Distributed Compressed Video Sensing with Enhanced Boundary Handling Based on Extended Convolutional Sparse Representation

Ibuki Muta* and Yoshimitsu Kuroki†
* National Institute of Technology (KOSEN), Kurume College, Fukuoka, Japan
E-mail: a4219im@kurume.kosen-ac.jp
† E-mail: Kuroki@kurume-nct.ac.jp

***Abstract*—Distributed Compressed Video Sensing (DCVS) is a video coding framework that shifts computational complexity from the encoder to the decoder, making it suitable for low-power encoding environments. Convolutional Sparse Representation (CSR), which models an image as the sum of convolutions between feature-representing dictionary filters and their corresponding sparse coefficients, has been incorporated into DCVS to enhance reconstruction quality and effectively suppress block artifacts. However, conventional CSR relies on convolution multiplication theorems based on the discrete Fourier transform (DFT), which assumes periodic boundary conditions. This often introduces artifacts near signal boundaries, limiting the representational fidelity of the model and degrading overall image quality. To address this limitation, an extended form of CSR (ECSR) has been proposed, offering more flexible boundary handling and improved representational fidelity. In this work, we present a DCVS method that integrates ECSR to enhance reconstruction accuracy. Experimental results on standard video datasets demonstrate that the proposed method consistently outperforms CSR-based DCVS in terms of PSNR and SSIM, particularly under low compression rates.**

I. Introduction

Distributed Compressed Video Sensing (DCVS) [1] is a video coding framework that combines distributed video coding [2] and compressed video sensing [3]. When applied to video sequences, the frames are divided into groups of pictures (GOPs), each consisting of one key frame (KF) and multiple non-key frames (NKFs). In contrast to conventional video coding, which performs motion estimation and compensation at the encoder, DCVS compresses each NKF by applying a measurement matrix to reduce its dimensionality and simply transmits or stores the resulting compressed data without further processing. The decoder first reconstructs the uncompressed KF as a still image. Then, it generates side information from the KF and uses it to independently reconstruct each compressed NKF. In this way, DCVS shifts the computational burden from the encoder to the decoder, making it well-suited for encoding in resource-constrained environments.

Convolutional Sparse Representation (CSR) [4] extends traditional sparse representation by incorporating convolution operations. It can be expressed as

$$\boldsymbol{s}_{(k)} \approx \sum_{m} \boldsymbol{d}_{\mathrm{sl}(m)} * \boldsymbol{x}_{(k,m)}. \tag{1}$$

In this model, the image $\boldsymbol{s}_{(k)}$ is represented as the sum of convolutions between globally shared dictionary filters $\boldsymbol{d}_{sl(m)}$ and the corresponding sparse coefficient maps $\boldsymbol{x}_{(k,m)}$.

In adjacent video frames, the visual features that constitute the image typically exhibit only minor changes. Leveraging this property, CSR-based DCVS learns a set of dictionary filters from a single KF through convolutional dictionary learning (CDL) [5], a process that extracts representative features of the image. Using the learned dictionary, convolutional sparse coding (CSC) [6] is then applied to the compressed NKFs to estimate sparse coefficient maps that indicate the presence and spatial location of those features. This framework enables motion to be effectively represented as spatial variations in the distribution of sparse coefficients across frames. Furthermore, by avoiding block-wise processing, the proposed approach suppresses block artifacts and enables smoother image reconstruction.

In conventional CSR, convolution operations are typically accelerated using the convolution multiplication theorem based on the discrete Fourier transform (DFT). However, this approach assumes periodic boundary conditions, which can introduce artifacts near signal boundaries and degrade the representational capability of the model. Specifically, circular convolution implicitly extends the signal periodically, resulting in discontinuities at the junctions between repeated signal blocks. These boundary artifacts impair the model's ability to accurately capture natural image structures. To address this limitation, recent studies have proposed a generalized CSR framework [7] that supports more flexible boundary handling and has demonstrated improved representational performance. Although [7] does not explicitly refer to this approach as "extended CSR," we adopt the term ECSR throughout this paper for clarity, as it generalizes the conventional CSR model through enhanced boundary handling strategies.

In this work, we propose a DCVS framework that replaces conventional CSR with ECSR to mitigate boundary artifacts and improve reconstruction quality. Experimental results on standard video datasets demonstrate consistent gains in PSNR and SSIM, particularly at low compression rates.

979-8-3315-7206-8/25/$31.00

II. Background

A. ECSR

ECSR is a generalization of the conventional CSR model, which relies on the convolution multiplication theorem based on DFT. Specifically, ECSR replaces the DFT-based convolution theorem with a set of 46 alternative theorems derived from 42 types of boundary handling [8]. These boundary handlings include two types of generalized discrete Fourier transforms (GDFTs) and 40 types of discrete trigonometric transforms (DTTs). For one-dimensional signals, all of these boundary handlings can be unified under the following convolution form:

$$\boldsymbol{w} = T_{\mathrm{b}}^{-1}\left(M_{\mathrm{d}}T_{\mathrm{d}}\boldsymbol{d} \odot M_{\mathrm{x}}T_{\mathrm{x}}\boldsymbol{x}\right), \tag{2}$$

where $\boldsymbol{w}$, $\boldsymbol{d}$, and $\boldsymbol{x}$ are vectors, and $\boldsymbol{w}$ represents the convolution result between $\boldsymbol{d}$ and $\boldsymbol{x}$. T denotes a GDFT or DTT matrix, and M is an identity matrix possibly with zeros. When applied to two-dimensional signals, (2) can be independently applied along the row and column directions. Using the notation $\mathcal{T} = MT$, the 2D convolution is written as:

$$W = T_{\mathrm{b,r}}^{-1}\left(\mathcal{T}_{\mathrm{d,r}}D\mathcal{T}_{\mathrm{d,c}}^{\mathsf{T}} \odot \mathcal{T}_{\mathrm{x,r}}X\mathcal{T}_{\mathrm{x,c}}^{\mathsf{T}}\right)T_{\mathrm{b,c}}^{-\mathsf{T}}, \tag{3}$$

where the subscripts r and c indicate the application of (2) to the row and column dimensions, respectively. It is worth noting that the standard DFT-based CSR is a special case of (3). In this paper, we adopt a vectorized representation using the vectorization operator $\mathrm{vec}(\cdot)$ and the Kronecker product $\otimes$ [9]. We redefine $\boldsymbol{d} = \mathrm{vec}(D)$, $\boldsymbol{x} = \mathrm{vec}(X)$ and $Q = (Q_c \otimes Q_r)$, and apply this notation consistently throughout the rest of the paper. In this representation, each transform matrix T satisfies the following property:

$$T^{\mathrm{H}} = n(A_{\mathrm{c}} \otimes A_{\mathrm{r}})^2 T^{-1}(B_{\mathrm{c}} \otimes B_{\mathrm{r}})^2, \tag{4}$$

where n is a positive scalar, and A and B are real-invertible diagonal matrices. By incorporating these properties, the CSR model can be reformulated using all 46 convolution theorems as:

$$\begin{aligned}\boldsymbol{s}_{(k)} \approx \sum_{m} T_{\mathrm{b}(k,m)}^{-1}\left\{\mathcal{T}_{\mathrm{d}(k,m)}\boldsymbol{d}_{(k,m)} \odot \mathcal{T}_{\mathrm{x}(k,m)}\boldsymbol{x}_{(k,m)}\right\} \\ \text{s.t. } \boldsymbol{d}_{(k,m)} = P_{(m)}\boldsymbol{d}_{\mathrm{sl}(m)},\end{aligned} \tag{5}$$

where P is a zero-padding matrix that inserts zeros into $\boldsymbol{d}_{\mathrm{sl}(m)}$ to expand it to $\boldsymbol{d}_{(k,m)}$.

Hereafter, for notational simplicity, we omit the subscript m from matrices and vectors in the manner described below. The same convention is applied to k.

$$Q_{(k)}\boldsymbol{u}_{(k)} = \begin{pmatrix} Q_{(k,0)} & & \\ & \ddots & \\ & & Q_{(k,M-1)} \end{pmatrix}\begin{pmatrix} \boldsymbol{u}_{(k,0)} \\ \vdots \\ \boldsymbol{u}_{(k,M-1)} \end{pmatrix}. \tag{6}$$

B. Dictionary Optimization in ECSR

In ECSR, a dictionary optimization problem is typically performed by solving the following minimization problem using the Alternating Direction Method of Multipliers (ADMM) [10]:

$$\underset{\boldsymbol{d},\,\boldsymbol{g}_{\mathrm{b}},\,\boldsymbol{g}_{\mathrm{d}}}{\text{minimize}} \quad \sum_{k,m} f_{\mathrm{d}(k)}\left(\boldsymbol{g}_{\mathrm{b}(k)}\right) + i_{C(m)}\left(\boldsymbol{g}_{\mathrm{d}(m)}\right) \tag{7}$$

$$\begin{aligned}\text{s.t.} \quad & G_{\mathrm{b}(k)}X_{\mathrm{rep}(k)}\boldsymbol{d}_{(k)} - G_{\mathrm{b}(k)}\boldsymbol{g}_{\mathrm{b}(k)} = G_{\mathrm{b}(k)}\boldsymbol{s}_{(k)}, \\ & G_{\mathrm{d}(k,m)}\boldsymbol{d}_{(k,m)} - G_{\mathrm{d}(k,m)}\boldsymbol{g}_{\mathrm{d}(m)} = \boldsymbol{0}\end{aligned} \tag{8}$$

Where, $G_{\mathrm{b}(k)}$ and $G_{\mathrm{d}(k,m)}$ are real-invertible, and $X_{\mathrm{rep}(k)}$ is defined as follows:

$$X_{\mathrm{rep}(k)} = T_{\mathrm{b}(k)}(I \quad I \quad \cdots)\, H_{(k)}\, \widehat{X}_{(k)}\, \mathcal{T}_{\mathrm{d}(k)}, \tag{9}$$

$$\widehat{X}_{(k,m)} = \mathrm{diag}\left(\mathcal{T}_{\mathrm{x}(k,m)}\, \boldsymbol{x}_{(k,m)}\right), \tag{10}$$

$$H_{(k,m)} = \pm I. \tag{11}$$

In addition, $i_{C_{(m)}} \in \{0, \infty\}$ denotes an indicator function defined with

$$C_{(m)} = \left\{\boldsymbol{u} \;\middle|\; (I - P_{(m)}P_{(m)}^{\top})\boldsymbol{u} = \boldsymbol{0},\ \|\boldsymbol{u}\|_2 \le 1\right\}. \tag{12}$$

C. Coefficient Optimization in ECSR

A coefficient optimization problem is typically performed by solving the following minimization problem using ADMM:

$$\underset{\boldsymbol{x},\,\boldsymbol{g}_{\mathrm{b}},\,\boldsymbol{g}_{\mathrm{x}}}{\text{minimize}} \quad \sum_{k,m} f_{\mathrm{x}(k)}\left(\boldsymbol{g}_{\mathrm{b}(k)}\right) + \lambda_{(k,m)}\left\|\boldsymbol{g}_{\mathrm{x}(k,m)}\right\|_1 \tag{13}$$

$$\begin{aligned}\text{s.t. } & G_{\mathrm{b}(k)}D_{\mathrm{rep}(k)}\boldsymbol{x}_{(k)} - G_{\mathrm{b}(k)}\boldsymbol{g}_{\mathrm{b}(k)} = G_{\mathrm{b}(k)}\boldsymbol{s}_{(k)}, \\ & G_{\mathrm{x}(k,m)}\boldsymbol{x}_{(k,m)} - G_{\mathrm{x}(k,m)}\boldsymbol{g}_{\mathrm{x}(k,m)} = \boldsymbol{0}\end{aligned} \tag{14}$$

Where, $D_{\mathrm{rep}(k)}$ is defined as follows:

$$D_{\mathrm{rep}(k)} = T_{\mathrm{b}(k)}\,(I \quad I \quad \cdots)\, H_{(k)}\, \widehat{D}_{(k)}\, \mathcal{T}_{\mathrm{x}(k)}, \tag{15}$$

$$\widehat{D}_{(k,m)} = \mathrm{diag}\left(\mathcal{T}_{\mathrm{d}(k,m)}\, P_{(m)}\, \boldsymbol{d}_{\mathrm{sl}(m)}\right), \tag{16}$$

$$H_{(k,m)} = \pm I. \tag{17}$$

Here, $\lambda_{(k,m)}$ is a positive scalar that controls the sparsity of the coefficients.

D. CSR-based DCVS

The encoder transmits the KF $\boldsymbol{k} \in \mathbb{R}^{N^2}$ without compression, while each NKF $\boldsymbol{n} \in \mathbb{R}^{N^2}$ is compressed by applying a random measurement matrix $\phi = (\Phi \otimes \Phi)$, where $\Phi \in \mathbb{R}^{L \times N}$. This process produces a compressed measurement $\boldsymbol{f} \in \mathbb{R}^{L^2}$.

$$\boldsymbol{f} = \phi\boldsymbol{n}. \tag{18}$$

At the decoder, dictionary learning is first performed using the KF. In (7) and (13), the ℓ_1-norm is used as the error term to ensure robustness against outliers, and a DFT-based formulation is applied. The dictionary is then obtained by solving the CDL optimization problem. Since CDL involves both dictionary optimization and coefficient optimization, solving them jointly is generally intractable. Therefore, an alternating optimization strategy is adopted, where dictionary optimization

and coefficient optimization are performed iteratively by fixing one while updating the other.

Next, given the learned dictionary filters $\boldsymbol{d}_{\mathrm{sl}(m)}$ and the compressed images $\boldsymbol{f}$, the sparse coefficients $\boldsymbol{x}_{(m)}$ are estimated such that the original NKFs $\boldsymbol{n}$ can be represented as a sum of convolutions with $\boldsymbol{d}_{(m)}$. The sparse coefficients are obtained by solving the following minimization problem:

$$\arg\min_{\boldsymbol{x}} \|D_{\mathrm{rep}}\boldsymbol{x} - \boldsymbol{f}\|_2^2 + \mu \|\boldsymbol{x}\|_1 . \tag{19}$$

Where, μ is a parameter that controls the sparsity of the coefficients, and D_{rep} and $\boldsymbol{x}$ is defined as follows using $\mathcal{F}$, the Kronecker product of DFT transform matrices.

$$D_{\mathrm{rep}} = \phi\mathcal{F}^{-1}\begin{pmatrix} I & I & \dots & I \end{pmatrix} \hat{D} \begin{pmatrix} \mathcal{F} & & & \\ & \ddots & & \\ & & & \mathcal{F} \end{pmatrix}, \tag{20}$$

$$\hat{D} = \begin{pmatrix} \mathrm{diag}(\hat{\boldsymbol{d}}_{(1)}) & & & \\ & \mathrm{diag}(\hat{\boldsymbol{d}}_{(2)}) & & \\ & & \ddots & \\ & & & \mathrm{diag}(\hat{\boldsymbol{d}}_{(M)}) \end{pmatrix}, \tag{21}$$

$$\hat{\boldsymbol{d}}_{(m)} = \mathcal{F}\, P_{(m)}\, \mathbf{d}_{\mathrm{sl}(m)}, \tag{22}$$

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_{(1)} \\ \boldsymbol{x}_{(2)} \\ \vdots \\ \boldsymbol{x}_{(M)} \end{pmatrix}. \tag{23}$$

Equation (19) is jointly convex and consists of the sum of a differentiable and a non-differentiable function. Therefore, it is solved using the proximal gradient method [11]. By defining the functions

$$f(\boldsymbol{x}) = \|D_{\mathrm{rep}}\boldsymbol{x} - \boldsymbol{f}\|_2^2, \quad g(\boldsymbol{x}) = \mu \|\boldsymbol{x}\|_1 ,$$

the following optimization problem is obtained:

$$\arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + g(\boldsymbol{x}). \tag{24}$$

The update equation for (24) is expressed as follows:

$$\boldsymbol{x}^{(j+1)} = \mathrm{prox}(\boldsymbol{z}), \tag{25}$$

$$\boldsymbol{z} = \boldsymbol{x}^{(j)} - \rho\nabla f(\boldsymbol{x}^{(j)}), \tag{26}$$

$$\mathrm{prox}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} g(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2. \tag{27}$$

To solve (19), the gradient $\nabla f(\boldsymbol{x})$ and the proximal operator prox_g are computed as follows:

$$\nabla f(\boldsymbol{x}) = \begin{pmatrix} \mathcal{F}^{-1} diag\left(\hat{\boldsymbol{d}_{(1)}}\right)^H \mathcal{F}\phi^T \left(D_{\mathrm{rep}}\boldsymbol{x} - \boldsymbol{f}\right) \\ \vdots \\ \mathcal{F}^{-1} diag\left(\hat{\boldsymbol{d}_{(M)}}\right)^H \mathcal{F}\phi^T \left(D_{\mathrm{rep}}\boldsymbol{x} - \boldsymbol{f}\right) \end{pmatrix}, \tag{28}$$

$$\mathrm{prox}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2. \tag{29}$$

The solution to (27) is obtained via soft-thresholding.

$$\mathrm{prox}(\boldsymbol{z}) = \mathrm{sign}(\boldsymbol{z}) \odot \max\left(0, |\boldsymbol{z}| - \frac{\mu}{\rho}\right). \tag{30}$$

The original NKF can be reconstructed by summing the convolutions of the dictionary filters with the corresponding sparse coefficients.

III. Proposed Method

In this study, we propose a DCVS framework based on ECSR. By employing the discrete cosine transform Type-I even (C_{1e}), which enables symmetric boundary extension and simplifies the formulation, the proposed method effectively suppresses artifacts at signal boundaries and enhances reconstruction quality.

The encoder-side processing remains identical to that of the conventional approach.

At the decoder, dictionary learning is first performed using the KF. In (7) and (13), the ℓ_1-norm is used as the error term to ensure robustness against outliers, and C_{1e} is applied for boundary handling. The dictionary filters are then obtained by solving the CDL optimization problem.

The sparse coefficients $\boldsymbol{x}_{(m)}$ are then estimated using the learned dictionary filters $\boldsymbol{d}_{\mathrm{sl}(m)}$ and the compressed image $\boldsymbol{f}$.

$$\arg\min_{\boldsymbol{x}} \|D_{\mathrm{rep}}\boldsymbol{x} - \boldsymbol{f}\|_2^2 + \mu \|\boldsymbol{x}\|_1 . \tag{31}$$

Where, μ is a parameter that controls the sparsity of the coefficients, and D_{rep} and $\boldsymbol{x}$ is defined as follows using $\mathcal{C}$, the Kronecker product of C_{1e} transform matrices.

$$D_{\mathrm{rep}} = \phi\mathcal{C}^{-1}\begin{pmatrix} I & I & \dots & I \end{pmatrix} \hat{D} \begin{pmatrix} \mathcal{C} & & & \\ & \ddots & & \\ & & & \mathcal{C} \end{pmatrix}, \tag{32}$$

$$\hat{D} = \begin{pmatrix} \mathrm{diag}(\hat{\boldsymbol{d}}_{(1)}) & & & \\ & \mathrm{diag}(\hat{\boldsymbol{d}}_{(2)}) & & \\ & & \ddots & \\ & & & \mathrm{diag}(\hat{\boldsymbol{d}}_{(M)}) \end{pmatrix}, \tag{33}$$

$$\hat{\boldsymbol{d}}_{(m)} = \mathcal{C}\, P_{(m)}\, \mathbf{d}_{\mathrm{sl}(m)}, \tag{34}$$

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_{(1)} \\ \boldsymbol{x}_{(2)} \\ \vdots \\ \boldsymbol{x}_{(M)} \end{pmatrix}. \tag{35}$$

Equation (31) is jointly convex and consists of the sum of a differentiable and a non-differentiable function. Therefore, it is solved using the proximal gradient method. By defining the functions

$$f(\boldsymbol{x}) = \|D_{\mathrm{rep}}\boldsymbol{x} - \boldsymbol{f}\|_2^2, \quad g(\boldsymbol{x}) = \mu \|\boldsymbol{x}\|_1 ,$$

the following optimization problem is obtained:

$$\arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + g(\boldsymbol{x}). \tag{36}$$

The update equation for (36) is expressed as follows:

$$\boldsymbol{x}^{(j+1)} = \mathrm{prox}(\boldsymbol{z}), \tag{37}$$

$$\boldsymbol{z} = \boldsymbol{x}^{(j)} - \rho \nabla f(\boldsymbol{x}^{(j)}), \tag{38}$$

$$\mathrm{prox}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} g(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2. \tag{39}$$

To solve (31), the gradient $\nabla f(\boldsymbol{x})$ and the proximal operator prox_g are computed as follows:

$$\nabla f(\boldsymbol{x}) = \begin{pmatrix} \Lambda \mathcal{C}^{-1} \Lambda^{-1} \mathrm{diag}\left(\hat{\boldsymbol{d}}_{(1)}\right)^H \Lambda \mathcal{C} \Lambda^{-1} \boldsymbol{\phi}^T \left(D_{\mathrm{rep}} \boldsymbol{x} - \boldsymbol{f}\right) \\ \vdots \\ \Lambda \mathcal{C}^{-1} \Lambda^{-1} \mathrm{diag}\left(\hat{\boldsymbol{d}}_{(M)}\right)^H \Lambda \mathcal{C} \Lambda^{-1} \boldsymbol{\phi}^T \left(D_{\mathrm{rep}} \boldsymbol{x} - \boldsymbol{f}\right) \end{pmatrix}, \tag{40}$$

$$\mathrm{prox}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2. \tag{41}$$

Here, Λ and Λ^{-1} in (40) are defined as the following matrices:

$$\Lambda = \begin{pmatrix} \frac{1}{2} & & \\ & \ddots & \\ & & \frac{1}{2} \end{pmatrix}, \quad \Lambda^{-1} = \begin{pmatrix} 2 & & \\ & \ddots & \\ & & 2 \end{pmatrix}. \tag{42}$$

The solution to (39) is obtained via soft-thresholding.

$$\mathrm{prox}(\boldsymbol{z}) = \mathrm{sign}(\boldsymbol{z}) \odot \max\left(0, |\boldsymbol{z}| - \frac{\mu}{\rho}\right). \tag{43}$$

IV. Experiments

We evaluate the performance of the proposed method on three standard video datasets: Foreman, Akiyo, and Coastguard. Each frame in the datasets is a grayscale image with a spatial resolution of 128×128 pixels (that is, $N = 128$). A single GOP is constructed using the first 150 frames, where the first frame serves as the KF, and the remaining 149 frames are treated as NKFs. In all experiments, we used $M = 10$ dictionary filters, each of size 12×12. To ensure a fair comparison between the proposed and conventional methods, the regularization parameter λ was chosen such that the ℓ_0-norm of the learned sparse coefficients after dictionary learning was approximately one-tenth of the total number of elements. The number of ADMM iterations was fixed at 1000 for both dictionary learning and coefficient optimization. The reconstruction quality of the 149 NKFs was evaluated under varying compression ratios defined as $MR = L^2/N^2$, where MR was set to 0.2, 0.5, or 0.7. For each dataset and each MR value, the parameter μ was tuned to maximize the PSNR [dB] of a randomly selected NKF during decoding. During decoding, the learning rate for both dictionary learning and coefficient estimation was empirically set to ensure convergence. Reconstruction performance was quantitatively evaluated using PSNR and SSIM, computed over the 149 NKFs.

Table I summarizes the mean and standard deviation of PSNR and SSIM for each dataset and measurement rate (MR). It also reports the improvements (ΔPSNR and ΔSSIM) achieved by the proposed method over the conventional approach. Frame-by-frame PSNR and SSIM results are presented in Fig. 1.

TABLE I: Mean and standard deviation of PSNR and SSIM for each dataset and compression ratio (MR). ΔPSNR and ΔSSIM represent improvements over the conventional method.

Dataset	MR	Method	PSNR [dB] (± std)	SSIM (± std)	ΔPSNR	ΔSSIM
Foreman	0.2	Conventional	22.18 ± 0.17	0.613 ± 0.014	–	–
		Proposed	24.68 ± 0.35	0.693 ± 0.010	+2.50	+0.079
Foreman	0.5	Conventional	26.77 ± 0.23	0.748 ± 0.013	–	–
		Proposed	28.27 ± 0.43	0.808 ± 0.008	+1.50	+0.060
Foreman	0.7	Conventional	29.30 ± 0.24	0.820 ± 0.010	–	–
		Proposed	29.72 ± 0.43	0.848 ± 0.007	+0.42	+0.027
Akiyo	0.2	Conventional	23.22 ± 0.08	0.562 ± 0.003	–	–
		Proposed	24.64 ± 0.08	0.634 ± 0.004	+1.42	+0.072
Akiyo	0.5	Conventional	26.16 ± 0.06	0.687 ± 0.003	–	–
		Proposed	27.40 ± 0.13	0.742 ± 0.004	+1.24	+0.055
Akiyo	0.7	Conventional	27.80 ± 0.09	0.745 ± 0.002	–	–
		Proposed	29.12 ± 0.12	0.801 ± 0.003	+1.31	+0.055
Coastguard	0.2	Conventional	19.64 ± 0.98	0.439 ± 0.017	–	–
		Proposed	20.74 ± 0.76	0.490 ± 0.017	+1.10	+0.051
Coastguard	0.5	Conventional	22.46 ± 0.74	0.606 ± 0.021	–	–
		Proposed	23.50 ± 0.99	0.650 ± 0.011	+1.03	+0.044
Coastguard	0.7	Conventional	24.81 ± 0.79	0.713 ± 0.014	–	–
		Proposed	25.32 ± 1.07	0.733 ± 0.013	+0.50	+0.020

Across all datasets and compression ratios, the proposed method consistently outperformed the conventional method in terms of reconstruction quality. The performance gap became more pronounced as the compression ratio decreased. Notably, the Coastguard dataset exhibited lower overall PSNR and SSIM values compared to Foreman and Akiyo, likely due to substantial object and camera motion. Nevertheless, the proposed method still achieved consistent improvements over the conventional CSR-based method, even under such challenging conditions.

V. Conclusion

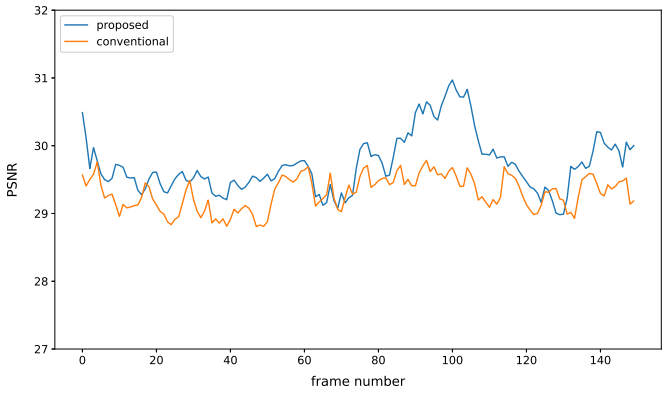
This paper proposed a DCVS method based on ECSR, incorporating improved boundary handling with C_{1e}.

Experiments showed that the method consistently outperformed the conventional CSR-based approach in PSNR and SSIM across all datasets and MR, with notable gains at lower compression ratios. These results confirm the effectiveness of the ECSR-based formulation and underscore the importance of proper boundary handling in DCVS.

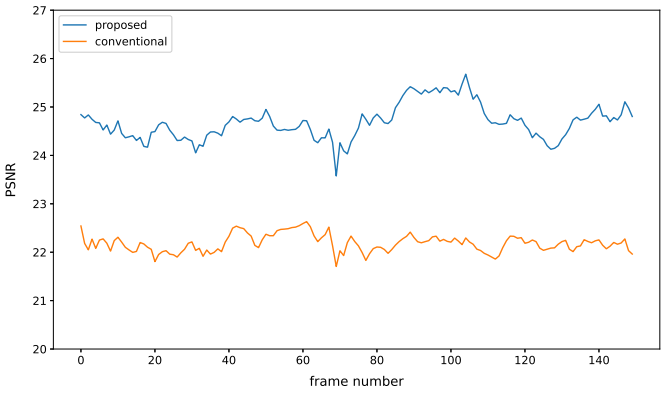
Reconstruction quality in DCVS depends strongly on the learned dictionary, estimated from the KF; thus, KF choice and hyperparameter tuning directly affect decoding performance. Future work includes adaptive KF selection and hyperparameter refinement to further improve accuracy. We also plan to examine alternative boundary handling, more flexible GOP structures for sequences with large motion such as Coastguard, and extend experiments to color videos and larger datasets.

Acknowledgment

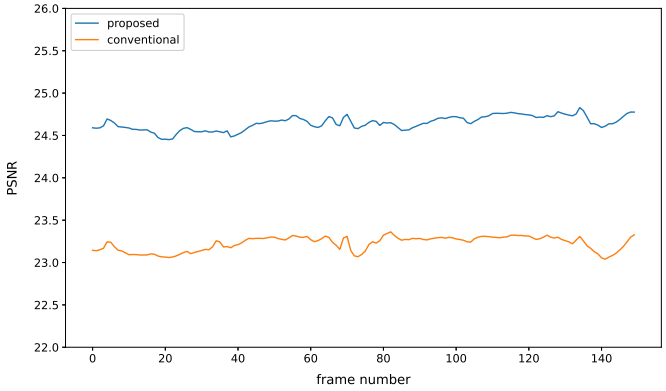
This work was supported by JSPS KAKENHI Grant Number JP23K11159.



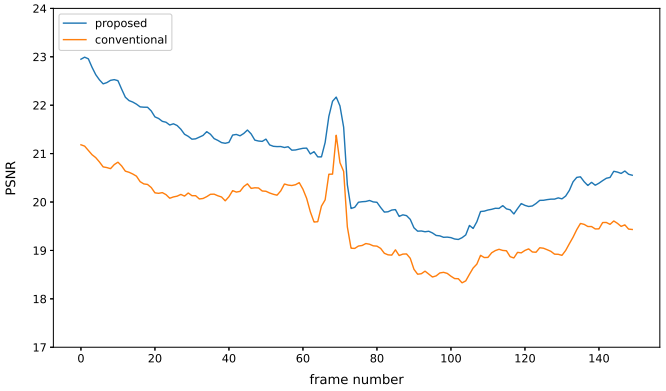
(a) Foreman, PSNR at MR = 0.7



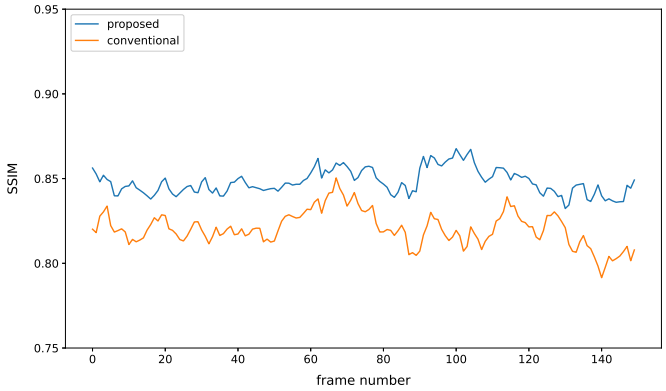
(b) Foreman, PSNR at MR = 0.2



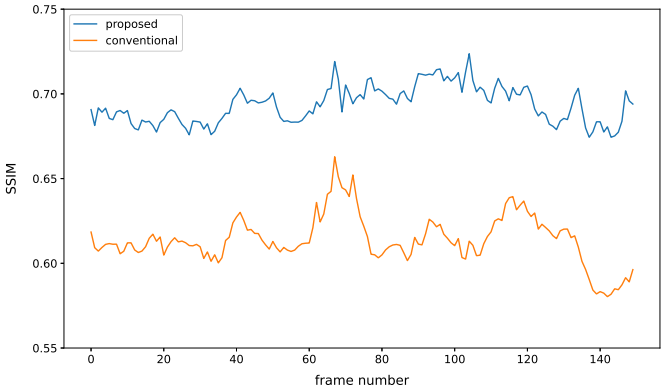
(c) Akiyo, PSNR at MR = 0.2



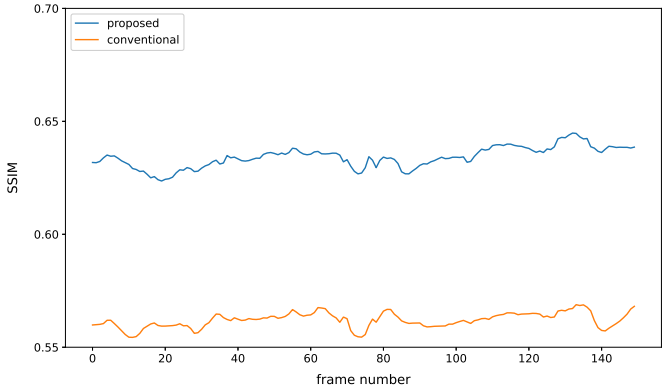
(d) Coastguard, PSNR at MR = 0.2



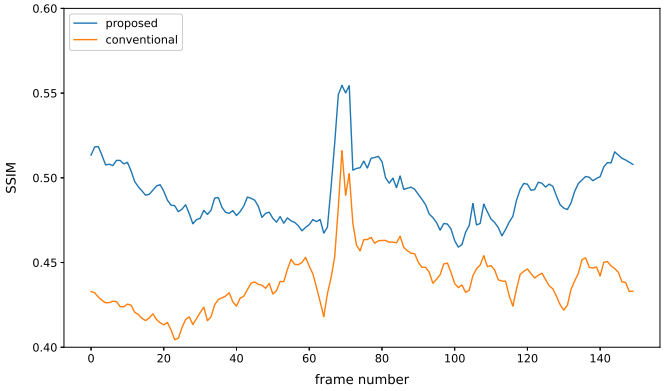
(e) Foreman, SSIM at MR = 0.7



(f) Foreman, SSIM at MR = 0.2



(g) Akiyo, SSIM at MR = 0.2



(h) Coastguard, SSIM at MR = 0.2

Fig. 1: Frame-by-frame PSNR and SSIM at MR = 0.2 (Akiyo, Coastguard) and MR = 0.2/0.7 (Foreman).

REFERENCES

[1] Thong T. Do, Yi Chen, Dzung T. Nguyen, Nam Nguyen, Lu Gan, and Trac D. Tran, "Distributed compressed video sensing," IEEE International Conference on Image Processing (ICIP), Nov. 2009.

[2] B. Girod, A.M. Aaron, S. Rane and D. Rebollo-Monedero, "Distributed Video Coding," Proceedings of the IEEE, vol. 93, pp. 71-83, Jun. 2005.

[3] D. L. Donoho, "Compressed sensing," in IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289-1306, April 2006, doi: 10.1109/TIT.2006.871582.

[4] M. Lewicki and T. Sejnowski, "Coding time-varying signals using sparse, shift-invariant representations." vol. 11, 01 1998, pp. 730–736.

[5] C. Garcia-Cardona and B. Wohlberg, "Convolutional dictionary learning: A comparative review and new algorithms," IEEE Transactions on Computational Imaging, vol. 4, no. 3, pp. 366–381, 2018.

[6] H. Bristow, A. Eriksson, and S. Lucey, "Fast convolutional sparse coding," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2013.

[7] Y. Tsukiashi and Y. Kuroki, "Enhancing boundary handling strategies for convolutional sparse representation model with 46 × 46 convolution multiplication properties," in ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

[8] S. Martucci, "Symmetric convolution and the discrete sine and cosine transforms," IEEE Transactions on Signal Processing, vol. 42, no. 5, pp. 1038–1051, 1994.

[9] S. Liu and G. Trenkler, "Hadamard, khatri-rao, kronecker and other matrix products," International Journal of Information and Systems Sciences, vol. 4, no. 1, pp. 160–177, 2008.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," Foundations and Trends ® in Machine Learning, vol. 3, no. 1, pp. 1–122, 2011. [Online]. Available: http://dx.doi.org/10.1561/2200000016

[11] N. Parikh and S. Boyd, "Proximal Algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.