SPLATTING-BASED MOTION CONTEXT ENCODING FOR DEEP VIDEO COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent video compression studies aim to compress videos in a more optimal space using deep neural networks. Most of them employ a strategy where they use motion information to warp the previous frame to align with the current frame, and then only compress the information newly appearing in the current frame. While this enhances the compression efficiency of each frame, additional bits are required to compress the motion information alongside it. In this paper, we explore a methodology that improves motion compression by warping previous motions just like frames. However, within the traditional backward warping-based framework, a dilemma arises where the decoded motion is needed to warp the reference motion. To solve this problem, we propose a forward warping-based framework for video compression called SVC (Splatting-based Video Compression). While SVC offers the advantage of enabling the use of motion context, forward warping has several issues compared to backward warping and we propose additional tricks to address these challenges. Intensive experiments on the UVG, HEVC, and MCL-JCV benchmarks demonstrate that motion context encoding through SVC is indeed more effective compared to various methods based on backward warping, including traditional codecs.

1 INTRODUCTION

Video compression is a well-established area of research that seeks to represent video content with fewer bits without compromising its visual quality. The majority of existing video compression techniques, such as H.264 (Wiegand et al., 2003), H.265 (Sullivan et al., 2012), and H.266 (Bross et al., 2021), aim to increase the compression rate by compressing only the remaining information that is not present in the previous frame. In this context, the frame to be compressed is called the *current frame*, and the frame referred to for removing redundancy is called the *reference frame*. To achieve this, most of the existing techniques follow a three-step process. The first step is **motion estimation**, which involves matching the pixels that exist simultaneously in the current and previous frames. The second step is **motion compensation**, which involves rearranging the pixels of the reference frame using various warping methods so that the reference frame can be aligned with the current frame. Subsequently, only the newly added information in the current frame is encoded through **redundancy reduction** step, instead of compressing each frame independently.

The recent application of deep learning for image and video compression has shifted the compression domain from the conventional Discrete Cosine Transform (DCT) space to a more optimal domain learned end-to-end from large datasets. Furthermore, deep learning-based estimators have replaced traditional block-based motion vectors with dense optical flows (Dosovitskiy et al., 2015; Ranjan & Black, 2017; Sun et al., 2018). These advancements have greatly improved each step of the video compression process and demonstrated performance surpassing that of hand-crafted codecs. Further details on this topic are available in Section 2.

This paper diverges from prior works that mainly have focused on reducing redundancy between adjacent frames and instead concentrates on *reducing the redundancy between adjacent optical flows* caused by inertia and we call this **motion context encoding**. To minimize redundancy in images, the reference frame needs to be warped to align with the current frame, for which the optical flow map from the current frame to the reference frame is required. This causes the necessity of additional bits for the optical flow map. Fortunately, in the case of motion context encoding, the required

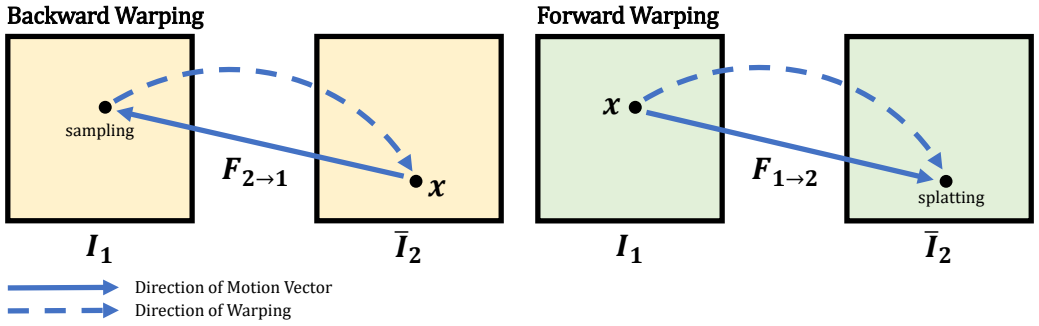

Figure 1: Backward Warping and Forward Warping.

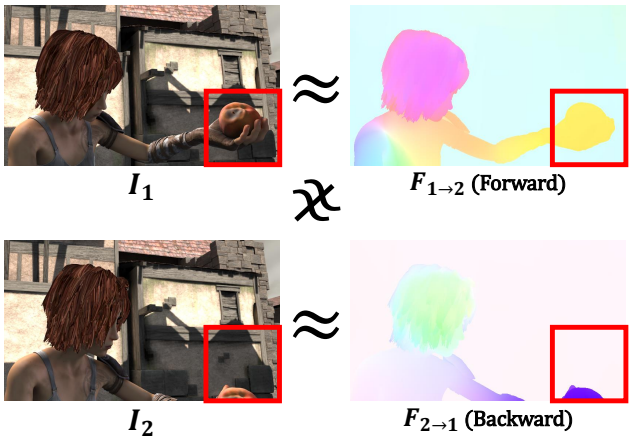

Figure 2: Edge alignment of the source image and optical flow.

motion vector for warping the reference optical flow map is the optical flow map itself, obviating the need for additional bits. However, the conventional backward warping-based paradigm presents a dilemma: to decode the current motion, the motion vector itself obtained from the decoding is needed to warp the reference optical flow map. In other words, the result of decoding is necessary for decoding. Detailed information on this issue is further addressed in Section 3.3.

To deal with this issue, we propose the use of forward warping instead of backward warping, as illustrated in Figure 1. In backward warping, since each motion vector corresponds to a pixel in the current frame, the target pixel is filled by sampling the pixel value at the location indicated by the motion vector. In contrast, in forward warping, each motion vector corresponds to a pixel in the reference frame, and these pixels are directly moved to the target position indicated by the motion vector to fill the target frame. By using forward warping, it becomes possible to warp the previous optical flow without problems, reducing redundancy between adjacent motions. This will be discussed in more detail in Section 3.3. Furthermore, unlike backward motion vectors, forward motion vectors are aligned with the reference frame (i.e. share the edges as shown in Figure 2), which allows for more efficient compression by utilizing redundancy between the optical flow map and the reference frame. Recently, an effective forward warping method called **splatting** has been introduced in the field of video frame interpolation (Niklaus & Liu, 2020; Niklaus et al., 2023).

In this context, we propose "splatting-based video compression (SVC)" that utilizes forward warping (splatting) to increase the compression efficiency of motion vectors. Unfortunately, unlike backward warping, splatting has some limitations such as the possibility of multiple pixels overlapping at the same location or creating holes. Therefore, this paper focuses on two main points. One is to minimize the aforementioned issues that forward warping has compared to backward warping. The other is to maximize the benefits of motion context encoding which is made possible through for-

ward warping. To achieve these, we first propose several methods to ensure that the forward warping module can achieve performance comparable to backward warping. Next, we propose a method to maximize the effect of motion context encoding.

2 Related Work

While several traditional codecs such as H.263, H.264 (Wiegand et al., 2003), H.265 (Sullivan et al., 2012), H.266 (Bross et al., 2021), AV1 (Chen et al., 2018) have been well-tuned and continue to be widely used, there is still a need for efficient and fast video compression methods that can match the quality of these legacy standards. Recent advances in machine learning (ML) have shown promise in achieving more efficient and flexible video compression.

One possible approach is to skip frames and then interpolate them back (Wu et al., 2018; Djelouah et al., 2019; Park & Kim, 2019). Unlike the indirect methods, Ballé et al. (2016) and Ballé et al. (2018) proposed end-to-end frameworks for image compression, leading to the emergence of various studies aiming to adapt these for video compression. Some works tried to (Lu et al., 2019; Liu et al., 2019; 2020) replace each part of the traditional codec with deep learning-based modules, allowing for the exploration of a more optimal compression domain. Thanks to these studies, the deep video compression area has also established a standardized framework that follows the three stages of motion estimation, motion compensation, and redundancy reduction, just like traditional codecs.

Subsequently, various papers have extended different aspects based on this baseline. Most notably, Agustsson et al. (2020) proposed a method of modeling motion using scale space flow, and various attempts based on this method have emerged (Yang et al., 2020b; Rippel et al., 2021). An approach using RNN or ConvLSTM without using CNN has also emerged (Yang et al., 2020a; Golinski et al., 2020). Hu et al. (2021) proposed Feature-space Video Coding (FVC) that uses warping and deformable convolutions in a smaller feature space and Habibian et al. (2019) introduced an approach using a 3D autoregressive entropy model. On the other hand, Li et al. (2021) suggested a framework called Deep Contextual Video Compression (DCVC) that concatenates features instead of simple subtraction of the reference image to find more optimal redundancy reduction and this work is our baseline.

Various mode prediction methods have been proposed that incorporate many of the tricks used in traditional codecs (Hu et al., 2022), and generative models are also being studied (Mentzer et al., 2022a; Yang et al., 2021; Ho et al., 2022). Recently, methods using transformers, which move away from convolutional operations, have been proposed (Mentzer et al., 2022b).

Research has also been actively conducted to efficiently compress not only images but also motion information. Lin et al. (2020) employed a method that predicts the next motion using decoded previous motions across multiple frames. Taking it a step further, Rippel et al. (2021) proposed a strategy to store only the residuals of the optical flow. However, since they utilize the unwrapped reference flow rather than aligning the reference flow to the current flow through warping, redundancy may not be effectively removed. Thus, in this paper, just as warping the reference frame in images has been proven effective, we propose a method that warps the reference flow to align it with the current flow.

3 Proposed Method

3.1 Preliminary

Let us consider compressing the t-th frame I_t from a video that consists of N frames $\{I_1, I_2, \cdots, I_N\}$. Most video compression methods aim to store only the newly appearing information in the current frame I_t by excluding the information that is already present in both the current frame and the previously decoded reference frame $\hat{I}_{t-1}$. The reason for using the decoded frame $\hat{I}_{t-1}$ instead of the original frame I_{t-1} is that the decoder cannot access the original frame.

To achieve this, most of the previous works obtain the warped frame $\overline{I}_t$ from the decoded reference frame $\hat{I}_{t-1}$ using backward warping $\overleftarrow{\omega}$ based on the decoded backward motion vector (optical flow)

$\hat{F}_{t\to t-1}$ from I_t to $\hat{I}_{t-1}$, as follows (Please refer to Equation 5 for the mathematical definition of $\overleftarrow{\omega}$.).

$$\overline{I}_t = \overleftarrow{\omega}(\hat{I}_{t-1}, \hat{F}_{t\to t-1}) \tag{1}$$

Various methods can be used to remove redundancy between $\overline{I}_t$ and I_t, but typically, the residual information ΔI_t is obtained by subtracting $\overline{I}_t$ from I_t as Equation 2.

$$\Delta I_t = I_t - \overline{I}_t \tag{2}$$

Then ΔI_t and the backward optical flow map $F_{t\to t-1}$ are encoded and transmitted. At the decoder end, the decoded reference frame $\hat{I}_{t-1}$, the decoded optical flow map $\hat{F}_{t\to t-1}$, and the decoded residual frame $\Delta\hat{I}_t$ are provided. The restored frame $\hat{I}_t$ is then obtained through the following process.

$$\overline{I}_t = \overleftarrow{\omega}(\hat{I}_{t-1}, \hat{F}_{t\to t-1}) \tag{3}$$

$$\hat{I}_t = \Delta\hat{I}_t + \overline{I}_t \tag{4}$$

3.2 Backward Warping and Forward Warping

Let us consider two frames, I_0 and I_1, and warp I_0 to align with I_1. For backward warping, we need backward motion $F_{1\to 0}$ from I_1 to I_0, and we rearrange the pixels as follows.

$$\overline{I}_1(\boldsymbol{x}) = \overleftarrow{\omega}(I_0, F_{1\to 0})(\boldsymbol{x}) = I_0(\boldsymbol{x} + F_{1\to 0}(\boldsymbol{x})) \tag{5}$$

In Equation 5, if $\boldsymbol{x} + F_{1\to 0}(\boldsymbol{x})$ does not point to a grid location of I_0, the pixel value is sampled using bilinear interpolation from the closest four pixels.

In the case of forward warping, we utilize the average splatting proposed in (Niklaus & Liu, 2020). To do this, we first define summation splatting as follows.

$$\beta(\boldsymbol{u} = [u_i, u_j]) = \max(0, 1 - |u_i|) \cdot \max(0, 1 - |u_j|) \tag{6}$$

$$\overrightarrow{\omega}_{sum}(I_0, F_{0\to 1})(\boldsymbol{x}) = \sum_{\forall \boldsymbol{p} \in I_0} \beta(\boldsymbol{x} - (\boldsymbol{p} + F_{0\to 1}(\boldsymbol{p}))) \cdot I_0(\boldsymbol{p}) \tag{7}$$

Then the average splatting $\overrightarrow{\omega}_{avg}$ is defined as follows (the notation for x is omitted because it is an independent operation for all locations x in Equation 8. $\mathbf{1}$ is an array consisting of ones and has the same resolution as I_0.).

$$\overline{I}_1 = \overrightarrow{\omega}_{avg}(I_0, F_{0\to 1}) = \frac{\overrightarrow{\omega}_{sum}(I_0, F_{0\to 1})}{\overrightarrow{\omega}_{sum}(\mathbf{1}, F_{0\to 1})} \tag{8}$$

3.3 Motion Context Encoding

The main purpose of this paper is to minimize redundancy not only between adjacent images but also between adjacent optical flow maps. To compress the current motion vector $F_{t\to t-1}$, a reference motion vector $F_{t-1\to t-2}$ is required. Therefore, two reference frames $\hat{I}_{t-1}$ and $\hat{I}_{t-2}$ are needed to compress I_t using this method. We can obtain the two motion vectors using a motion estimation network ME as follows.

$$F_{t\to t-1} = ME(I_t, \hat{I}_{t-1}), \quad F_{t-1\to t-2} = ME(\hat{I}_{t-1}, \hat{I}_{t-2}) \tag{9}$$

Then we can get the warped reference motion vector $\overline{F}_{t\to t-1}$ by backward warping $F_{t-1\to t-2}$ using $F_{t\to t-1}$ as follows (We should warp according to the decoded flow $\hat{F}_{t\to t-1}$, but since it's before decoding, let's assume we use $F_{t\to t-1}$ for now.).

$$\overline{F}_{t\to t-1} = \overleftarrow{\omega}(F_{t-1\to t-2}, F_{t\to t-1}) \tag{10}$$

$$\Delta F_{t\to t-1} = F_{t\to t-1} - \overline{F}_{t\to t-1} \tag{11}$$

This part can be confusing, but simply put, F_{AB} has no direct relationship with I_B and is just the same image with a different modality that shares the same structure with I_A, such as edges (as shown in Figure 2).

Obtaining $\overline{F}_{t\to t-1}$ through backward warping is not a problem in the encoding process, but it causes a dilemma in the decoding process. During decoding, we cannot use the original flow $F_{t\to t-1}$, so we have to use the decoded flow $\hat{F}_{t\to t-1}$; however, this is also unobtainable before decoding (see Equations 12 and 13.).

$$\overline{F}'_{t\to t-1} = \overleftarrow{\omega}(F_{t-1\to t-2}, \hat{F}_{t\to t-1}) \tag{12}$$

$$\hat{F}_{t\to t-1} = \Delta F_{t\to t-1} + \overline{F}'_{t\to t-1} \tag{13}$$

A simple solution to this dilemma is to use forward motion instead of backward motion as follows.

$$F_{t-1\to t} = ME(\hat{I}_{t-1}, I_t), \quad F_{t-2\to t-1} = ME(\hat{I}_{t-2}, \hat{I}_{t-1}) \tag{14}$$

Then, we can obtain $\overline{F}_{t-1\to t}$ by forward warping $F_{t-2\to t-1}$ using $F_{t-2\to t-1}$ itself as follows.

$$\overline{F}_{t-1\to t} = \overrightarrow{\omega}_{avg}(F_{t-2\to t-1}, F_{t-2\to t-1}) \tag{15}$$

$$\Delta F_{t-1\to t} = F_{t-1\to t} - \overline{F}_{t-1\to t} \tag{16}$$

As a result, at the decoding stage, we only need the reference motion $F_{t-2\to t-1}$ which can be easily obtained from the two reference frames instead of the current motion $F_{t-1\to t}$. Therefore we can decode $\hat{F}_{t-1\to t}$ without any problems as shown in Equations 17 and 18.

$$\overline{F}_{t-1\to t} = \overrightarrow{\omega}_{avg}(F_{t-2\to t-1}, F_{t-2\to t-1}) \tag{17}$$

$$\hat{F}_{t-1\to t} = \Delta F_{t-1\to t} + \overline{F}_{t-1\to t} \tag{18}$$

3.4 Baseline Architecture

The baseline model used in this paper is fundamentally based on DCVC (Li et al., 2021). In other words, instead of simply subtracting the current and reference frames as in Equation 2, we adopt an approach that encodes the context obtained from the features extracted from both frames. Consequently, the warping operations (Equations 5 and 8) involve warping their feature maps rather than directly warping the image or flow map. However, for clarity, we will omit this detail in the notation. In addition, the obtained context is encoded via Hyperprior (Ballé et al., 2018) method. Moreover, we improve the baseline by applying several effective tricks recently proposed. The following changes are applied to our baseline.

Checkerboard Context Module. To utilize the mutual information between feature vectors within a single frame, DCVC adopts the autoregressive encoding (Minnen et al., 2018) approach. However, this method has the drawback of slow computation speed since it cannot process all pixels in a frame simultaneously in parallel. Therefore, instead of this module, we use the checkerboard context model (He et al., 2021), which can refer to the already decoded surrounding pixels while enabling parallel processing.

Multiscale Feature. DCVC only uses features extracted from a single scale to extract context. However, it has been proven that utilizing multi-scale features is more effective in various research

fields, including video compression (Li et al., 2022). Therefore, our baseline also uses features extracted from five different scales, each of which is warped to obtain context.

Rate Controllable EASN. DCVC uses Generalized Divisible Normalization (GDN)(Ballé et al., 2016) as the activation function for the hyperprior encoder and decoder. Instead of GDN, we use EASN (Shin et al., 2022) which has been proposed and shown to be more stable and perform better than GDN as the activation function in our model. Additionally, we improve the model by incorporating a learnable vector table to EASN and passing the sampled vector along with the encoded features. This allows us to adjust the rate parameter to control the rate-distortion tradeoff without retraining the model.

3.5 Minimizing the Drawbacks of Forward Warping

As mentioned in Section 1, although forward warping has the advantage of enabling motion context encoding, it has some problems compared to backward warping. One of the most typical issues is that two or more pixels can be projected onto the same location, or no pixels may be projected on somewhere else, resulting in holes. Fortunately, we do not directly calculate the residual through subtraction like Equation 2, but instead extract context like DCVC (Li et al., 2021). This allows us to have some robustness against holes and pixel overlapping. However, forward warping still has some drawbacks compared to backward warping, and there is room for improvement. In this section, we propose three effective tricks to minimize the performance degradation of the proposed forward warping module compared to traditional backward warping.

Reference Image Guidance. As shown in Figure 2, unlike backward motion $F_{t\to t-1}$, forward motion $F_{t-1\to t}$ shares the structure with the reference frame I_{t-1}, which leads to mutual information. Therefore, we provide I_{t-1} as guidance to the motion encoder and decoder, enabling them to omit the flow edge information.

Flow Reversing. The problems of overlapping pixels and holes can also interfere with the smooth flow of gradients during end-to-end learning. For example, if all pixels are projected to one point and all other locations are holes, the gradient flows only through one pixel. Therefore, instead of directly splatting the reference image (or feature map in the case of the DCVC baseline) like Equation 8, we obtain a pseudo backward flow $\tilde{F}_{t\to t-1}$ by reversing and splatting the reversed optical flow $-F_{t-1\to t}$ like (Niklaus et al., 2023), as follows.

$$\tilde{F}_{t\to t-1} = \overrightarrow{\omega}_{avg}(-F_{t-1\to t}, F_{t-1\to t}) \tag{19}$$

Then we can get the forward warped result by backward warping with $\tilde{F}_{t\to t-1}$ as follows.

$$\overline{I}_t = \overrightarrow{\omega}_{rev}(\hat{I}_{t-1}, F_{t-1\to t}) = \overleftarrow{\omega}(\hat{I}_{t-1}, \tilde{F}_{t\to t-1}) \tag{20}$$

Equation 20 is essentially forward warping, but the operation actually applied to $\hat{I}_{t-1}$ is backward warping, which allows for smooth gradient flow.

Gradient Stopping. While flow reversing enables smooth gradient flow for the images or feature maps being warped, the gradient of Equation 19 can still affect the optical flow $F_{t-1\to t}$ and performance of motion estimation. To prevent it, we apply a gradient stop $S[\cdot]$ to the second term in Equation 19, and only allow the gradient to flow through the first term as follows.

$$\tilde{F}_{t\to t-1} = \overrightarrow{\omega}_{avg}(-F_{t-1\to t}, S[F_{t-1\to t}]) \tag{21}$$

3.6 Maximizing the Advantages of Motion Context Encoding

In Section 3.3, we obtaine the reference $\overline{F}_{t-1\to t}$ of current motion $F_{t-1\to t}$ by warping $F_{t-2\to t-1}$ using Equation 17. However, as mentioned in Section 3.5, forward warping operation has several limitations. Fortunately, unlike in the case of warping images, there exists an excellent alternative that has a strong correlation with $F_{t-1\to t}$ and does not require warping. It is the reverse of $F_{t-1\to t-2}$.

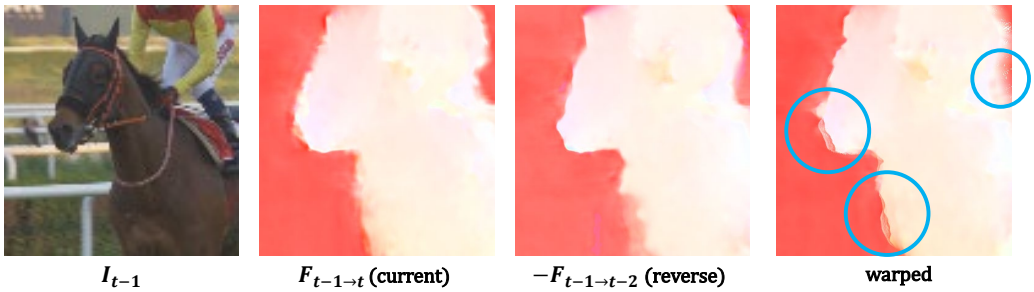

Figure 3: Reference candidates of current motion.

$$\overline{F}_{t-1\to t} = -F_{t-1\to t-2} \tag{22}$$

As can be seen in Figure 3, the reference obtained through warping (4$^{\text{th}}$ image of Figure 3) suffers from artifacts due to pixel overlapping compared to $-F_{t-1\to t-2}$ (3$^{\text{rd}}$ image of Figure 3).

3.7 Loss Function

We use Rate-Distortion (RD) trade-off loss following (Ballé et al., 2016; 2018) to train the proposed model.

$$L = R + \lambda D = R + \lambda d(I, I^*), \tag{23}$$

where R denotes the bit rate and d denotes distortion function. We use mean squared error (MSE) loss as d.

4 Experiments

4.1 Settings

To demonstrate the effectiveness of the contribution and main idea of this paper, we test various versions of our method and some comparison algorithms on UVG (Mercat et al., 2020), HEVC (Sullivan et al., 2012), and MCL-JCV (Wang et al., 2016) datasets. The UVG and MCL-JCV datasets consist of frames of size 1920×1024, while the HEVC dataset is Composed of four classes B, C, D, E with various sizes. All test videos are center-cropped to have a width and height that are multiples of 64, and the test GOP (Group Of Pictures) size is fixed at 12.

4.2 Training

Training Dataset. We use the Vimoe-90K (Xue et al., 2019) septuplet dataset consisting of 89,800 video clips with 7 frames of size 448×256 and. For training, we randomly crop each frame to size 256×256 and use random horizontal/vertical flips and rotations for data augmentation.

Implementation Detail. We implement the proposed model using the PyTorch (Paszke et al., 2019) library and utilize the Cupy (Okuta et al., 2017) library for splatting operation. We use the AdamW (Loshchilov & Hutter, 2017) optimizer with a batch size of 4 and a fixed learning rate of 10^{-4}. We train the model for a total of 46 epochs, where the first 6 epochs are dedicated to training the motion encoder, the next 6 epochs are dedicated to training the image encoder, and the remaining 34 epochs are dedicated to training the entire model simultaneously.

4.3 Ablation Study

This section verifies whether the contributions of the paper, including forward warping framework and the tricks introduced in Section 3.5 and Section 3.6, are actually effective.

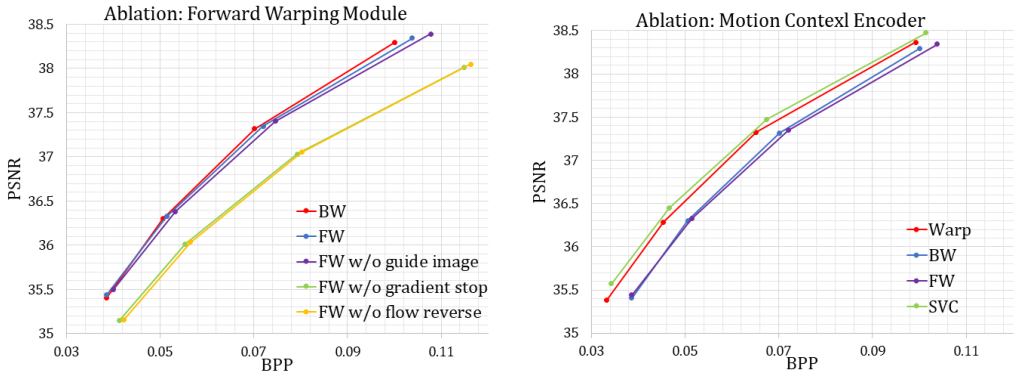

Figure 4: (a) Ablation study on forward warping module (BW means backward warping and FW means forward warping.). (b) Ablation study on motion context encoder.

Figure 4 (a) presents the results of the experiments verifying the tricks introduced in Section 3.5. All versions in Figure 4 (a) are the results without applying motion context encoding. As mentioned, the performance of the forward warping operation itself is inferior to that of the backward warping. To minimize this gap, the version that applies all three tricks in Section 3.5 is FW. It shows that our forward warping module has succeeded in approaching the performance of backward warping. In particular, gradient stop and flow reverse can be concluded as essential tricks for stable training, therefore they are necessary for forward warping.

Figure 4 (b) demonstrates that the version that reverses the backward flow (SVC) performs better than the version that warps the reference motion vector (Warp) as introduced in Section 3.6. In addition, despite the drawback of forward warping, the attempt to reduce the redundancy of motion vectors through motion context encoding shows better performance than the backward warping-based baseline, proving the effectiveness of the main idea of this paper.

4.4 Comparison

We compare our SVC with x265 (Sullivan et al., 2012) veryslow preset, VTM (Bross et al., 2021) and HM codec with low delay mode, and DCVC (Li et al., 2021), a representative deep learning-based video compression method. Most of the previous deep learning-based video compression papers have often compared their models with other state-of-the-art models without standardizing the intra mode, which cannot be considered a fair comparison. Our model has two characteristics related to intra frame coding. The first is that it can start compression from the third frame because it requires two reference frames. The second is that we do not propose a fixed intra method, so any model can be plugged in without any issues. Therefore if we are comparing its performance with another codec $\mathcal{C}$, the first and second frames are compressed with $\mathcal{C}$. This method allows for a fair comparison with any other codec. Figure 5 shows that the proposed model, SVC, clearly outperforms x265 veryslow, DCVC, and HM, and performs similarly or even slightly better for large λ values compared to VTM.

5 Limitation

The method proposed in this paper is one of many possible forward warping methods. It is not optimal and there are still many issues such as unstable training to be addressed compared to backward warping. We hope that the forward warping-based paradigm will continue to evolve through future research in this field.

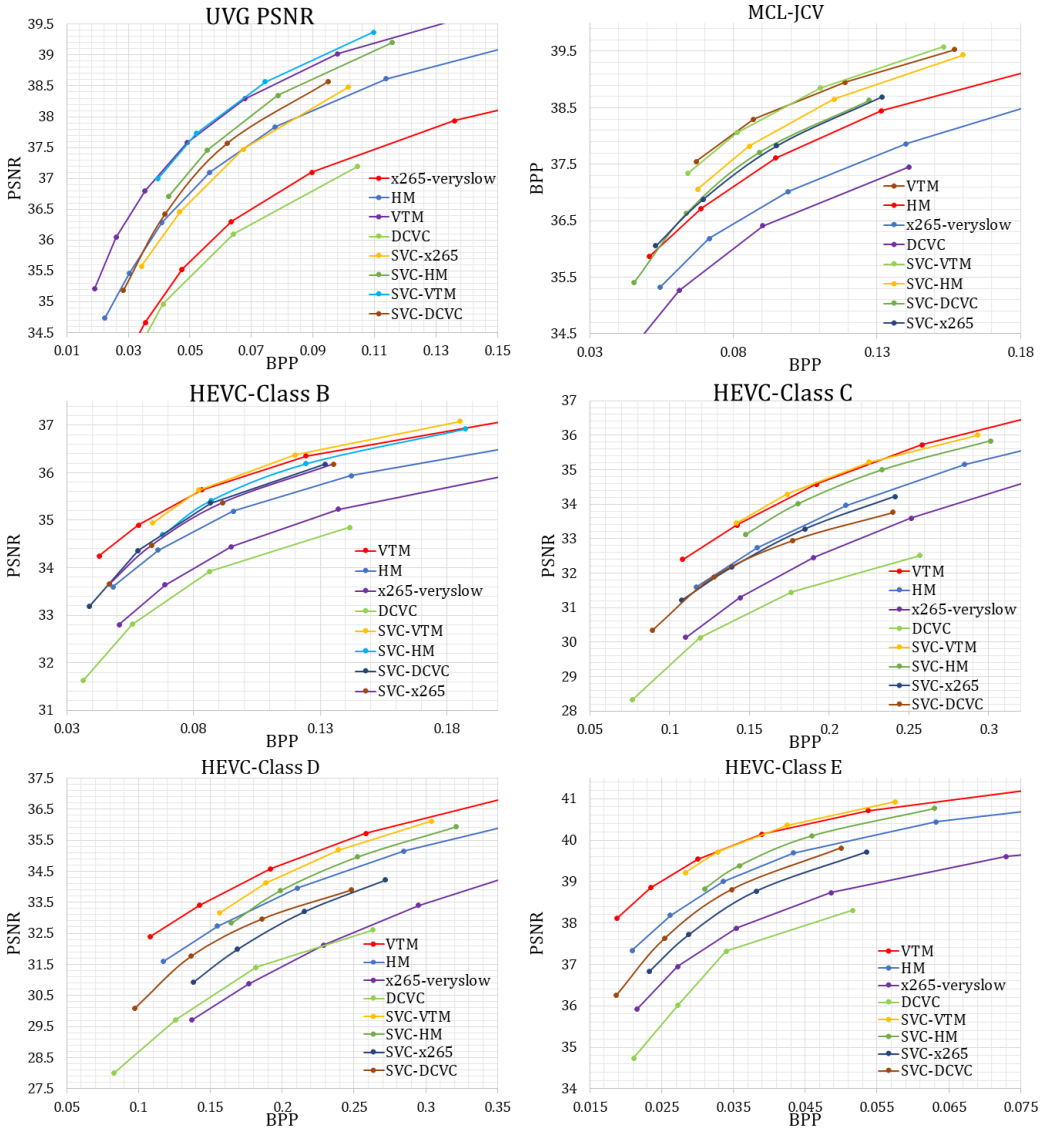

Figure 5: The comparisons of rate distortion curves on various test datasets.

6 Conclusion

In this paper, we explore the dilemma arising from directly applying the traditional backward warping, used for compressing frames, to motion. To circumvent this issue, we propose a forward warping-based method. This allows the reference optical flow map to align with the current motion, enabling more efficient motion compression. However, some inherent challenges associated with forward warping become evident. We therefore introduce several tricks to address these challenges. Through experimentation, we confirm that motion context warping via forward warping is more effective than existing methods. We anticipate that simply reversing the warping direction, as demonstrated, could not only benefit video compression but also serve as a game changer in other motion-related video processing fields.

References

Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8503–8512, 2020.

Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021.

Yue Chen, Debargha Murherjee, Jingning Han, Adrian Grange, Yaowu Xu, Zoe Liu, Sarah Parker, Cheng Chen, Hui Su, Urvang Joshi, et al. An overview of core coding tools in the av1 video codec. In *2018 picture coding symposium (PCS)*, pp. 41–45. IEEE, 2018.

Abdelaziz Djelouah, Joaquim Campos, Simone Schaub-Meyer, and Christopher Schroers. Neural inter-frame compression for video coding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6421–6429, 2019.

Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2758–2766, 2015.

Adam Golinski, Reza Pourreza, Yang Yang, Guillaume Sautiere, and Taco S Cohen. Feedback recurrent autoencoder for video compression. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

Amirhossein Habibian, Ties van Rozendaal, Jakub M Tomczak, and Taco S Cohen. Video compression with rate-distortion autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7033–7042, 2019.

Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14771–14780, 2021.

Yung-Han Ho, Chih-Peng Chang, Peng-Yu Chen, Alessandro Gnutti, and Wen-Hsiao Peng. Canf-vc: Conditional augmented normalizing flows for video compression. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI*, pp. 207–223. Springer, 2022.

Zhihao Hu, Guo Lu, and Dong Xu. Fvc: A new framework towards deep video compression in feature space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1502–1511, 2021.

Zhihao Hu, Guo Lu, Jinyang Guo, Shan Liu, Wei Jiang, and Dong Xu. Coarse-to-fine deep video coding with hyperprior-guided mode prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5921–5930, 2022.

Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. *Advances in Neural Information Processing Systems*, 34:18114–18125, 2021.

Jiahao Li, Bin Li, and Yan Lu. Hybrid spatial-temporal entropy modelling for neural video compression. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1503–1511, 2022.

Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. M-lvc: Multiple frames prediction for learned video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3546–3554, 2020.

Haojie Liu, Tong Chen, Ming Lu, Qiu Shen, and Zhan Ma. Neural video compression using spatio-temporal priors. *arXiv preprint arXiv:1902.07383*, 2019.

Jerry Liu, Shenlong Wang, Wei-Chiu Ma, Meet Shah, Rui Hu, Pranaab Dhawan, and Raquel Urtasun. Conditional entropy coding for efficient video compression. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII*, pp. 453–468. Springer, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11006–11015, 2019.

Fabian Mentzer, Eirikur Agustsson, Johannes Ballé, David Minnen, Nick Johnston, and George Toderici. Neural video compression using gans for detail synthesis and propagation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pp. 562–578. Springer, 2022a.

Fabian Mentzer, George Toderici, David Minnen, Sung-Jin Hwang, Sergi Caelles, Mario Lucic, and Eirikur Agustsson. Vct: A video compression transformer. *arXiv preprint arXiv:2206.07307*, 2022b.

Alexandre Mercat, Marko Viitanen, and Jarno Vanne. Uvg dataset: 50/120fps 4k sequences for video codec analysis and development. In *Proceedings of the 11th ACM Multimedia Systems Conference*, pp. 297–302, 2020.

David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018.

Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5437–5446, 2020.

Simon Niklaus, Ping Hu, and Jiawen Chen. Splatting-based synthesis for video frame interpolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 713–723, 2023.

Ryosuke Okuta, Yuya Unno, Daisuke Nishino, Shohei Hido, and Crissman Loomis. Cupy: A numpy-compatible library for nvidia gpu calculations. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017. URL `http://learningsys.org/nips17/assets/papers/paper_16.pdf`.

Woonsung Park and Munchurl Kim. Deep predictive video compression with bi-directional prediction. *arXiv preprint arXiv:1904.02909*, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4161–4170, 2017.

Oren Rippel, Alexander G Anderson, Kedar Tatwawadi, Sanjay Nair, Craig Lytle, and Lubomir Bourdev. Elf-vc: Efficient learned flexible-rate video coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14479–14488, 2021.

Chajin Shin, Hyeongmin Lee, Hanbin Son, Sangjin Lee, Dogyoon Lee, and Sangyoun Lee. Expanded adaptive scaling normalization for end to end image compression. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pp. 390–405. Springer, 2022.

Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8934–8943, 2018.

Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C-C Jay Kuo. Mcl-jcv: a jnd-based h. 264/avc video quality assessment dataset. In *2016 IEEE international conference on image processing (ICIP)*, pp. 1509–1513. IEEE, 2016.

Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7): 560–576, 2003.

Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 416–431, 2018.

Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127:1106–1125, 2019.

Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. Learning for video compression with recurrent auto-encoder and recurrent probability model. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):388–401, 2020a.

Ren Yang, Luc Van Gool, and Radu Timofte. Perceptual learned video compression with recurrent conditional gan. *arXiv preprint arXiv:2109.03082*, 1, 2021.

Ruihan Yang, Yibo Yang, Joseph Marino, and Stephan Mandt. Hierarchical autoregressive modeling for neural video compression. *arXiv preprint arXiv:2010.10258*, 2020b.