Recovery of Compressive Video Sensing via Dictionary Learning and Forward Prediction

Nasser Eslahi, Ali Aghagolzadeh, Seyed Mehdi Hosseini Andargoli
Faculty of Electrical and Computer Engineering
Babol University of Technology, Babol, Iran
Email: nasser.eslahi@stu.nit.ac.ir, aghagol@nit.ac.ir, smh_andargoli@nit.ac.ir

Abstract—In this paper, we propose a new framework for compressive video sensing (CVS) that exploits the inherent spatial and temporal redundancies of a video sequence, effectively. The proposed method splits the video sequence into the key and non-key frames followed by dividing each frame into the small non-overlapping blocks of equal sizes. At the decoder side, the key frames are reconstructed using adaptively learned sparsifying (ALS) basis via $\ell_0$ minimization, in order to exploit the spatial redundancy. Also, three well-known dictionary learning algorithms are investigated in our method. For recovery of the non-key frames, a prediction of the current frame is initialized, by using the previous reconstructed frame, in order to exploit the temporal redundancy. The prediction is employed in a proper optimization problem to recover the current non-key frame. To compare our experimental results with the results of some other methods, we employ pick signal to noise ratio (PSNR) and structural similarity (SSIM) index as the quality assessor. The numerical results show the adequacy of our proposed method in CVS.

Keywords—Compressive Video Sensing, Sparse Recovery, Split Bregman Iteration, Dictionary Learning, Spatial/Temporal Redundancy.

I. INTRODUCTION

Due to great efforts by Candès et al. [1], [2] and Donoho [3], compressive sensing (also called compressed sensing or compressive sampling) suggests a new framework for simultaneous sampling and compression of signals at a rate significantly below the Nyquist rate. It also permits that under certain conditions, the original signal can be reconstructed properly from a small set of measurements via solving the convex optimization problem or iterative greedy recovery algorithms. Recently, the idea of compressive sensing (CS) for imaging (single pixel camera [4]) has been extended to the conventional predictive/distributed video coding, to develop highly desirable compressive video sensing (CVS)/distributed compressive video sensing (DCVS). CVS employs both data acquiring (video sensing) and compression into a unified task which emerges a new procedure to directly acquiring compressed video data via random projection for each individual frame (or blocks of frame) at a low-complexity encoder.

Several CVS recovery methods have already been proposed, e.g., Wakin et al. [5] proposed an intuitive (motion JPEG motivated) approach which extends compressive image sensing to video applications by considering each frame of the video sequence independently and recovers each frame using the 2D discrete cosine transform (2D DCT) or a 2D discrete wavelet transform (2D DWT), individually. However, this simple extension fails to address the temporal redundancy in video. Prades-Nobet et al. [6] proposed a block-based selective video sampling scheme where first divides frames of the video sequence into key frames and non-key frames; then each frame is divided into the small non-overlapping blocks of equal sizes. The key frames are sampled fully using conventional video compression techniques, e.g., MPEG4/H.264. Non-key frames are projected and recovered using CS techniques, with an adaptive redundant dictionary built by picking a set of local (spatially neighboring) blocks which are extracted from its co-located blocks in the previous decoded key frame. A similar approach was proposed in [7]. Nevertheless, such local dictionary-based basis may not work very well for blocks with large motion or when the entire scene undergoes translation motion. Another dictionary based approach is presented in [8], where the dictionary is learned from a set of blocks globally extracted from the previous reconstructed neighboring frames together with the side information generated from them. This work has been extended to assign dynamic measurement rate allocation (for different local regions) by incorporating a feedback channel [9]. Liu et al. [10] proposed a redundant dictionary generation scheme for compressed video sensing, which follows the sparse representation approach of [6]-[7].

In this paper we propose a novel for sampling and recovering of compressed sensed video data. The proposed method divides the video sequence into the key and non-key frames followed by dividing each frame into the small blocks of equal size, similar to [6]. Each blocks of key (non-key) frames are sampled using the same sensing matrix $\Phi_{B_K}$ ($\Phi_{B_{BK}}$). The compressed key frame data are reconstructed initially using multi-hypothesis block compressed sensing recovery [11], in order to use as the initial image in the dictionary leaning algorithm step, to obtain an adaptively learned sparsifying (ALS) basis to exploit the spatial redundancy of frame, by an iterative procedure. Also, we investigate the effectiveness of three well-known dictionary learning methods, to adopt in our scheme. The obtained ALS basis is incorporated into an optimization problem, for the whole CS frame recovery in the form of $\ell_0$ quasi-norm. In this step, a split Bregman iteration (SBI) [12]-[13] based technique is employed to solve the non-convex $\ell_0$ minimization, efficiently. For the recovery of non-key frames, first we initialize a prediction of current frame $F_t$ by means of the previous reconstructed frame $F_{t-1}$, in order to exploit the temporal redundancy. The prediction $F_t$ is employed into a SBI based method together with the achieved ALS basis of current frame and current CS data, to recover the current non-key frame $F_t$ by solving the proper minimization.
problem. The experimental results show the high competitive performance of our proposed method comparing with the other state-of-the-art CVS techniques.

The rest of this paper is organized as follows. In section II, we introduce an abstract framework for CS theory and SBI method. Also, we review three well-known techniques of dictionary learning, briefly. The proposed method is described in section III. Numerical results and comparisons for our proposed method are given in section IV, and finally section V concludes the paper.

II. BACKGROUND

This section reviews an abstract framework of CS theory and SBI algorithm, accompanied by investigation of three well-known dictionary learning methods, briefly.

A. Compressed Sensing

Suppose we wish to recover a real value finite length signal \( u \in \mathbb{R}^n \) from a finite length observation \( f \in \mathbb{R}^m \); so that \( m \ll n \) and there is a linear projection between them

\[
f_{m \times 1} = \Phi_{m \times n} u_{n \times 1} + e_{m \times 1}
\]

where \( \Phi \in \mathbb{R}^{m \times n} \) is a sensing matrix and \( e \in \mathbb{R}^{m \times 1} \) denotes the additive noise. Since the number of unknowns is much more than the observations, clearly we are not able to recover every \( u \) from \( f \) and it is generally considered as an ill-posed problem. However, if \( u \) be sufficiently sparse in the sense that it can be written as a superposition of a small number of vectors taken from a known (sparsifying) transform domain basis (\( t = n \)) or frame (\( t > n \)) \( \Psi \in \mathbb{R}^{t \times n} \) or even adaptive learned sparsifying basis (e.g., see [14]), then exact recovery of \( u \) is possible. So sparsity plays a key role in recovering of \( u \) from observation vector \( f \). Also, \( u \) would be called \( s \)-sparse if only its \( s \) coefficients in the set of transform domain \( \vartheta = \Psi u \) are nonzero and the other \( n - s \) coefficients are zero. In other word, the transform domain signal \( \vartheta \) can be well approximated using only \( s < m \ll n \) nonzero entries. In order to solve the reconstruction problem with a reasonable accuracy and robustness to the noise, the estimation of \( u \) is formulated as an unconstrained Lagrangian optimization problem which incorporates the prior information about the original signal

\[
\min_{\vartheta} \left\{ \frac{1}{2} \| f - \Phi \Psi^{-1} \vartheta \|_2^2 + \lambda \| \vartheta \|_{\ell_p} \right\}
\]

(2)

Here, the first term is a penalty that represents the closeness of the solution to the observed scene and quantifies the prediction error with respect to the measurements. The second term is a regularization term that represents the priori sparse information of the original scene and also it is designed to penalize an estimate that would not exhibit the expected properties. Also, \( \lambda \) is a regularization parameter that balances the contribution of both terms. In the second term of (2), \( \ell_p \) is usually considered as \( \ell_0 \) or \( \ell_1 \). This minimizing problem can be solved easily by iterative shrinkage/thresholding (IST) methods (see, e.g., [15]), Bregman iterative algorithms (see, e.g., [12] and [13]). Since the \( \ell_0 \) minimization is non-convex and its solution is considered as \( NP-hard \), the common method is to replace \( \ell_0 \) quasi-norm with the \( \ell_1 \) norm, because it is the closest convex norm to non-convex \( \ell_0 \) quasi-norm and minimizing the \( \ell_1 \) norm instead. However, a fact that is often neglected is, for some practical problems, i.e., image inverse problems, the conditions guaranteeing the equivalence of \( \ell_0 \) minimization and \( \ell_1 \) minimization are not necessarily satisfied.

B. Split Bregman Iteration

The split Bregman iteration (SBI) method was recently proposed by Goldstein and Osher [12] for effectively solving \( \ell_1 \)-regularized optimization problem with multiple \( \ell_1 \)-regularized terms [13]. The basic idea of SBI is to convert the unconstrained minimization problem into a constrained one by introducing the variable splitting technique and then invoke the Bregman iteration [16] to solve the constrained minimization problem. Another advantage of the SBI is that it has relatively small memory footprint and is easy to program by users [13], which such properties are very attractive for large-scale problems.

Consider an unconstrained optimization problem

\[
\min_{v \in \mathbb{R}^N} \{ F_1(v) + F_2(Gv) \}
\]

where \( G \in \mathbb{R}^{M \times N} \), \( F_1 : \mathbb{R}^N \rightarrow \mathbb{R} \) and \( F_2 : \mathbb{R}^M \rightarrow \mathbb{R} \). The SBI frameworks is presented in Table I.

C. Dictionary Learning

As stated previously, the key of the sparse representation modeling lies in the choice of sparsifying basis (or dictionary) \( D \). In this context \( D = [d_1, d_2, \cdots, d_t] \in \mathbb{R}^{N \times t} \) is called a dictionary and each of its columns is called an atom. One crucial problem in a sparse-representation problem is how to choose the efficient dictionary. There are many pre-specified (non-adaptive) sparsifying dictionaries (basis or frame), e.g., Fourier, discrete cosine transform, wavelets, Ridgelets, Curvelets, Contourlets, Shearlets and etc. Thought being simple and having fast computations, the non-adaptive dictionaries are not able to efficiently (sparsely) represent a given class of signals. However, learning the atoms from a set of training signals belonging to the signal class of interest, would result in dictionaries with the capability of better matching the content of the signals [17]. It has been experimentally shown that these adaptive dictionaries outperform the non-adaptive ones in many signal processing applications.

Dictionary learning algorithms iteratively perform the two stages of sparse coding and dictionary update. In the first stage, which is actually the clustering of the signals into a union subspace, the sparse approximation of the signals are computed using the current dictionary. The second stage is
the update of the dictionary. Up to our best knowledge, most of dictionary learning algorithms differ mainly in the way of updating the dictionaries [18]-[20]. Some algorithms such as K-singular value decomposition (K-SVD) [19] are based on updating the atoms one-by-one, while some others such as the method of optimal directions (MOD) [18] update the whole set of atoms at once. In [20], a MOD-like algorithm was proposed in which more than one atom along with the non-zero entries in their associated row vectors in coefficient matrix are updated at a time. We refer to this algorithm as the multiple dictionary update (MDU) algorithm (for further reading see [20]).

### III. THE PROPOSED METHOD

By contrast with the conventional/distributed video coding scheme, in which data acquiring and compression tasks are performed disjointly, CVS employs both data acquiring (video sensing) and compression into a unified task, which emerges a new procedure to directly acquiring compressed video data via random projection (without temporally storing the complete raw data) for each individual frame (or blocks of frame) at a low complexity encoder. In this case, the majority of computational burden is shifted from the encoder to the decoder side, which is more suitable to deploy in modern video applications, e.g., wireless multimedia sensor networks. In this section, first we discuss about the encoding of the proposed method, and then we propose the decoding scheme of key and non-key frames.

#### A. Encoding

As mentioned before, the proposed method, firstly, divides the video sequence into the key and non-key frames followed by dividing each frame, of size $I_r \times I_c$, into small non-overlapping blocks of equal size (i.e., size $B \times B$), and then the same sensing (sampling/measurement) matrix $\Phi_B$\(^1\) (i.e., size $m_B \times B^2$, where $m_B = \lceil n B^2 \rceil$, $n = I_r I_c$) is applied for sampling of each block. In this case, we have:

\[
f_i = \Phi_B u_i \tag{4}
\]

where, here, $u_i$ is a (column) vector representing of block $i$ of the input image, $f_i$ is its corresponding measurement vector and $\Phi_B$ is an orthonormalized i.i.d Gaussian matrix. Using this technique has several benefits comparing to use of a random sampling operator to the entire image; i.e., the encoder does not need to wait until the entire image is measured, but each block is sent after its linear projection. In addition, at the decoder side, each block is processed independently; therefore the speed of encoding and decoding procedure is increased. Also in this case, we just need to store a $m_B \times B^2$ sensing matrix instead of a $m \times n$ sensing matrix.

#### B. Recovery of key frame

At the decoder side, the key frame is reconstructed initially using method of the multi-hypothesis block CS recovery [11], in order to use as the initial training image for the process of learning an ALS basis (dictionary). Now, in this case, by considering $u = Do$, The equation (2) using ALS basis can be written as:

\[
\min_{\alpha,v} \frac{1}{2} \| f - \Phi Do \|_{\ell_2}^2 + \lambda \| \alpha \|_{\ell_0}. \tag{5}
\]

Here, $D$ replaces $\Psi^{-1}$ in equation (2), standing for ALS basis and $\Phi = \text{diag}(\Phi_1, \cdots, \Phi_B)$. Also, $\alpha$ denotes the patch-based redundant sparse representation for the whole image over $D$, which can be find by patch-based redundant sparse recovery method (i.e., see [19]). As discussed previously, when $\ell_p$ is replaced with $\ell_0$, since $\ell_0$ is non-convex and NP-hard, the usual routine is to solve its optimal convex approximation, i.e., $\ell_1$ minimization. However, for some practical problems, i.e., image inverse problems, the conditions guaranteeing the equivalence of $\ell_0$ minimization and $\ell_1$ minimization are not necessarily satisfied. Now, let’s go back to equation (5) and point out how to solve it efficiently. By considering $v = Do$ (where, $D$ is the ALS basis) and $\ell_p = \ell_0$, the equation (5) can be formulated as:

\[
\min_{\alpha,v} \frac{1}{2} \| f - \Phi v \|_{\ell_2}^2 + \lambda \| \alpha \|_{\ell_0}. \tag{6}
\]

In order to solve the above minimization problem, an alternating SBI algorithm (as illustrated in Table. 1) is applied. A similar approach was proposed in [14], where equation (6) can be efficiently solved via ALS basis and $\ell_0$ minimization. In this part, we adopt the proposed scheme of [14] to solve (6). Also, in our scheme, we investigate three well-known dictionary learning algorithms (e.g., K-ŠVĐ, MOD and MDU) to adopt in our scheme for finding an efficient solution for equation (6), which leads to a better recovery of the key frames.

#### C. Recovery of non-key frame

While spatial domain compression is performed by CS, the temporal redundancy is not exploited fully, because no motion estimation and compensation is performed at the CVS encoder. To incorporate the temporal redundancy, in order to efficient recovery of the non-key frames, the temporal correlation between adjacent frames is exploited through the inter-frame sparsity model. Here, we adopt an iterative approach for the reconstruction of the non-key frames, where the approach initially estimates an approximation of the non-key frame using the previous reconstructed frame, in order to take the advantage of the inherent temporal structure between successive frames. Then, we utilize the initial estimated frame in an optimization problem to recover and refine the current non-key frame. Suppose that $u_{t-1} = Do_{t-1}$, where $u_{t-1}$ is the previous reconstructed frame using ALS basis $D$, and $o_{t-1}$ denotes the patch-based redundant sparse representation for the whole previous reconstructed frame over $D$. The initialization step can be formulated as:

\[
\min_{\alpha} \frac{1}{2} \| f - \Phi o \|_{\ell_2}^2 + \lambda \| \alpha \|_{\ell_1} + \tau \| \alpha - o_{t-1} \|_{\ell_1}, \tag{7}
\]

where the second term promotes sparsity in the spatial transform of current frames, and the third term promotes sparsity in the inter-frame difference to achieve the temporal redundancy between the current frame and the previous reconstructed frame. Assume that $v = Do$, then (7) can be formulated as:

\[
\min_{\alpha,v} \frac{1}{2} \| f - \Phi v \|_{\ell_2}^2 + \lambda \| \alpha \|_{\ell_1} + \tau \| \alpha - o_{t-1} \|_{\ell_1} \quad \text{s.t.} \quad v = Do. \tag{8}
\]
The above equation can be solved easily by SBI framework. We utilize the solution of (8) into the refinement optimization problem to enhance the recovery of current frame. Suppose \( v^* \) is the solution of (8), then the refinement optimization problem can be formulated as:

\[
\{ u^*_t, \{ \alpha_t^* \}_{t=1}^T, D \} = \min_{u_t, \{ \alpha_t \}, D_t} \left\{ \frac{1}{2} \| f - \Phi u_t \|_2^2 + \frac{1}{2} \| u_t - v^* \|_2^2 + \sum_{l=1}^j \mu_l \| \alpha_t \|_2 \| + \frac{\lambda}{2} \sum_{l=1}^j \| D_t \alpha_t - R_t u_t \|_2 \right\},
\]

(9)

where \( \mu_l > 0 \) are some regularization parameters that control the image patch sparsity. Besides, \( \lambda \) is weight parameter controls the trade-off between the data fidelity and the image prior. Here, \( D_t \) is the ALS basis of current frame that can be learned (initially using \( D \) and \( v^* \)) and updated via the methods of [18]-[20].

IV. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the proposed method. To evaluate our simulation results, we use two applicable quality assessors, the pick signal-to-noise (PSNR) in dB and the structural similarity (SSIM) [21]. For an 8 bits gray scale image (frame) of size \( I_r \times I_c \), PSNR is calculated as

\[
PSNR = 20 \log_{10} \frac{\sqrt{I_r \times I_c \times 255}}{\| u - \hat{u} \|},
\]

(10)

where \( u \) and \( \hat{u} \) denote the original image and the reconstructed image, respectively.

SSIM [21] is another assessor for measuring the similarity between two images, and it is near to human eye perception. Here, the obtained results are evaluated by a specific form of the SSIM index:

\[
SSIM(u, \hat{u}) = \frac{(2\mu_u \mu_{\hat{u}} + c_1)(2\sigma_{u \hat{u}} + c_2)}{\mu_u^2 + \mu_{\hat{u}}^2 + c_1(\sigma_u^2 + \sigma_{\hat{u}}^2 + c_2)}
\]

(11)

where \( \mu_u \) (\( \mu_{\hat{u}} \)) and \( \sigma_u \) (\( \sigma_{\hat{u}} \)) denote the average and variance of \( u \) (\( \hat{u} \)), respectively. \( \sigma_{u \hat{u}} \) is the covariance of \( u \) and \( \hat{u} \). \( c_1 \) and \( c_2 \) are two variables to stabilize the division with weak denominator, such that \( c_1 = (k_1 L)^2 \), \( c_2 = (k_2 L)^2 \) and \( 0 < k_1, k_2 \ll 1 \). \( L \) is the dynamic range of pixel values (255 for 8 bits gray scale images). For our experiments, we use \( k_1 = 0.01 \) and \( k_2 = 0.03 \). All experiments were performed using MATLAB 2013a, on a computer equipped with Intel® core(TM) i7, 3.07 GHz processor, with 48 GB of RAM, and running on Windows 7. The performances of our experiments are evaluated on two well-known video test sequences (e.g., “Foreman” and “Coastguard”) with a CIF resolution of 352 x 288 pixels. Also, in all experiments we use the block size of 32 x 32 and the size of each patch is set to 8 x 8. Note that, in this paper we do not consider the quantization and entropy encoding of measurements, since they are beyond the scope of this paper.

A. Experiment 1

In this first experiment, we evaluate the effectiveness of the discussed dictionary learning methods of [18]-[20] for the proposed key frame recovery algorithm. The key frame is reconstructed initially using the method of [11], in order to use as the initial training image in the process of learning an ALS basis (dictionary). Here, we used the implementation codes obtained from authors for [11]. The ALS basis is obtained by each of the methods provided in [18]-[20], in which the default parameter setting is as follows: the size of sparsifying basis (dictionary) is 256 and number of training iteration is 20. Also, in corresponding recovery problems (i.e., see (6)) \( \lambda \) is set empirically and \( \mu \) is set to be \( 2.5 \times 10^{-5} \) (in the SBI framework). In Fig. 1, a twenty-time-iteration of our method is illustrated as an example to show the performance of MOD [18], K-SVD [19] and MDU [20] as the dictionary learning methods, for recovery of 21st frame of “Foreman” with different \( MR_{R,K} \). In Fig. 1, \( i \) shows the average time of dictionary learning at each iteration. The results obviously show that MDU provides a better recovery performance (in quality) compared to the mentioned methods, but with the cost of higher computational complexity. The extra cost is only generated from dictionary update step in each iteration. Also, the PSNR performance of K-SVD and MOD are very close to each other, but the first one gains lower computational complexity. So, it seems reasonable to adopt K-SVD as the dictionary learning algorithm in our method. The same experiments were run on “Coastguard” sequence (with fast motion scene) and the obtained results proved the accuracy of our assumption. Fig. 2 shows the decoding of the 21st frame of “Foreman” produced by 2D DDWT intra-frame decoder (with 3 levels of decomposition) [Fig. 2(b)], the multi-hypothesis [11] decoder [Fig. 2(c)] and the proposed key frame recovery method using K-SVD [Fig. 2(d)], MOD [Fig. 2(e)] and MDU [Fig. 2(f)]. Note that, for fair comparison, the same test conditions (the same sensing matrix and the same DDWT sparsifying basis) are used in all experiments. It can be observed that the fixed basis intra-frame decoder and multi-hypothesis-based decoder suffer noticeable performance loss over the whole image, while the proposed key frame recovery method demonstrates a considerable reconstruction quality improvement.

B. Experiment 2

In this experiment, we evaluate the performance of the proposed method (via K-SVD dictionary learning) for decoding of the first 50 frame of the test video sequences, with different measurement ratio \( MR \) scenario. The group of the picture (GOP) size is set to 5 and the default parameter setting is as well as experiment 1. The number of iterations in algorithm is set to 6; it is clear that the higher number of iteration yields slightly better performance in quality of recovered sequence, but at the cost of higher computational complexity. Fig. 3 shows the PSNR and SSIM performance of the proposed method in various measurement ratios, compared with the other mentioned recovery algorithms for “Foreman” and “Coastguard” sequence. The PSNR and SSIM shown in Fig. 3, are averaged over all PSNR and SSIM values of the reconstructed frames. Fig. 3(a) and Fig. 3(b) show the PSNR and SSIM performance of the proposed method, respectively, compared with the 2D DDWT intra-frame and multi-hypothesis (MH) recovery method (\( MR_{K} = MR_{K,K} \)), in various measurement ratios. Fig. 3(a)-(b) are obtained for “Foreman” sequence, while Fig. 3(c)-(d) show the same scenario as used in Fig. 3(a)-(b), but for “Coastguard” sequence. Obviously, the numerical results show that, the proposed


$$MR_{K} = 0.1: \bar{t}_{K-SVD} = 11.41 \text{ sec}, \bar{t}_{MOD} = 12.57 \text{ sec}, \bar{t}_{MDU} = 112.33 \text{ sec}. \quad (b) MR_{K} = 0.3: \bar{t}_{K-SVD} = 15.1 \text{ sec}, \bar{t}_{MOD} = 16.05 \text{ sec}, \bar{t}_{MDU} = 169.8 \text{ sec}. \quad (c) MR_{K} = 0.5: \bar{t}_{K-SVD} = 16.96 \text{ sec}, \bar{t}_{MOD} = 17.78 \text{ sec}, \bar{t}_{MDU} = 203.25 \text{ sec}.$$  

Fig. 1: Performance of different dictionary learning algorithms for the proposed method, on the 21st frame of Foreman sequence with various $MR_{K}$. (a) $MR_{K} = 0.1$: $\bar{t}_{K-SVD} = 11.41$ sec, $\bar{t}_{MOD} = 12.57$ sec, $\bar{t}_{MDU} = 112.33$ sec. (b) $MR_{K} = 0.3$: $\bar{t}_{K-SVD} = 15.1$ sec, $\bar{t}_{MOD} = 16.05$ sec, $\bar{t}_{MDU} = 169.8$ sec. (c) $MR_{K} = 0.5$: $\bar{t}_{K-SVD} = 16.96$ sec, $\bar{t}_{MOD} = 17.78$ sec, $\bar{t}_{MDU} = 203.25$ sec.

Fig. 2: Different decoding of the 21st frame of Foreman with respect to recovery as a key frame ($MR_{K} = 0.1$). (a) Original frame. Reconstructed 21st frame by using (b) The 2D DDWT basis (PSNR=26.82 dB, SSIM=0.783). (c) The multi-hypothesis method [11] (PSNR=28.93 dB, SSIM=0.824). The proposed method via: (d) K-SVD (PSNR=30.72 dB, SSIM=0.875). (e) MOD (PSNR=30.8 dB, SSIM=0.876). (f) MDU (PSNR=30.88 dB, SSIM=0.878).

Fig. 3: PSNR and SSIM performance of proposed method on the first 50 frame of Foreman and Coastguard sequence. Foreman: (a) PSNR vs. MR. (b) SSIM vs. MR. Coastguard: (c) PSNR vs. MR. (d) SSIM vs. MR.

method (in both states, $MR_{K} = MR_{NK}$ and $MR_{K} = 0.5$) gains better performance, in both PSNR and SSIM, compared with the other mentioned methods. By increasing $MR_{K}$, the reconstruction quality will increase, but poor compression is its consequence.

V. Conclusion

The motivation of this paper is to propose a novel framework for CVS. In our proposed method, we employed the ALS basis and $\ell_0$ minimization for recovering of key frames, while non-key frames are recovered firstly by initialing a prediction of the current non-key frame using the previous reconstructed frame (in order to exploit the temporal redundancy), and then adopting the prediction into a proper optimization problem. Also, we investigated the effectiveness of three well-known dictionary learning algorithms (in order to learn an ALS basis), and we found out that MDU provides a better recovery performance (in quality) compared to the K-SVD and MOD, but at the cost of higher computational complexity. We found it reasonable to use the K-SVD as the dictionary learning algorithm in our proposed scheme. The numerical results show the adequacy of our proposed method in CVS, compared to the mentioned methods.

References


