Augmented Deep Contexts for Spatially Embedded Video Coding

Supplementary Material

This supplementary material document provides additional details of our proposed Spatially Embedded Video Codec (SEVC). The remainder of the supplementary material is divided into three parts. Section 6 gives the detailed network architectures of our proposed modules. Section 7 provides additional comparison results. Section 8 gives the configuration of the traditional codec—VTM-13.2[1].

6. Network Architechture

Our SEVC is implemented based on DCVC-DC [25] but focuses on exploiting additional spatial references for augmenting the contexts and latent prior.

Motion and Feature Co-Augmentation. As shown in Figure 4, the Motion and Feature Co-Augmentation (MFCA) module progressively improves the quality of the base MVs and the spatial feature through several Augment Stages. Figure 12 shows the architecture of one Augment Stage. It takes two steps to augment the base MVs $\bar{v}_i^l$ and spatial feature $\bar{F}_i^l$ within one Augment Stage: Firstly, base MVs $\bar{v}_i^l$, spatial feature $\bar{F}_i^l$, and temporal feature $\hat{F}_{t-1}^l$ are fed into an Augment Unit to generate augmented base MVs $\bar{v}_{i+1}^l$. Secondly, the augmented base MVs $\bar{v}_{i+1}^l$ leads a better alignment of $\hat{F}_{t-1}^l$ and the aligned $\hat{F}_{t-1}^l$ are fed into another Augment Unit with spatial feature $\bar{F}_i^l$ to generate augmented spatial feature $\bar{F}_{i+1}^l$. Figure 13 shows the architecture of one Augment Unit. This example is the Augment Unit for motion augmentation in the largest scale, where $N^l = 48$. Two convolution layers with a stride equal to 2 are used to reduce the resolution and two subpixel layers [48] are used to upsample the residual back to the original resolution.

Spatial-Guided Latent Prior Augmentation The proposed latent prior $\bar{y}_t$ is generated by adding the residual queried from multiple temporal latent representations $\hat{y}_{t-1}, \hat{y}_{t-2}, \hat{y}_{t-3}$ to the upsampled spatial latent $\hat{y}_t^b$. To implement this, two subpixel layers are first used to upsample the spatial latent $\hat{y}_t^b$. The upsampled $\hat{y}_t^b$ and temporal latent representations $\hat{y}_{t-1}, \hat{y}_{t-2}, \hat{y}_{t-3}$ are concatenated and fed into several Residual Swin Transformer Blocks (RSTBs) [29, 47] to generate the residual. Within each RSTB, there are several Swin Transformer Layers (STLs) that utilize 3D window partitions to capture correlation across the spatial and temporal dimensions. We set the head number to 8, the window size to 8, and the embedding dimension to 128. A Swin Transformer Layer with shifted window partitions is denoted as STL-SW. In our SEVC, the Transformers are calculated on low-resolution latent representations, which will not bring too much computation cost. Table 9 gives the complexity comparison when different numbers of temporal latent representations are used for augmentation. It can be observed that introducing more temporal latent representations only results in a linear complexity increase.

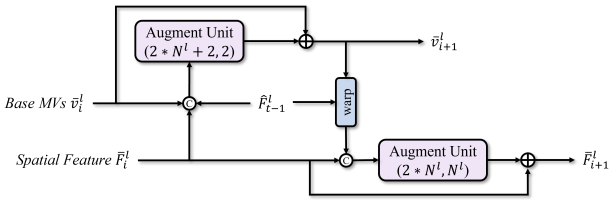

Figure 12. The network architechture of the ith Augment Stage. The numbers in an Augment Unit refer to the number of input channels and number of output channels. N^l refers to the number of channels in the lth scale.

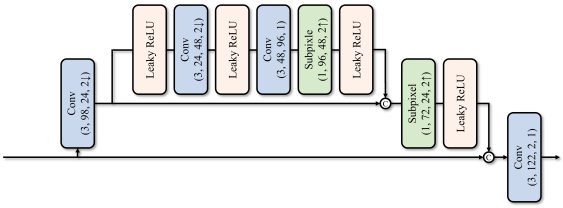

Figure 13. The network architecture of the Augment Unit. The numbers in a Conv block refer to the kernel size, number of input channels, number of output channels, and stride. This example is the Augment Unit for motion augmentation in the largest scale.

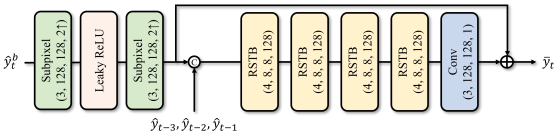

Figure 14. The network architecture of the latent prior generation. The numbers in a Residual Swin Transformer Block (RSTB) [29] refer to the depth, head number, window size, and the embedding dimension.

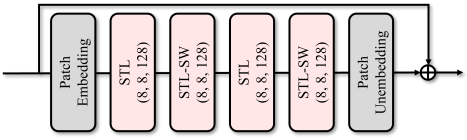

Figure 15. The network architecture of the RSTB. The numbers in a Swin Transformer Layer (STL) refer to the head number, window size, and the embedding dimension. STL-SW indicates STL with shifted window partitions.

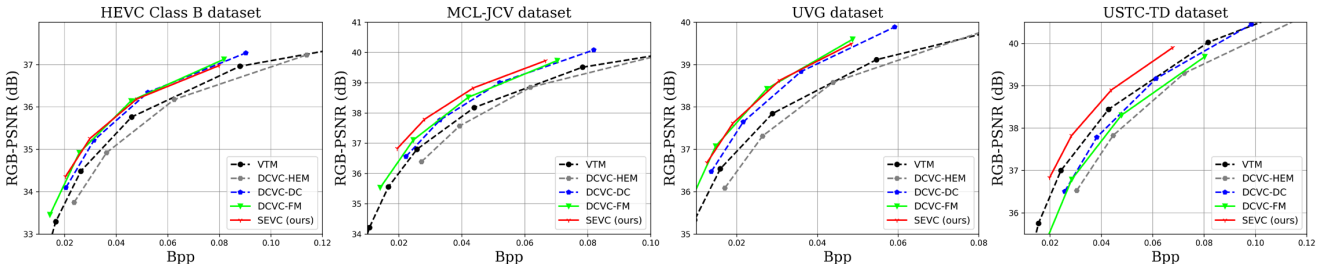

Figure 16. Rate and distortion curves on four 1080p datasets. The Intra Period is –1 with 96 frames.

Table 8. BD-Rate (%) comparison for RGB PSNR with BT.709. The Intra Period is –1 with 96 frames. The anchor is VTM-13.2 LDB.

	HEVC B	MCL-JCV	UVG	USTC-TD	Average
DCVC-HEM [24]	13.4	11.4	12.5	27.1	16.1
DCVC-DC [25]	–13.4	–13.5	–20.6	11.8	–8.9
DCVC-FM [26]	**–18.1**	–15.9	**–27.9**	23.1	–9.7
SEVC (ours)	–16.5	**–24.5**	–27.0	**–14.5**	**–20.6**

Table 9. Complexity Comparison.

ΔT	0	1	2	3	4
MACs	50G	75G	100G	125G	150G

[1] Tested on 1080p sequences.

7. Additional Results

7.1. Results on RGB PSNR with BT.709

When testing RGB videos, we use FFmpeg to convert YUV420 videos to RGB videos, where BT.601 is employed to implement the conversion. However, BT.709 is used in [25, 26] for a higher compression ratio under a similar visual quality. Thus we provide additional results with BT.709 on four 1080p datasets. We focus on high-resolution videos because a 4x downsampling process is conducted in our spatially embedded codec, and the spatial references with too small resolution are meaningless.

Figure 16 and Table 8 give the RD curves and BD-Rate comparisons for four 1080p datasets with BT.709. Compared to PSNR with BT.601 shown in Figure 10, although PSNR with BT.709 is significantly higher than PSNR with BT.601 under the same bpp, the relative bitrate savings compared to VTM are similar. When using BT.601 and compared to VTM, DCVC-DC achieves an average bitrate saving of 8.3%, DCVC-FM achieves 6.2%, and SEVC achieves 23%. When using BT.709 and compared to VTM, DCVC-DC averages 8.9% savings, DCVC-FM averages 9.7%, and SEVC averages 20.6%. The slight performance degradation is attributed to that the selected train set is not constructed by BT.709 conversion.

7.2. Visualization of MVs and contexts

As shown in Table 3, our SEVC performs much better than DCVC-DC in sequences with large motions and significant emerging objects. There are two main reasons for this: On the one hand, the base MVs progressively augmented by our proposed MFCA module have a higher quality than the reconstructed MVs in DCVC-DC, which improves the utilization of temporal references. On the other hand, the augmented spatial feature can provide an additional description for regions with emerging objects that are not well described by temporal references.

As shown in Figure 17, Figure 18, and Figure 19, the augmented MVs in our SEVC has a higher warp PSNR and a better subjective quality compared to reconstructed MVs in DCVC-DC. However, the residuals are still large in regions where new objects appear (marked in red boxes), indicating that the temporal references are not rich enough to describe the emerging objects. Therefore, temporal contexts in DCVC-DC fail to predict the emerging objects well. By contrast, our SEVC utilizes an additional spatial feature, and the augmented spatial feature complements those regions. It can be observed that in the hybrid spatial-temporal contexts, those emerging objects are well described, thus providing a better prediction.

8. Configuration of the Traditional Codec

When testing traditional codec—VTM-13.2 [1], the input video sequences are in YUV444 format to achieve a better compression ratio [24, 25, 45]. The YUV444 video sequences are converted from the RGB video sequences which are used as the inputs of NVCs. The configuration

parameters for encoding each video are as:

- EncoderAppStatic

  -c encoder_lowdelay_vtm.cfg

  –InputFile={*Input File Path*}

  –InputBitDepth=8

  –OutputBitDepth=8

  –OutputBitDepthC=8

  –InputChromaFormat=444

  –FrameRate={*Frame Rate*}

  –DecodingRefreshType=2

  –FramesToBeEncoded=96

  –IntraPeriod={*Intra Period*}

  –SourceWidth={Width}

  –SourceHeight={Height}

  –QP={QP}

  –Level=6.2

  –BitstreamFile={Bitsteam File Path}

  –ReconFile={Output File Path}

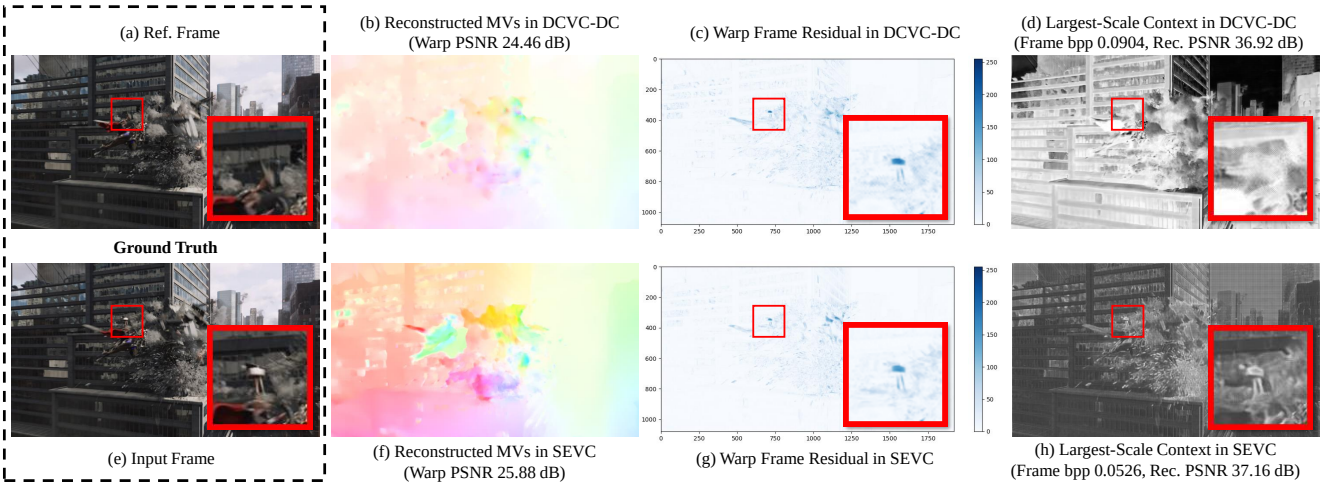

Figure 17. Visualization of the MVs and contexts in DCVC-DC and our SEVC. This example is from *videoSRC22_1920x1080_24* video of MCL-JCV [53].

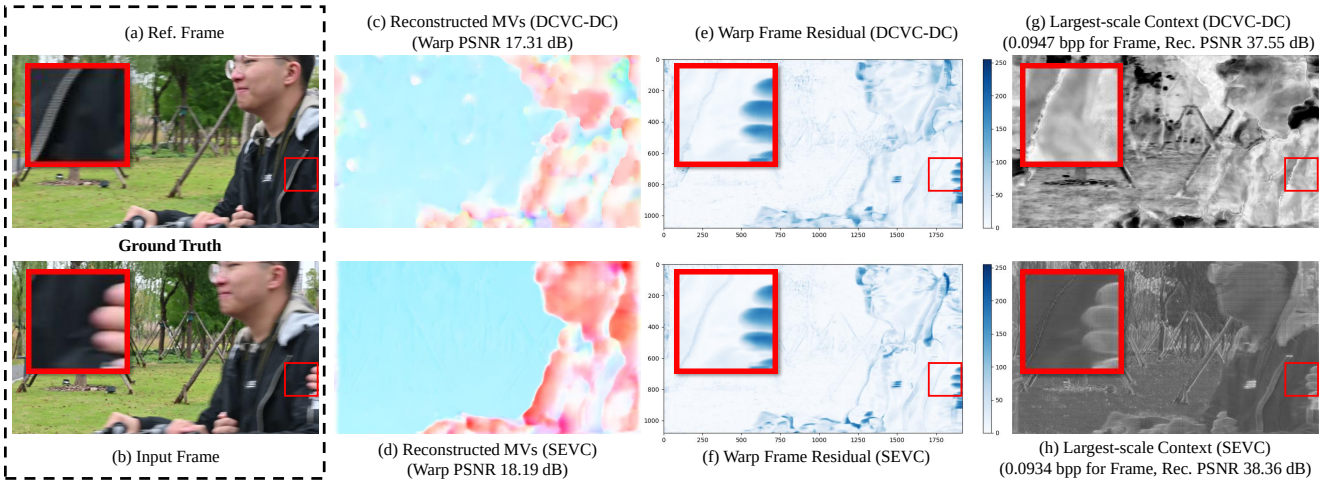

Figure 18. Visualization of the MVs and contexts in DCVC-DC and our SEVC. This example is from *USTC_BycycleDriving* video of USTC-TD [28].

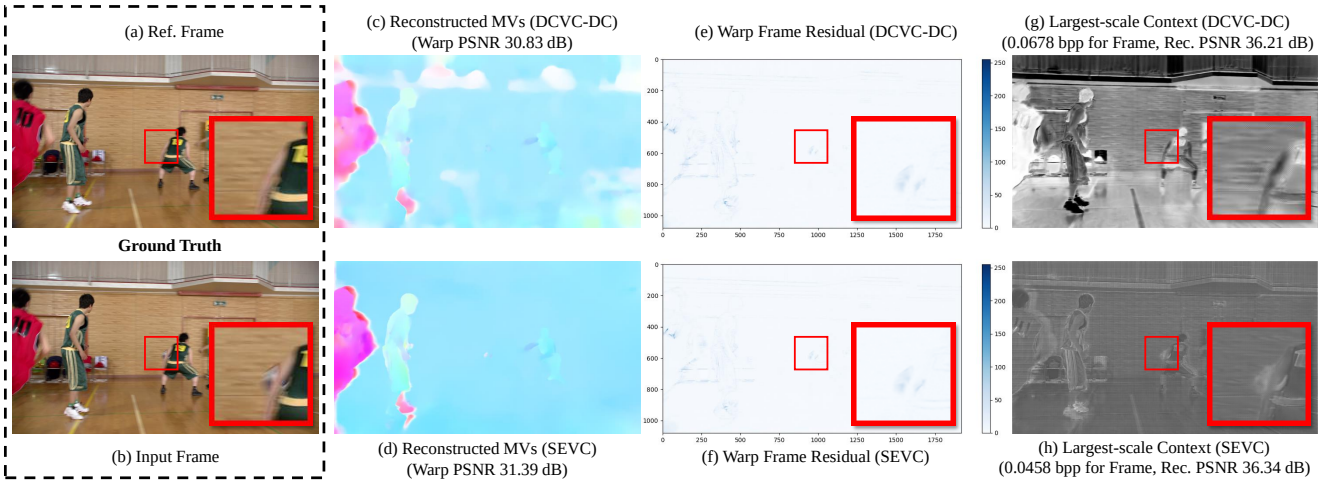

Figure 19. Visualization of the MVs and contexts in DCVC-DC and our SEVC. This example is from *BasketballDrive_1920x1080_50* video of HEVC B [28].