https://doi.org/10.1049/iet-ipr.2019.0116>
15. Das, S., Mandal, J. K. (2021). An enhanced block-based compressed sensing technique using orthogonal matching pursuit. *Signal, Image and Video Processing*, 15(3), 563–570. <https://doi.org/10.1007/s11760-020-01777-2>
16. Belyaev, E., Codreanu, E., Juntti, E., Egiazarian, K. (2020). Compressive sensed video recovery via iterative thresholding with random transforms. *IET Image Processing*, 14(6), 1187–1200. <https://doi.org/10.1049/iet-ipr.2019.0661>
17. Heng, S., Aimtongkham, P., Vo, V. N., Nguyen, T. G., So-In, C. (2020). Fuzzy adaptive-sampling block compressed sensing for wireless multimedia sensor networks. *Sensors*, 20(21), 6217. <https://doi.org/10.3390/s20216217>
18. Xu, Y. G., Xue, Y. Z., Hua, G., Cheng, J. (2020). An adaptive distributed compressed video sensing algorithm based on normalized bhattacharyya coefficient for coal mine monitoring video. *IEEE Access*, 8, 158369–158379. <https://doi.org/10.1109/ACCESS.2020.3020140>
19. Unde, A. S., Pattathil, D. P. (2020). Adaptive compressive video coding for embedded camera sensors: Compressed domain motion and measurements estimation. *IEEE Transactions on Mobile Computing*, 19(10), 2250–2263. <https://doi.org/10.1109/TMC.2019.2926271>
20. Tong, Z., Wang, F., Hu, C., Wang, J., Han, S. (2020). Preconditioned generalized orthogonal matching pursuit. *EURASIP Journal on Advances in Signal Processing*, 2020(1), 21. <https://doi.org/10.1186/s13634-020-00680-9>
21. Poovathy J. F. G., Sankararajan, R. (2021). Split-process-merge technique-based algorithm for accelerated recovery of compressively sensed images and videos. *Wireless Personal Communications*, 118(1), 93–108. <https://doi.org/10.1007/s11277-020-08003-9>
22. Das, S. (2021). Hyperspectral image, video compression using sparse tucker tensor decomposition. *IET Image Processing*, 15(4), 964–973. <https://doi.org/10.1049/ipr2.12077>
23. Banerjee, R., Bit, S. D. (2019). Low-overhead video compression combining partial discrete cosine transform and compressed sensing in WMSNs. *Wireless Networks*, 25(8), 5113–5135. <https://doi.org/10.1007/s11276-019-02119-y>

24. Ma, N. (2019). Distributed video coding scheme of multimedia data compression algorithm for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2019(1), 254. <https://doi.org/10.1186/s13638-019-1571-5>
25. Zhou, J. W., Fu, Y. C., Yang, Y. C., Ho, A. T. S. (2019). Distributed video coding using interval overlapped arithmetic coding. *Signal Processing: Image Communication*, 76, 118–124.
26. Zheng, S., Chen, J., Kuo, Y. H. (2019). A multi-level residual reconstruction-based image compressed sensing recovery scheme. *Multimedia Tools and Applications*, 78(17), 25101–25119. <https://doi.org/10.1007/s11042-019-07746-3>
27. Zheng, S., Chen, J., Zhang, X. P., Kuo, Y. (2021). A new multihypothesis-based compressed video sensing reconstruction system. *IEEE Transactions on Multimedia*, 23, 3577–3589. <https://doi.org/10.1109/TMM.2020.3028479>
28. Zhao, Z. F., Xie, X. M., Liu, W., Pan, Q. Z. (2020). A hybrid-3D convolutional network for video compressive sensing. *IEEE Access*, 8, 20503–20513. <https://doi.org/10.1109/ACCESS.2020.2969290>
29. Ebrahim, M., Adil, S. H., Raza, K., Ali, S. S. A. (2020). Block compressive sensing single-view video reconstruction using joint decoding framework for low power real time applications. *Applied Sciences*, 10(22), 7963. <https://doi.org/10.3390/app10227963>
30. Shi, W. Z., Liu, S. H., Jiang, F., Zhao, D. (2021). Video compressed sensing using a convolutional neural network. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(2), 425–438. <https://doi.org/10.1109/TCSVT.2020.2978703>
31. Kang, L. W., Lu, C. S. (2009). Distributed compressive video sensing. *IEEE International Conference on Acoustics*, pp. 1169–1172. Taipei, Taiwan.
32. Zheng, S., Zhang, X. P., Chen, J., Kuo, Y. (2019). A high-efficiency compressed sensing-based terminal-to-cloud video transmission system. *IEEE Transactions on Multimedia*, 21(8), 1905–1920. <https://doi.org/10.1109/TMM.2019.2891415>
33. Zhao, C., Ma, S. W., Zhang, J., Xiong, R., Gao, W. (2017). Video compressive sensing reconstruction via reweighted residual sparsity. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(6), 1182–1195. <https://doi.org/10.1109/TCSVT.2016.2527181>
34. Zhang, J., Zhang, Y., Cai, X., Xie, L. (2023). Three-stages hyperspectral image compression sensing with band selection. *Computer Modeling in Engineering & Sciences*, 134(1), 293–316. <https://doi.org/10.32604/cmescs.2022.020426>
35. Mun, S., Fowler, J. E. (2011). Residual reconstruction for block-based compressed sensing of video. *IEEE Data Compression Conference*, pp. 183–192. Snowbird, USA.
36. Cai, X., Cao, Y., Ren, Y., Cui, Z., Zhang, W. (2021). Multi-objective evolutionary 3D face reconstruction based on improved encoder-decoder network. *Information Sciences*, 581(20), 233–248. <https://doi.org/10.1016/j.ins.2021.09.024>
37. Cai, X., Hu, Z., Chen, J. (2020). A many-objective optimization recommendation algorithm based on knowledge mining. *Information Sciences*, 537(3), 148–161. <https://doi.org/10.1016/j.ins.2020.05.067>
38. Cai, X., Geng, S. (2021). A multicloud-model-based many-objective intelligent algorithm for efficient task scheduling in internet of things. *IEEE Internet of Things Journal*, 8(12), 9645–9653. <https://doi.org/10.1109/JIOT.2020.3040019>
39. Yang, W., Chen, L., Li, Y., Abid, F. (2022). A many-objective particle swarm optimisation algorithm based on convergence assistant strategy. *International Journal of Bio-Inspired Computation*, 20(2), 104–118. <https://doi.org/10.1504/IJBIC.2022.126773>