Supplementary Material of Structure-Preserving Motion Estimation for Learned Video Compression

Han Gao
han.gao@std.uestc.edu.cn
School of CSE, University of Electronic Science and Technology of China
Chengdu, China

Jinzhong Cui
jzcui@uestc.edu.cn
School of CSE, University of Electronic Science and Technology of China
Chengdu, China

Mao Ye*
cvlab.uestc@gmail.com
School of CSE, University of Electronic Science and Technology of China
Chengdu, China

Shuai Li*
shuaili@sdu.edu.cn
School of Control Science and Engineering, Shandong University
Jinan, China

Yu Zhao
School of CSE, University of Electronic Science and Technology of China
Chengdu, China

Xiatian Zhu
Surrey Institute for People-Centred Artificial Intelligence, CVSSP, University of Surrey
Guildford, UK

1 OVERVIEW

This document provides the supplementary material to our proposed structure-preserving motion estimation for learned video compression, including more implementation details, deeper explanation, additional experimental results and ablation studies to demonstrate the effectiveness of the proposed scheme.

2 MORE DETAILS

2.1 Implementation details

We set up four λ values (MSE: 256, 512, 1024, 2048; MS-SSIM: 8, 16, 32, 64) to fit rate distortion tradeoff. For each value of λ, we use Eq. 7 in the paper to train the model for 80 epochs in an end-to-end manner, including 60 normal epochs and 20 fine tuning epochs. When MS-SSIM is used to measure performance, we further use MS-SSIM loss function to continue optimization from epoch 80 to achieve the best performance. We set the batch size as 4 and use the Adam optimizer [2]. The learning rate is setting to $5e-5$ and $1e-5$ at the normal stage and the fine tuning stage respectively. The model is implemented on Pytorch and trained on a single NVIDIA RTX 3090 GPU for 8 days.

2.2 Details for DCVC-based model

As for DCVC-based model, in order to train the model more effectively, and fully integrate our proposed method into the DCVC [3] framework, we adopt a phased training strategy. Specifically, based on the released pretrained model by the author of DCVC, we first warm up our modules for 2 epochs with the learning rate as $1e-4$, including Highly-fitting motion field generation and Context enhancement, meanwhile the parameters of other parts are frozen. After that, we reopen the other parts and further perform the end-to-end training of whole framework with the learning rate as $1e-5$ for 3 epochs. The loss function used in the whole training process is Eq. 7 in the paper. The other settings such as datasets and evaluation metrics are the same as that mentioned in Sec. 2.1.

*Mao Ye and Shuai Li are corresponding authors. This work was supported by the National Key R&D Program of China (2018YFE0203900) and Sichuan Science and Technology Program (2020YFG0476).

3 DEEPER EXPLANATION

3.1 The source of performance gain

Video coding aims to minimize the distortion under a rate constraint, or in other words, achieve a good rate-distortion (R-D) trade-off. Our method is R-D optimized, but by employing both the original and reconstructed/decoded reference frame, instead of only reconstructed reference frame.

Our method strikes a good trade-off between structure preserving and R-D performance. It provides the motion estimation with the structure information required and generates a motion field with appropriate R-D cost. First, as discussed in the Abstract and Introduction in the paper, due to the strong CNN representation capability, for a CNN to encode the motion field and the corresponding residual efficiently, structure information is better preserved. Consider a motion field with a face against one with random noise of similar (or even smaller) average magnitudes, a CNN obviously encodes the structured face image better than the random noise. Second, encoding a high-fidelity image generally costs more bits than a blurred one. Therefore, instead of using the original optical flow, R-D optimization on the motion estimation is also necessary to improve the overall coding performance.

3.2 Mismatching problem

There might be a misunderstand that there is a mismatch between the decoded reference frame used for motion compensation and the original reference frame used for motion estimation, leading to sharp decline in performance. We would like to explain it as follows:

Structure information is helpful for CNN coding. We would like to emphasize the difference between the conventional video coding and the deep learning based one. For the conventional coding, the smaller residuals usually lead to a smaller rate, but for CNN based coding, the structure in both motion fields and residuals matters a lot. Our method provides the structure information to the motion estimation and the corresponding residual under the R-D optimization framework, thus improving the overall performance. Let's consider a valid case that the decoded reference frame is distorted and very blurry. When performing motion estimation,

Table 1: Ablation on the HEVC Class E dataset

	Motion bpp	Residual bpp	PSNR
FVC*	0.003355	0.008675	36.545179
SPME (FVC*)	0.003194	0.008186	36.755618

there exist many motion vectors for each pixel to achieve similar residuals, and the best one is affected by the distortion (which can be treated as random). Accordingly, the estimated motion fields are rather random and the resulted residuals are also random to some extent. This would significantly increase the bit rate for CNN coding, lowering the R-D performance. By contrast, using the original reference frame, our method provides structured but blurry motion field, which requires much less bits and provides slightly larger residuals but also structured. This can also be efficiently encoded considering the large improvement from the existing compressed video quality enhancement methods. Furthermore, our method also keeps the temporal correlation from the distortion, in turn enhancing the temporal prediction efficiency over the motion fields and reducing the error propagation in the temporal dimension as shown in Fig. 8 in the paper, improving the overall coding performance.

4 ADDITIONAL EXPERIMENTAL RESULTS

4.1 Running Time and Model Complexity

We conducted the complexity analysis for verify the effectiveness of our proposed scheme. We tested FVC* [1] (w/ 16.4M parameters) as the base model on HEVC Class D dataset [4] with a single NVIDIA 2080Ti GPU. For the single frame inference time, FVC* takes 23.9ms vs 34.3ms when our SPME (w/ 3.3M parameters) is added on top. This small cost increase is well worthy for achieving a save of 5.05% bit-rate.

4.2 Concrete bit rate comparison

To verify that "the structure-damaged motion (e.g. noised motion of a very blurry reference area) would significantly increase the bit rate for CNN coding when compared to the structure-preserved motion", we have compared the bit-rate of different models, using the HEVC Class E dataset [4] with $lambda = 1024$ as an example. The results in the Table 1 show clearly that our method can lower the bit-rate of both motion field and corresponding residual consistently. Moreover, the reconstruction quality in PSNR metric is also improved thanks to the structure-preserved motion field, all together leading to a better overall R-D performance (see Fig. 7 in the paper).

REFERENCES

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

[2] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[3] Jiahao Li, Bin Li, and Yan Lu. 2021. Deep contextual video compression. *Advances in Neural Information Processing Systems* 34 (2021).

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